SIEMENS

Remote Access to SIMATIC HMI Comfort Panels

User-defined web pages

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 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 will be excluded. Such an exclusion will 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 a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The compensation for damages due to a breach of a fundamental contractual obligation is, 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 Industry Sector.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.automation.siemens.com.

Table of Contents

1 Task

Gaining information about your plant quickly and easily requires viewing this information at a glance, clearly prepared according to your requirements. However, you do not need the same technical details or control options that are required by the plant operator. You need to be able to see whether the production data complies with the schedule at a glance, and send adjustments directly to the operator, if necessary. The operation of the panel on site should not be affected by your action.
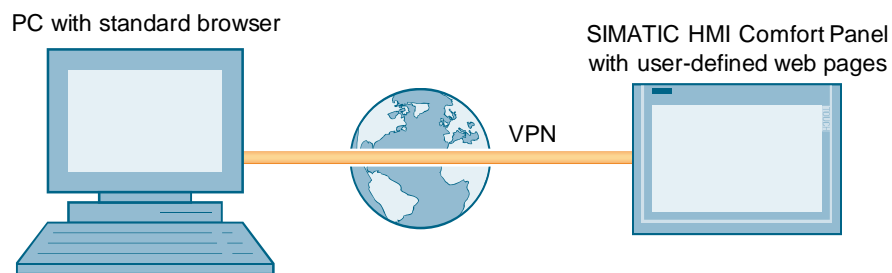
Figure 1-1
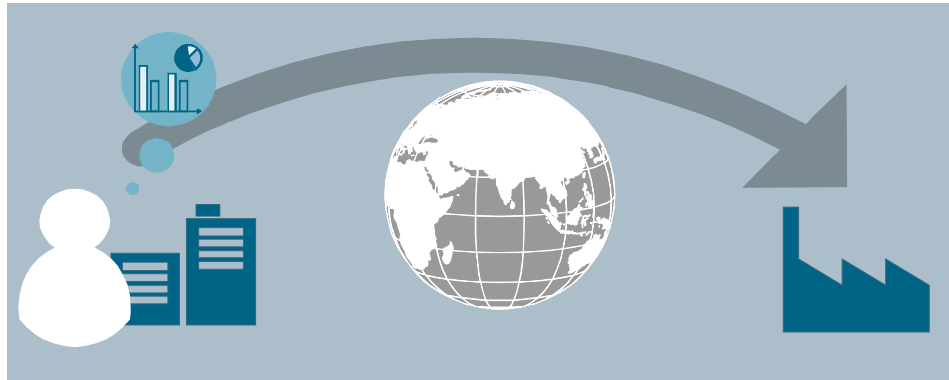
2 Solution

2.1 Overview

Schematic layout

The figure below shows a schematic overview of the most important components of the solution:

Figure 2-1

Setup

For the SIMATIC HMI Comfort Panel, user-defined web pages can be created. The HMTL pages can then be loaded into the operator panel along with the configuration. It can be generated using any text editor. You can use the full functionality of HMTL and CSS for creating the web pages.

You can then access the web pages of the operator panel from any device via a standard web browser.

Advantages

The solution presented here, offers you the following advantages

- Display via standard web browser, no additional software is necessary
- No additional costs - contained in the delivery scope of the comfort panel

Topics not covered in this application

This application does not contain a description of:

- the basics on configuring with WinCC (TIA Portal)
- the configuration process in STEP7 (TIA Portal).
- setting up a VPN connection for the SIMATIC HMI Comfort Panel. Information on this topic is available in the "VPN connection" document.
- comprehensive information on creating HMI pages.

2.2 Hardware and software components

2.2.1 Validity

This application is valid for

- SIMATIC HMI Comfort Panels

2.2.2 Components used

The application was created with the following components:

Hardware components

Table 2-1

Component	Qty	Article number	Note
TP1200 Comfort	1	6AV2124-0MC01-0AX0	
S7-1516-3 PN/DP	1	6ES7 516-3AN00-0AB0	The program blocks are only used for simulating process values; a controller is not necessary for the "user-defined web pages" functionality.
Standard PC / laptop	1		Standard browser installed, such as MS Internet Explorer

Software components

Table 2-2

Component	Qty	Article number	Note
WinCC Comfort V13 SP1	1	6AV2101-0AA03-0AA5	
STEP 7 Professional V13 SP1	1		
MS FrontPage	1		Other software tools for creating and editing HTML pages possible
Web browser	1		

Configuration and use of the VPN connection requires further components. Further information is available in the "VPN connection" document.

2.2 Hardware and software components

Sample files and projects

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

Table 2-3

Component	Note
109476153_Remote_Panels_HTML_PROJ.zip	This zip file includes the TIA Portal project as well as the prepared HTML pages.
109476153_Remote_Panels_HTML_DOKU_en.pdf	This document

3 Basics

3.1 HTML (Hypertext Markup Language)

HTML (Hypertext Markup Language) is not a programming language but a text-based markup language for structuring contents such as texts, pictures, headers etc..

HTML documents are the basis of the world wide web (WWW) and are displayed by a web browser (for example, MS Internet Explorer).
Apart from the contents of a web page displayed by the browser, HTML contains additional information in the form of meta data about the language used in the text, the author etc.

The markup language is developed further by the World Wide Web Consortium (W3C).

3.1.1 Structure of HTML tags

A HTML tag refers to an individual HTML command. It starts with the character "<" and ends with the character ">". The instruction is located between the characters. The instruction is an English abbreviation.

Example:
HTML Code: This is an example
Browser view: **This is an example**

The "b" is an instruction and represents "bold". Due to this command, the text within the instruction is displayed in "bold" print.

The "" end tag is almost identical with the start tag. The only difference is that the "<" character is followed by a "/".

HMTL tags always come in small print.

3.1.2 HTML instructions

Below, there are some instructions that also have been used in the application. The instructions listed here already offer a variety of design options.

On the Internet you will find an extensive collection of "HTML instructions" and their application spectrum. You will furthermore find detailed descriptions regarding the individual parameters which you can use with an instruction.

HTML basic structure

The screen below shows the general structure of an HTML page.
For some HTML editors, this set up is automatically generated when creating a new page. Nevertheless, you can also execute this yourself or copy the existing "basic structure" of an HTML page and paste it into your new page.

3.1 HTML (Hypertext Markup Language)

Figure 3-1

```
 1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
 2
 3 <html>
 4
 5     <head>
 6
 7         <meta http-equiv="content-type" content="text/html; charset=windows-1252">
 8         <meta name="author" content="Siemens Industry Online Support">
 9
10         <link href="mycssfile.css" rel="stylesheet" type="text/css">
11
12         <title>Your Webpage title</title>
13
14     </head>
15
16
17     <body>
18
19         Your HTML-Code
20
21     </body>
22
23 </html>
```

Table 3-1

	HML Code	Notes
1.	<!DOCTYPE ...>	Information on the used HTML version. Can be used like this as template.
2.	<html> </html>	Start and end of the HTML code.
3.	<head> </head>	Header area where general information is given.
4.	<meta ...>	Part of the "header area". (e.g., <meta name="author" content= "your name").
5.	<link ...>	Reference to the CSS file.
6.	<title> </title>	The "Title" has several functions: • caption of the display window • name of set bookmark
7.	<body> </body>	The actual content of the HTML page is located within this instruction.

General HTML commands

Table 3-2

	HTML code	Notes
8.	<!-- Text -->	**Comment:** This is how comments are noted in HTML code. The text between the instructions is not shown in the browser.
9.	Text	**Paragraph:** The distance between to text lines is enlarged.
10.	 Text 	**Bold:** The text between the instructions is displayed in "bold" print.
11.	<hx> Text <hx/>	**Title:** For the "x" specify a value from 1 to 6. (1 very big, 6 very small)

3.1 HTML (Hypertext Markup Language)

	HTML code	Notes

12.	`<img src="" alt="">`
	
	Example:
	`<img src="picture.jpg" alt="name of graphic">`
		Reference to picture:
		An <img> tag is a standalone tag without element content and without end tag. With the help of attributes you determine the details of the graphic reference. Two attributes are mandatory and must always be specified.
		
		• **src=**
		The src attribute specifies the desired graphics file.
		• **alt=**
		Here, you enter an alternative text in case the graphic cannot be shown.

13.	`class="..."`	Class=" " specifies that the HTML element belongs to a certain "style sheet". The element copies formats that have been stored centrally in the CSS file.

14.	`<form>`
	
	`</form>`
		With the instruction you define a form.
		
		Everything between the introductory <form> tag and the final </form> tag is part of the form. These are mainly elements such as input fields, selection lists or buttons.

15.	`<a href="start.html"> a name </a>`	**Internal link:** "a name" stands for any text which may, for example, contain link specifications.
16.	`Id="…"` `Name="…"`	**ID/name:** Defines an ID or a name for the HTML element (e.g. picture) to be able to reference this object.
17.	`<input ..>`	**Input field:** The user can enter data via an input field. The Input field is part of a form.

Creating tables

Tables are often used to design HTML pages.
You can define tables, for example, to display data in tables or to distribute texts and graphics more appealingly on the screen.

Table 3-3

	HTML code	Notes

1.	`<table>`
	
	`</table>`
		The basis is the <table> HTML tag.
		Every table starts like this and ends with the corresponding END tag.
		It is important that they are both set.

2.	`<tr>`
	
	`</tr>`
		Creating the table **row**.
		"tr" stands for "table row".

3.	`<th>`
	
	`</th>`
		Create table **header**.
		"th" stands for table header

3.2 MWSL (Mini Web Server Language)

	HTML code	Notes
4.	<td> </td>	Creating a table **cell**. "td" stands for "table data".

3.2 MWSL (Mini Web Server Language)

If the "HTML pages" option has been activated on the operator panel, the integrated web server is automatically started when starting the project or the operator panel.

The web server enables you to access the tag area of the operator panel via the integrated script language "MWSL" (Mini Web Server Language).

The syntax of the "MWSL" language is similar to the syntax of Javascript.
The script is executed on the server (operator panel).

Example for accessing the process tags:
<BODY >

....

Tag value: <MWSL><!-- write(GetVar("Tag_01")); --></MWSL>

....

</BODY>

Note	In HTML, the "<!-- Text -->" instruction has the effect that the text within the instruction is interpreted as a comment. Due to the preceding instruction "<MWSL>", the operator panel detects that the text is not a comment but an instruction to access the tag area.

3.3 CSS (Cascading Style Sheets)

CSS is a text-based design language for defining the appearance (design) of an HTML page.

The particularity of CSS is that by outsourcing the formatting to a separate file, the formatting is valid for **all** created HTML pages. Accordingly, changes in the "design" are only made in **one** central place.

Linking the individual formats to the CSS file is performed by the **Class=""** instruction.

Basic structure of the CSS file

In Chapter "Structure of HTML tags" you learnt how HTML instructions are structured.

Example:
<td> (introduces a normal data cell)

In the CSS file the "td" instruction is assigned to any "reference name". Between the instruction and the reference name there is dot. The assigned formats are listed within a "curly bracket". The individual formatting specifications are each separated with a semicolon ";".

Example:
td**.**View1 **{**

> format 1;
>
> format 2;
>
> format 3;
>
> **}**

For this application, the templates also used in the standard web pages of the operator panels were employed. Commissioning the application does not require any further knowledge on the topic of CSS.

If you still wish to create your own design for your user-defined web pages, plenty of information is available on the internet.

3.4 JS (JavaScript)

JavaScript is a script language used for dynamizing the static HTML pages. As opposed to MWSL, JS is a client-side programming language; that is, the JS programs are executed in the web browser.

JS can be used for evaluating the user interactions and changing the contents.

Use the HMTL tag "<script> ... </script>" to integrate JS commands into an HMTL file.

If you wish to use many JS functions in different HMTL pages as well, it is recommended to create a separate JS file and reference it in the HMTL file. Use the command
"*<script src="JSFile.js" type="text/javascript"> </script>*"
to integrate the JS file named "JSFile.js" into your HTML page.

4 Mode of Operation

The functionality of the project contained in this entry mainly corresponds to the functions in "Getting started SIMATIC S7-1500". Should you require further information, please refer to the entry in \3\.
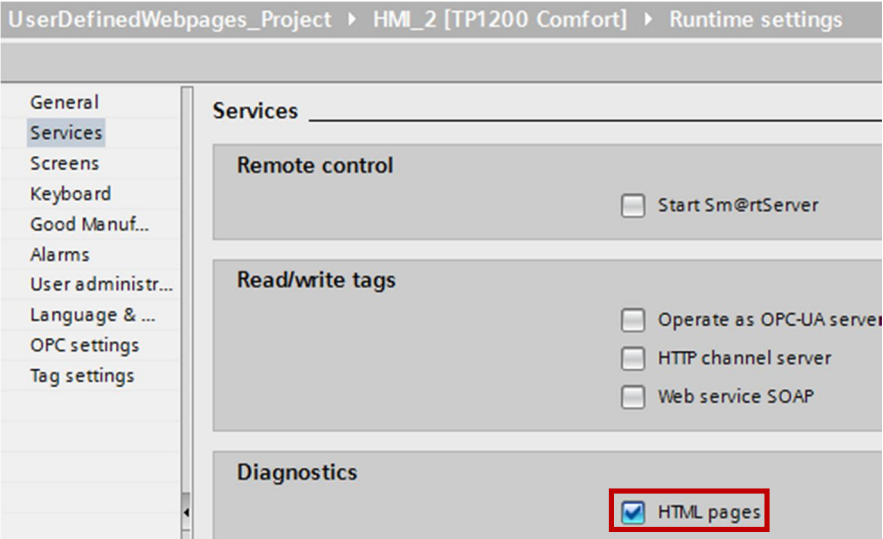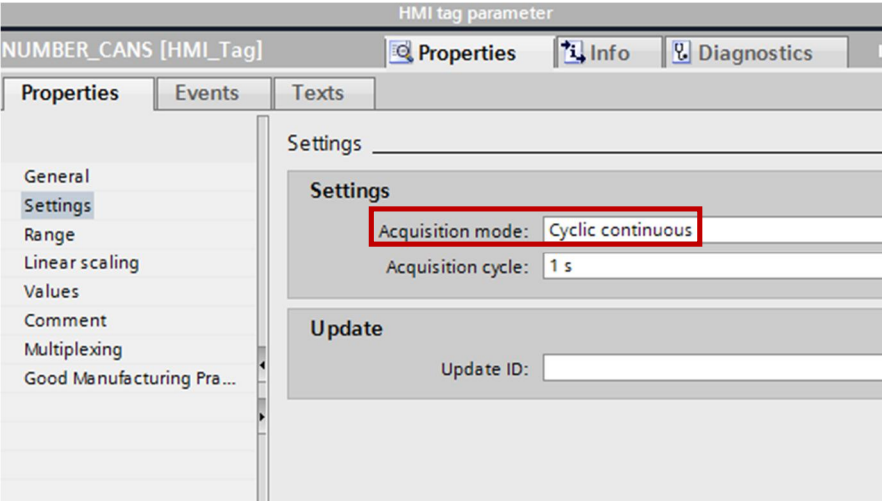
Some predefined standard web pages for your operator panel are already contained in the delivery scope of WinCC (TIA Portal). More information about the functionality of the standard web pages is available in the "Integrated web pages" document which you can download on the download page of this document.

The user-defined web page contained in the entry can be used to remotely access some of the process data. You have a clear view of the current filling level of the individual tanks as well as the current production status.

5 Configuration and Settings

5.1 WinCC (TIA Portal) project

Table 5-1

No.	Action
1.	In order to activate the user-defined web pages in your project, open the "Runtime settings" of the operator panel. Open the "Services" menu and activate the "HTML pages" checkbox.
2.	By default, the value of a process tag is only continuously updated if it is shown on the display. To receive current data when accessing via the web server of the operator panel, the acquisition mode of the respective tag must be changed. To do so, open the HMI tag table with the respective tags. Open the properties of the tag, and in "Settings" you change the acquisition mode to "Cyclic continuous".

5.2 User-defined web page

General

First of all, some general adjustments are necessary, such as inserting links to your own web page or adjusting the title.

Table 5-2

No.	Action
1.	Copy the "WebContent" folder from the "...\Siemens\Automation\Portal V13\Data\Hmi\Transfer\13.0.1\CP..." directory into any directory on your programming device. The exact path depends on your operator panel. For a "4" Comfort Panel you open the "CP_4" folder. For the 7 - 12 inch devices, you open the "CP_7_9" folder, and for the 15 - 22 inch devices the "CP_15_19_22" folder.
2.	Unzip the "WebContent.zip" archive with the respective program (e.g. "7zip").
3.	Copy the web page "start.html" and name it "ProcessData.html".
4.	By default, the web pages are write-protected. Copy the web pages to which you want to insert a link to the new user-defined web pages so you can edit them.
5.	Open the web pages with your editing program and insert a link to the newly created web pages. Enter the name of the web page in 1. In 2 you enter the text to be displayed as a link.
6.	Open the newly created web page with your editing program.
7.	Replace the title in row 9 with the title to be displayed in the title bar of the browser window when calling the web page. `9 <title>Process Data Overview</title>`

5.2 User-defined web page

No.	Action
8.	Copy the link of the start page (row 69-76) and insert it elsewhere. In 1 you enter the text to be displayed as a link. ```html\n<tr>\n <td width="8"></td>\n <td width="7"><img src="/Images/NaviPfeilweiss.gif"\n alt="" border="0"></td>\n <td width="101" class="ad_nav_link_selected">\n Process Data \n </td>\n</tr>\n``` ① (pointing to "Process Data")
9.	Replace the link to the start page (row 69-76) with the link given in the figure below. ```html\n<tr>\n <td width="8"></td>\n <td width="7"><img src="/Images/NaviPfeilblau.gif"\n alt="" border="0"></td>\n <td width="101" class="ad_nav_link">\n Start page</td>\n</tr>\n```
10.	Replace the text in row 132 with the header to be displayed on the web page. ```html\n132 <td height="46" valign="top" class="ad_headline2">Process Data Overview</td>\n```
11.	When the page is called, the user authorization "RuntimeAccess" for the logged in user is verified in order to protect the process data. Otherwise, a page appears which prompts the user to log on with the appropriate authorization. This requires that die following JavaScript function is inserted at the beginning of the web page. ```javascript\n13 <script type=text/javascript>\n14\n15 onload = function()\n16 {\n17 if (!:=ShareRealm("RuntimeAccess"):)\n18 {\n19 location.href =\n20 "Templates/Loginpage.html?Redirection=/wwwSiemens&Realm=RuntimeAccess";\n21 return false;\n22 }\n23 }\n24 </script>\n```
12.	Delete the contents of row 139 to 161. The actual content of the HTML page is now entered here.

5.2 User-defined web page

Overview of the production status

For a clear representation of the production status, the status is displayed via a picture (red and green status display). Additionally, the status is output via a field if your browser suppresses the display of pictures.

Figure 5-1

Table 5-3

No.	Action
1.	To separate the area visually from the other areas, a separate paragraph is defined with ... .
2.	The text between and is formatted as a header in the browser view.
3.	For display, a table without frame lines (border="0") is crated. This defines the area in which the contents are displayed.
4.	Since the status is output via an input field in addition to the picture, the area must be defined as "form".
5.	The picture is integrated with <img...>. *name="IMG_ProductionState"* gives the picture the unique name it can be addressed with in the script. *src="Image/StatusON.png"* specifies a standard picture. During runtime, the path is adjusted depending on the respective status tag (see step 7). *alt="ProductionState"* defines a text that appears if the picture cannot be displayed. This is the case, for example, when your browser suppresses the representation of pictures. *width="20px"* assigns a defined width to the picture.
6.	<input..> defines an input field. The field can receive a unique reference with *name="Input_ProductionState"* and i*d="ID_ProductionState"*. With *type="text"* you specify the type of the input field. With *value="<MWSL>WriteVar("HMI_ProductionState");</MWSL>"* you write the value of HMI tag "HMI_ProductionState" into this field. *readonly* specifies that the value of this field cannot be modified.
7.	The integrated JavaScript function reads the value of the "ID_ProductionState" field and changes the path of the "IMG_ProductionState" field depending on the tag.

5.2 User-defined web page

Representation of the SOLL/IST (setpoint/actual) values

The SOLL/IST values of the currently active production, as well as the overall production are displayed via a table. Furthermore, you can adjust the overall SOLL (setpoint) value if necessary and inform the staff on site.

Figure 5-2

```
190  <p>                                                                    ①
191      <table class="sph_table" width="400" style="table-layout:fixed">   ②
192          <tr>
193              <th class="sph_th"> </th>
194              <th class="sph_th"> Ist </th>
195              <th class="sph_th"> Soll </th>
196          </tr>
197          <tr>
198              <td class="sph_td"> Aktuell </td>
199   ③         <td class="sph_td">  <MWSL><!-- write(GetVar("FILLING_COUNTER")); --></MWSL>  </td>
200              <td class="sph_td">  <MWSL><!-- write(GetVar("NUMBER_CANS")); --></MWSL>  </td>
201          </tr>
202
203          <form>
204          <tr>
205              <td> Gesamt </td>
206              <td>  <MWSL><!-- write(GetVar("HMI_Number_Cans_IST")); --></MWSL>  </td>
207   ④         <td>  <input name="Input_SOLLWERT" type="text"
208              value="<MWSL>WriteVar(HMI_Number_Cans_SOLL);</MWSL>"> </td>
209          </tr>
210          </form>
211      </table>
212  </p>                                                                    ①
```

Table 5-4

No.	Action
1.	To separate the area visually from the other areas, a separate paragraph is defined with ... .
2.	A table is defined for displaying the values. The layout of this table is specified in the predefined CSS file "Miniweb.css". The class "sph_table", "sph_th" or "sph_td" is therefore assigned to the individual elements (table / td / th).
3.	The value of HMI tag "Tag name" is displayed on the HTML page via *<MWSL><!--write(GetVar("Tag name")); --></MWSL>*.
4.	<input..> defines an input field. The field can receive a unique reference with *name="Input_SOLLWERT"*. With *type="text"* you specify the type of the input field. The value of HMI tag "HMI_Number_Cans_SOLL" is written into this field with *value="<MWSL>WriteVar("HMI_Number_Cans_SOLL");</MWSL>"*. To specify the tag value via the web page as well requires inserting the following code at the beginning of the web page: <pre>24 <MWSL>
25   <!--
26   if(ExistVariable("Input_DesiredValue", "URL"))
27   {SetVar("HMI_Number_Cans_Desired", GetVar("Input_DesiredValue", "URL"));}
28   -->
29 </MWSL></pre> MWSL interprets the "Input_SOLLWERT" field as a tag. A "URL" (or "MWSL2_URL" for version 13.0.0.x operator panels) expresses that this "tag" is located on the web page.   The value of HMI tag "HMI_Number_Cans_SOLL" is overwritten with the value of the input field.

Representation of the filling levels

The filling levels of the paint tanks are output via a table.

Figure 5-3

```
214 <p>
215 <h1>Tank Level</h1>
216 <table class="sph_table" width="400" style="table-layout:fixed">
217     <tr>
218         <th class="sph_th"> Tank </th>
219         <th class="sph_th"> Level </th>
220     </tr>
221     <tr>
222         <td class="sph_td"> Cyan </td>
223         <td class="sph_td"> <MWSL><!-- write(GetVar("Filling_FillingLevel_CMYK_C")); --></MWSL> </td>
224     </tr>
225     <tr>
226         <td class="sph_td"> Black </td>
227         <td class="sph_td"> <MWSL><!-- write(GetVar("Filling_FillingLevel_CMYK_K")); --></MWSL>  </td>
228     </tr>
229     <tr>
230         <td class="sph_td"> Magenta </td>
231         <td class="sph_td">  <MWSL><!-- write(GetVar("Filling_FillingLevel_CMYK_M")); --></MWSL>  </td>
232     </tr>
233     <tr>
234         <td class="sph_td"> Yellow </td>
235         <td class="sph_td">  <MWSL><!-- write(GetVar("Filling_FillingLevel_CMYK_Y")); --></MWSL>  </td>
236     </tr>
237 </table>
238 </p>
```

The used syntax is described as before.

6 Installation and Commissioning

6.1 Installing the hardware

Commissioning the application example requires a SIMATIC HMI Comfort Panel that has to be connected to the 24 V supply voltage. In addition, the SIMATIC HMI Comfort Panel has to be integrated into a network with the programming device.

6.2 Installation of the software

To use this application example, at least software version SIMATIC STEP7 Professional V13 SP1 and SIMATIC WinCC Comfort V13 SP1, as well as a text or HTML editor has to be installed on your programming device.
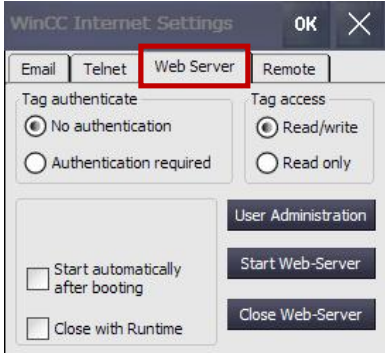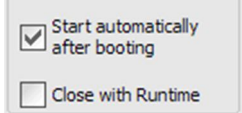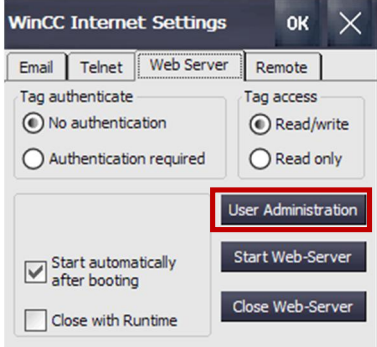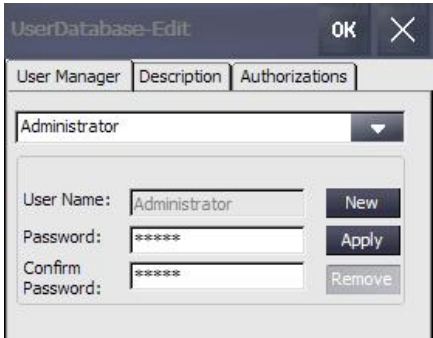
6.3 Commissioning

Table 6-1

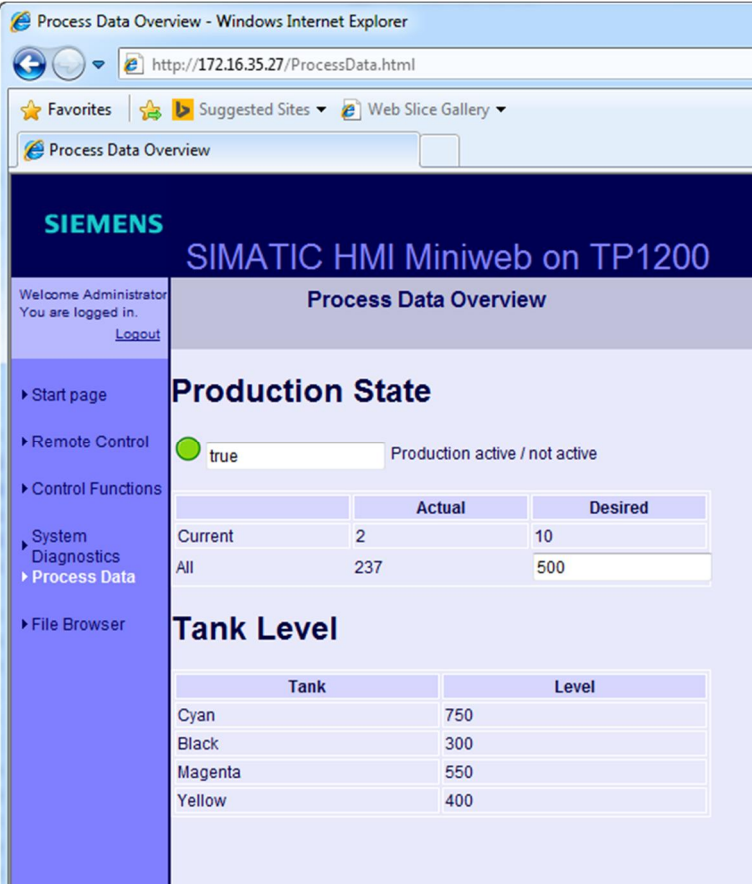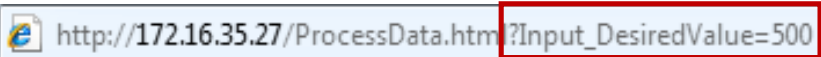| No.	Action
1.	Download the "109476153_Remote_Panels_HTML_Project" example project along with the HTML pages and unzip.
2.	Open the project with WinCC (TIA Portal).
3.	Open directory "...\Siemens\Automation\Portal V13\Data\Hmi\Transfer\13.0.1\CP_7_9" and secure the contained folder "WebContent.zip" into a secure storage path.
4.	Copy the downloaded folder "WebContent.zip" contained in this application into the directory. This also automatically transfers the created user-defined web pages onto the operator panel during a project download.
5.	Download the project into operator panel and controller.
6.	Open Control Panel on the operator panel by clicking on "Settings" in the Start Center.
7.	Double-click on "WinCC Internet Settings" in the Control Panel of the operator panel. WinCC Intern...

6.3 Commissioning

No.	Action
8.	Open the "Web server" tab.
9.	Activate "Start automatically after booting", if the web server shall start together with the operator panel and not later together with Runtime. Activate "Close with Runtime", if the web server shall be terminated together with the runtime.
10.	Click on the "User Administration" button.
11.	Assign an administrator password and create further users. In the "Authorizations" tab you specify the authorizations of the users. Operating the "ProcessData" web page requires "RuntimeAccess" authorization.
12.	Start Runtime.

7 Operating the Application

Table 7-1

No.	Action
1.	Start the Internet Explorer on your PC.
2.	In the address bar you enter the IP address of the operator panel.
3.	Log on with the user name and password of a user with "Runtime Access" authorization.
4.	Open the "Process Data" page. In order to update the displayed values, the page must be reloaded.
5.	If you wish to specify a new setpoint, enter it in the field and acknowledge with "Enter". This value is then displayed in the address bar of the browser window. Before refreshing the page again, delete the "?Input_DesiredValue=..." part from the address bar, since otherwise, the setpoint will be sent again to the operator panel after each refresh.

8 Links & Literature

Table 8-1

	Topic	Title
\1\	Siemens Industry Online Support	http://support.automation.siemens.com
\2\	Download page of the entry	https://support.industry.siemens.com/cs/wwde/109476153
\3\	Getting Started S7-1500	https://support.industry.siemens.com/cs/ww/en/view/78027451

9 History

Table 9-1

Version	Date	Modifications
V1.0	05/2015	First version