

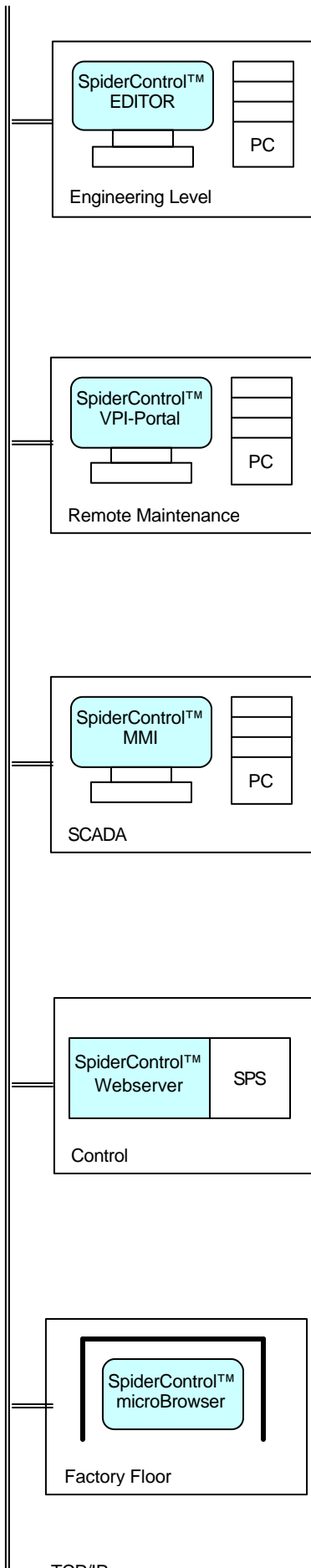


SpiderControl™ EDITOR

**PLC Edition
Version 4.0
SIMATIC S7**



iniNet AG. Erfolgreiche Kunden
E-mail: info@ininet.ch, www.ininet.ch, Tel: +41 61-716 96 26



TCP/IP

Overview

Thank you for choosing SpiderControl™ EDITOR. SpiderControl™ is an advanced technology for Web-based graphical Man-Machine Interfaces (MMI's), with an easy to use development tool. SpiderControl™ is based on open Java-Applet standards.

Web-based MMI's can be accessed from everywhere. They can be used on the factory floor by operator panels, from within the intranet by any PC and a standard browser as well as from remote via modems, WLAN, Internet Router, etc.

Several MMI's from different PLC's/Webserver can be made visible on the Internet with a SpiderControl™ VPI-Portal.

SpiderControl™ is available for most of the commonly used PLC's as well as for custom-made microcontroller platforms in the embedded control area. If you are interested in integrating SpiderControl™ on your own hardware, please contact us.

The SpiderControl™ EDITOR allows of a pure graphical design of MMI's. The communication between the Applet and the PLC is already solved, no manual coding is necessary at all.

A regular WEB-Browser with a JAVA VM 1.1 support, e.g. MS Internet Explorer, Mozilla, Opera,... is sufficient. A multitude of panels and operating systems can be used.

The browser communicates with the Webserver through the TCP/IP – Protocol which can be routed over different medias like Ethernet, Modem, Wireless LAN.

As the entire datas of the MMIs were kept on the PLC as JAVA-Applet, no version mismatch or code-redundancy will arise anymore between control- and visualisation-software. Developing and maintenance will be reduced and you will get a real object-oriented design.

The visualisation-software can be updated without change of the PLC-software and vice versa.

For on-site "Operating Panels" we are able to supply you with a OEM SpiderControl™ microBrowser with which you can easy build microcontroller-driven panel solutions. Contact us for further information.

All the software has been developed by iniNet Solutions GmbH and can therefor be ported to almost any target system.



SpiderControl™ EDITOR PLC Edition

No programming skills are necessary to create a MMI with SpiderControl™ EDITOR PLC Edition. The workflow consists of three steps:

1. Drawing of the MMI-views with the help of basic graphical symbols.
They can be linked to process points (PPOs) of your PLC to display and change values or color switching for alarms. The PPOs were based on symbolical names which are known only within the MMI-views and which were linked to the real PLC-datas in the next step. The HTML-startpage is created by the Editor and can be tested with a PC-Browser.
2. In the second step the symbolical names have to be linked with the real PLC process points (Inputs, Outputs, Flags, ..). This is done with the help of the `InitPPO` dialog.
3. The entire project has to be transferred into the PLC. This step varies for different PLCs and is described in the Chapter *Initialising Process Points*.

File Formats

This chapter explains all files which are necessary to create a SpiderControl™ based MMI.

The browser addresses a HTML-Page with

http://<target_system_ip_adr.>/<your_project>.html.

This file is generated through the SpiderControl™ EDITOR (Menu: Project->Generate HTML...). It contains a reference to the JAVA.-Applet `IMasterSimatic.jar` which will build up the views in your browser. The views itself were defined in the `*.teq` files.

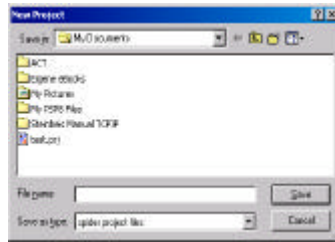
To get a web-based visualisation, the following files have to be copied to your target system:

- The entry page `<your_projectname>.html`
- The JAVA-Applet `IMasterSimatic.jar`
- All Views (`*.teq` – files)
- All additional custom HTML-files (i.e: helppages, ..)
- All used image ressources.
- The variable-initialising files `<your_projectname >.tcr` and `<your_ projectname >.itq`

Getting started

After starting SpiderControl™ EDITOR the main window appears. It is divided into two subwindows. In the left one, all referenced files used in the current project were displayed. In the right subwindow the views can be edited.

Project creation



Choose the menu command `File->New Project` to get a standard Windows file chooser dialog. Create a new directory, go into it and indicate the project name as file name and click on `Save`.

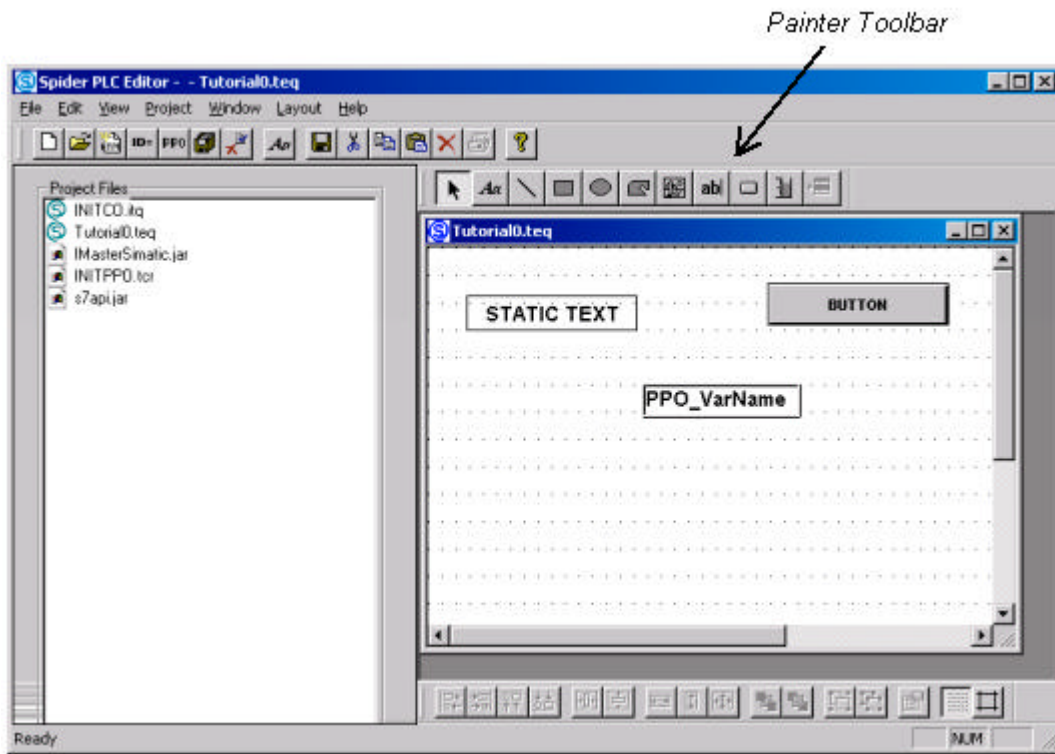
A projectfile `*.prj` and all additional necessary files were created and displayed in the left subwindow.

View Creation

Now we want to create our first view. Choose the menu command `Project->Add->New...`. The Editor will propose a filename for the new view which consists out of the projectname, an incrementing view counter and the suffix `*.teq`. The file name can be changed by the user, if so desired.

`<your_projectname0>.teq`

Now you can start inserting the painters into the view. All painters can be chosen through the `painter's toolbar`. A detailed description of these painters can be found in the chapter „*Painter's Reference*“

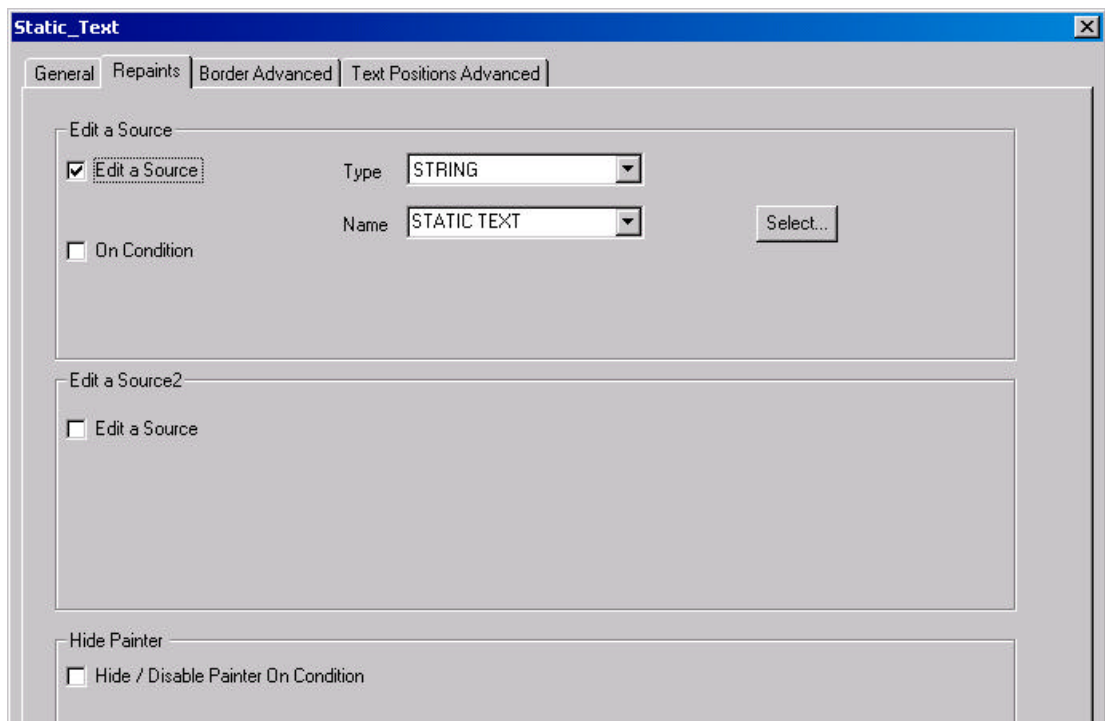


Adding a Static Text or Reading a Process Variable



In the tutorial, a 'Static Text' painter object was selected. The object can be placed in the view by pressing the left mouse button to mark the left most top position of the text, then keeping the mouse down and dragging towards the rightmost down position, then releasing the button. The user now might want to change the text. This is done by double-clicking the text object. This will open the object properties dialog box. This dialog box contains different tabs, which allow the user to define all properties of this object. The first tab 'General' contains options, which are applicable to any object, e.g. position, size, colors and font. The second tab 'Repaints' lets the user define what the object will show on the screen. In the tutorial, a string containing the text 'Welcome To The Tutorial' should be shown. To do so, the 'Edit a Source' checkbox should be selected, the 'Type' in the selection on the left should be set to 'STRING' and the desired text should be entered in the name field. The user can now press the 'OK' button to close the dialog.

Alternatively, the type of a "Static text" - painter can be set to HTML_TAG, CONTAINER or PPO. If set to PPO the "Static text" - painter can be used to display a process point of your PLC. If you want to read and edit a process point, then you have to use the "Editbox" – painter which is described in the next chapter "Editing a Process Variable".



Edit a Process point (PPO)

In the next step, an 'Editbox' element will be added. This object allows reading and modifying a process point from the target system inside the view. To do so, the user should press the 'Edit Box' button in the toolbox and placing the object the same way like the 'Static Text' object. Then, a double click on the object will open a dialog box, where the 'Repaints' tab should be selected.

The 'Edit a Source' checkbox must be selected and the 'Type' choice should be set to 'PPO', where 'PPO' stands for 'Process Point', indicating that a variable from the target system should be used. In the 'Name' edit field you should now enter a symbolic name. A valid name could look like "TempVessel_1". It is not necessary to define the process point that the user wants to use before this step. The 'OK' button will save these settings to the view. The linking of these symbolic names (i.e. "TempVessel_1") to the real process point is later done by using the menu function "Project -> Init PPOs".

Creating a second view

A second view should now be added the same way. This can be done by the menu as presented above, or alternatively, simply by pressing the 'New File' button from the toolbox. The suggested name will be 'Tutorial1' and the file will be added automatically to the project. The user can also add a text object indicating the text 'This is the second view...' to the file.

Navigating between views



The next step will be to add buttons, which allow the user to navigate between the two pages. To do so, the 'Button' object in the toolbar must be selected and the object should be placed in the view 'Tutorial0'. A double-click will open the dialog box again, where the 'Repaints' tab should be selected. On this tab, the user can enter the text that should be displayed on the button. This is done the same way like shown with the 'Static Text' object.

Next, the user should go to the 'Actions' tab, where the 'View Jump' checkbox must be selected. Inside the 'view name' editfield, the user can now enter the name of the view where the button should jump to, e.g. 'Tutorial1'.



A similar button can now be added to the second view 'Tutorial1.teq', where the name in the action tab should refer to 'Tutorial0'.

Alternatively you can activate the checkmark `Back Button` to jump to the last displayed view or you can activate the `URL Jump` to display any website you want.

Creating an Application

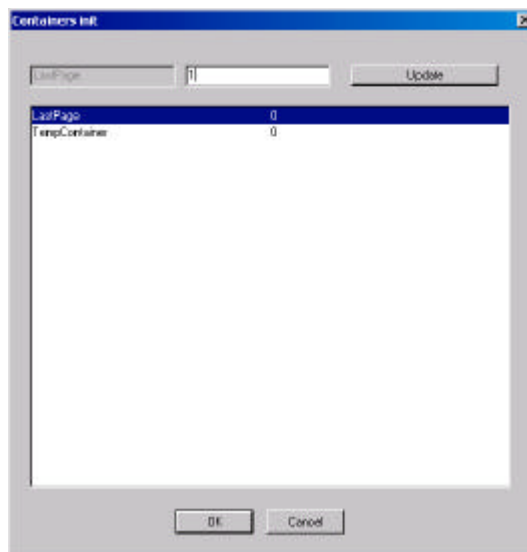
In the next step, the 'HTML' file, which contains the tutorial applet, will be generated. This can be done with the menu 'Project->Generate HTML' or with the respective button from the toolbox. The 'SpiderControl Editor' will suggest the filename `Tutorial.html` and add the file to the project. Then, all files should be saved using the menu 'File->Save All' or by pressing the respective button from the toolbox.

First Test Inside the Browser

For a first test you can open the HTML-Startpage created just before with any PC-Browser (MS Internet Explorer, Netscape Navigator, Mozilla, Firefox, Opera, ...). Of course no PPOs can be displayed yet. Those have to be linked first to your PLC and the whole project has to be transferred to your target system (PLC). This is all described in the chapter `Initialising Process Points`. Also the MMI internal variables named `CONTAINERS` have to be initialised first. This is described in the next chapter.

Initialising Containers

Containers are local variables, which are visible only within the views. With the menucommand Project->Init container... you can open the configuration dialog for containers. There, all containers were listed. By selecting one, its value will be copied into the upper edit field, where it can be altered and saved with the button Update. With OK you can save alle changings and close the configuration dialog. The file <your_projectname>.itq will be updated. This file has also to be transferred to your target system.



Initialising Process Points (PPOs)

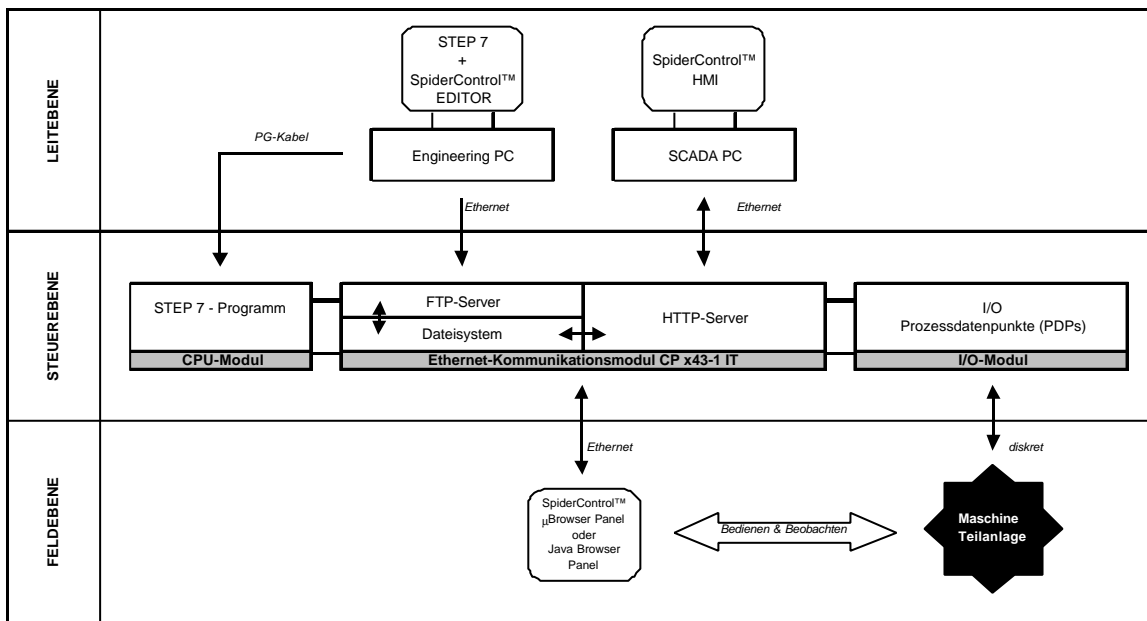
SpiderControl™ SIMATIC Concept

SpiderControl™ has been adapted to the PLCs of type SIMATIC (Siemens). SpiderControl™ SIMATIC is compatible with the series S7/200, S7/300 and S7/400. As ethernet interface, webserver, ftp-server and filesystem, the SIMATIC communication module CP x43-1 IT (x=2,3,4) is used.

The MMI-files developed with SpiderControl™ EDITOR PLC Edition 4 SIMATIC have to be transferred with ftp to the filesystem of the communication module (Flash). The files can be HTML-pages, textfiles, JAVA-Classes, GIF-Images as well as SpiderControl™ views (*.teq) and variables initialisation files (*.itq, *.tcr).

The SpiderControl™ JAVA-Applet called IMasterSimatic.jar is using the Siemens JAVA-Beans of the communication module CP x43-1 IT.

Through these JAVA-Beans you can access process points (PPOs) like Inputs, Outputs, Flags and Datablocks. Different data formats were supported (BIT, BYTE, WORD, REAL, STRING).



Within the SpiderControl™ views the process points were not addressed in an absolute way but with SpiderControl™ internal symbolical names. The link between these names and the real process points is configured with the help of the InitPPO dialog. This dialog creates a file called <your_projectname>.tcr, which defines the connection between STEP7-Software and SpiderControl™ MMI.

InitPPO SIMATIC Dialog

After creating the different views of your MMI, you have to link the symbolical references used within the painters with the real process points of your SIMATIC PLC. This is done with the following dialog which you can open with the menucommand `Project->InitPPO`.

PPO Name	PPO Min Val	PPO Max Val	PPO format Val	PPO unit Val	PPO Read type	S7 Var Type
					individual	S7
S7 Var Resp Factor	S7 Var Mem Area	S7 Var Sub Area	S7 Var Byte offset	S7 Var Bit offset	Update	
	M		0	0	0	
HeiztemP	2	2	1	129	0	0
Befehl_Aus	2	1	1	130	0	0
Befehl_Ein	2	1	1	130	0	0
Betriebsstunden	2	2	1	129	0	0
Energieverbrauch	2	2	1	129	0	0
SollTemp	2	2	1	129	0	0
Stellr	2	2	1	129	0	0
Stromaufnahme	2	2	1	129	0	0
WTN1	2	2	1	129	0	0

All used symbolical references were listed here. Select one with the mouse and its parameters were copied into the upper part of the dialog. There you can edit the individual parameters and save them by clicking `Update`.

PPO min Val & PPO max Val

For Edit Field painters you can specify here the lower and upper limit of a user input. By exceeding this limits SpiderControl™ reports an error.

PPO format Val

Can be `int` or `string`. Will determine the formatting of a displayed process point.

PPO Unit Val

The unit of a process point. Any unit can be chosen as a string. When using this variable in an edit field or static text, the unit will automatically be added.

PPO Read Type

Can be `individual` or `MB_Block`. With `individual` each process point will be requested by its own from the target system. With

MB_Block all process points were requested together at once. This will save communication overhead.

S7 Var Type

Specifies the SIMATIC S7 datatype as described in the documentation of the SIMATIC communication module (Keyword: JAVA-Beans, S7Put/GetApplets).

S7 Rep. Faktor

String length of S7 Strings.

S7 Var Mem Area

Specify if you want to access Input, Outputs, Flags or Datablocks.

S7 Var Sub Area

For datablocks you have to indicate the datablock number. For S7/200 there is only DB1.

S7 Var Byte Offset

The address of a flag, datablock-entry offset, input or output.

S7 Var Bit Offset

To access single bits.

Filetransfer to SIMATIC S7

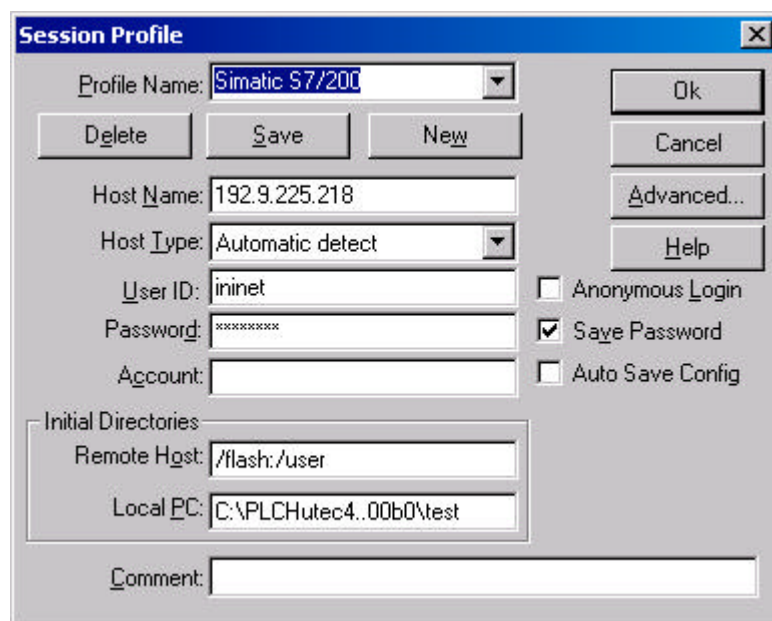
All files created or referenced through a SpiderControl™ MMI like HTML-pages, GIF-images, *.teq-files, IMasterSimatic.jar etc. reside in the project directory and have to be transferred to the communication module CPx34-IT.

This is done with the ftp-server of the communication module. For configuring the ftp-server there exists an "Internet Assistant" in the developing environment of SIMATIC Step 7 (menu „Extras“).

With the help of this „Internet-Assistent“ you can assign an IP-address and subnet mask to the communication module. Additionally you have to configure an administrator account to be able to log into the ftp-server.

For the file transfer itself you can use any available FTP-tool for Windows™ like i.e. WS_FTP, which comes with the SpiderControl™ EDITOR package.

After startup of WS_FTP the following dialog appears:



As „Host Name“ you have to indicate the IP-adress of your communication module. „User ID“ and „Password“ correspond to its Administrator-account. As initial directories you can specify for “Remote Host” the path of the filesystem of your communication module. Use “/flash/user” as default MMI root directory. For “Local PC” you have to indicate the path to your MMI project directory on your PC. It is useful to specify a “Profile Name” for these settings and save the profile with the button “Save”.

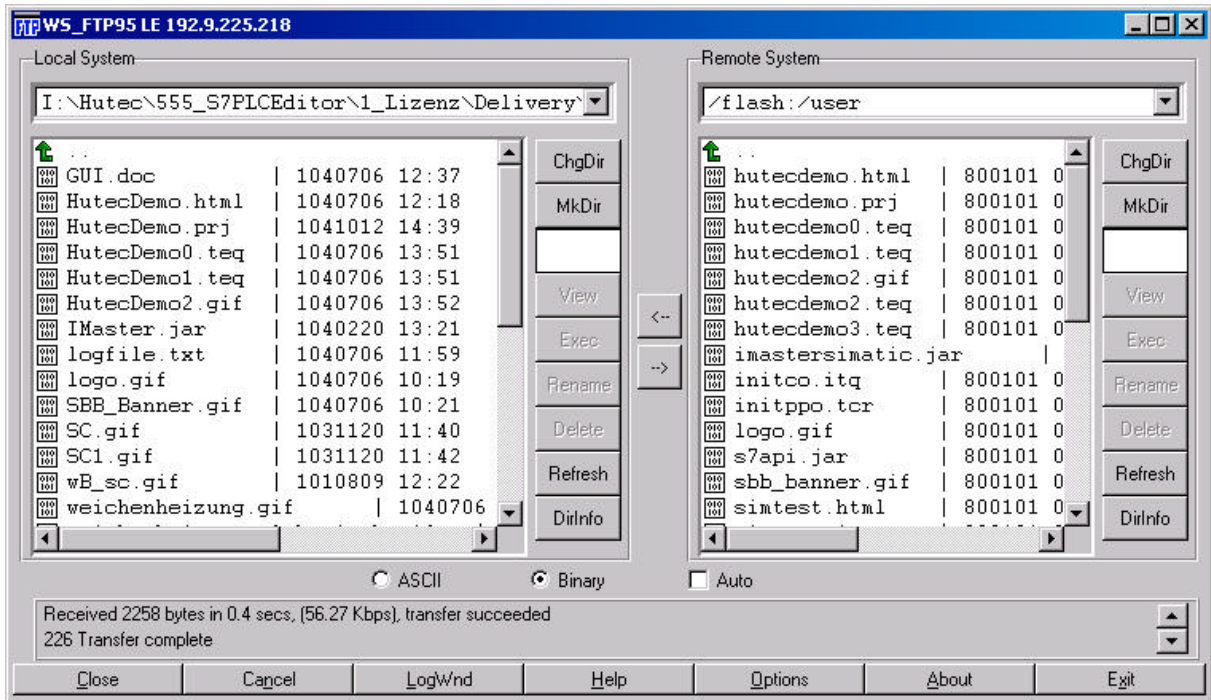
With “Ok” you leave this dialog and the main window of WS_FTP appears. In the left area the source directory of your MMI project on your PC is listed. At the right side the content of the Flashdisk of the

communication module is listed. In the middle there is the button ">" which will start the transfer process.

After a successful transfer of your project you can start it. Open a PC-Browser (i.e. MS Explorer) and type

http://<ip-adr.-comm.-module>/user/<your_projectname>.html.

i.e: <http://192.9.225.218/user/hutecdemo.html>.



Updating Values from the Target

Each view contains a number of references to 'process points' (PPOs) of the PCD System. In order to achieve a good performance in updating these values, the painters do not ask the webserver to update each value separately. When the applet is initializing, it collects all 'process points' that are referenced, makes a list and sends this list to the server. This request informs the server, the values of which have to be maintained for the current view. When this view needs to update itself, it asks for the updated request list, which will be returned by the server. This list contains all values that are actually needed.

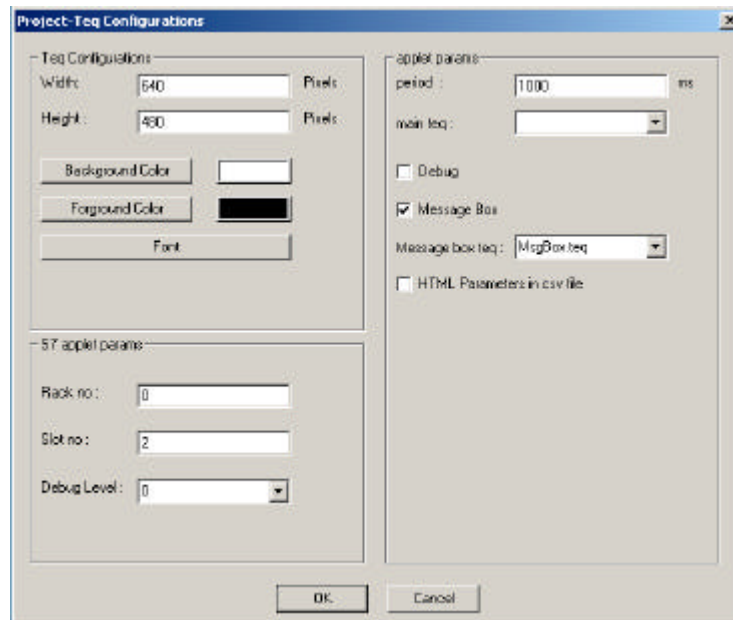
Single request comprising all views

Only one request list is generated for the applet; it covers all views that are included. The advantages are:

- less files and therefore less overhead on the embedded file system.
- instant initialization, because all process points are already known to the server.

Project Settings

This dialog allows the user to set the general project settings.



Teq Configurations

These settings will be applied whenever the user creates a new view. Existing views are not affected.

S7 Applet Params

These Parameters are handed over to the S7Bean by Siemens, which handles the communication with the CP-Module. Please refer to the Simatic documentation for details.

Update Period

Sets the period to poll for process points from the Web-Server.

Main Teq

The user should indicate the name of the view which should be first opened after initialisation of the applet.

Message Box

SpiderControl supports a message-box concept to inform the user about problems during the operation of the MMI. Such problems may include communication breakdown, etc. These information can be viewed by the 'Message Box', which can be modified by the user. If the checkbox is not selected, no messaging will be performed during runtime. If selected, a button will appear on the actual view, indicating errors. When the user presses the button, the respective 'Message Box' will appear.

Conditional Actions

For certain properties of the painters you can indicate a condition, which has to be true to make the property active. Examples are the “Edit a Source” property in the “Repaints” tab of the property dialog or the “Border Advanced” tab, where you can set the color of a painter depending of a condition. The format of these conditions are always the same and were explained here.

First you choose the type of the variable of which the condition shall depend. Choose “PPO” for real process points or “CONTAINER” for HMI-internal variables. Then indicate the name of the variable in the field “Name”. Now you can set the value int the field “Value” to which the variable shall be compared to and the type of comparison which can be:

- != not equal
- < smaller than
- <= smaller or equal than
- == equal
- >= greater or equal than
- > equal than

The screenshot shows a dialog box titled "Button" with a tabbed interface. The "Actions" tab is selected. It contains two sections for configuring "Edit a Source" properties.

Edit a Source

- Edit a Source
- Type:
- On Condition
- Name: - Type: Name:

Edit a Source2

- Edit a Source
- Type:
- On Condition
- Name: - Type: Name: Value:

Multiple views

This section explains the user how to set up an HMI consisting of multiple views. As we are developing for a networked environment, there are certain points that must be considered for the design of such a solution in order to obtain the best results in terms of performance and memory footprint.

In general, there are two different ways of how to change from one view to another:

Multiple views in one applet

The user requests one HTML page with the browser. This page (<your projectname>.html) contains one applet (Imaster.jar), which references multiple views (stored as *.teq files). This is the fastest way to switch from one view to another. All views and the corresponding applet code are loaded and started when loading the HTML file <your projectname>.html into the browser. This file is created from the project menu: 'Project->Generate HTML'.

```
<!-- HTML FILE GENERATED FROM PLC EDITOR by ININET Solutions  
GmbH on Wednesday, July 14, 2004 -->  
<HTML>  
<HEAD>  
    <META HTTP-EQUIV="Content-Type"  
CONTENT="text/html;CHARSET=iso-8859-1">  
    <META NAME="GENERATOR" CONTENT="IniNet SpiderControl  
TM">  
    <TITLE>Tutorial</TITLE>  
</HEAD>  
<BODY bgcolor="#FFFFFF">  
<p>  
<APPLET ARCHIVE="IMaster.jar" CODE="IMaster.class"  
ALIGN="BOTTOM" WIDTH="640" HEIGHT="480">  
<PARAM NAME="AppletName" VALUE="Tutorial">  
<PARAM NAME="ProjectName" VALUE="Tutorial">  
<!--PARAM NAME="DrawTeqsAtInit" VALUE="true"-->  
<PARAM NAME="Period" VALUE="1000">  
<PARAM NAME="FirstFileIndex" VALUE="0">  
<PARAM NAME="LastFileIndex" VALUE="2">  
<PARAM NAME="Main" VALUE="0">  
<PARAM NAME="OrderPerView" VALUE="off">  
<PARAM NAME="debug" VALUE="off">  
<PARAM NAME="SpiderHTMLVersion" VALUE="2">  
<PARAM NAME="MainTEQName" VALUE="Tutorial0">  
<PARAM NAME="TEQ0NAME" VALUE="Tutorial0">  
<PARAM NAME="TEQ1NAME" VALUE="Tutorial1">  
<PARAM NAME="TEQ2NAME" VALUE="Tutorial2">
```

Link to another HTML file

It is also possible to link an applet view to another HTML view, which may contain an applet, too. The disadvantage is, of course, that the browser needs a certain time to load and initialise the new page and applet. Nevertheless, this feature can be useful in certain cases like:

- links to help files (stored as HTML)
- links to HMI's from other webservers, to create 'virtually' one single HMI
- links to other applets or pages that are not created with the SpiderControl Editor.

Multilingual MMIs

To support multilingual HMI's there is a data source type called "HTML-TAG" which you can use instead of the type "STRING" in the "Edit a Source" region of the property dialog tab "Repaints". Whereas the type "STRING" will always output the static string indicated in the "Name" field of the "Edit a Source" region, the type "HTML-TAG" will do the same if you do not specify more. With "HTML-TAG" defined strings can be altered to different languages by including manually the following line into your projects HTML-entry page <your_projectname>.html:

```
<PARAM NAME = "<string defined in the name field>" VALUE = "new string name">
```

Example:

In the "Repaints" tab of the property dialog you specify:

Edit a Source: Type = STRING NAME = "Off"

which will display the painter with the static text "Off".



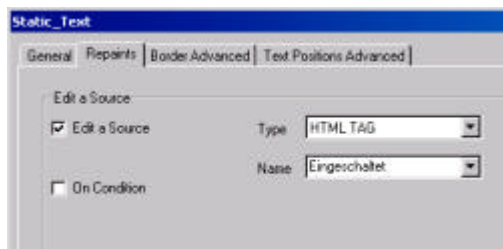
If you now alternatively specify

Edit a Source: Type = HTML-TAG NAME = "Off"

You can add the following line to your "your_projectname>.html" file:

```
<PARAM NAME = "Off"                  VALUE = "Ausgeschaltet">
```

which will replace the string "Off" with "Ausgeschaltet" (german) for each painters using the HTML-TAG "Off". By creating different "your_projectname>.html" files for each language you can create multilingual HMIs.



Painter Gallery Reference Guide

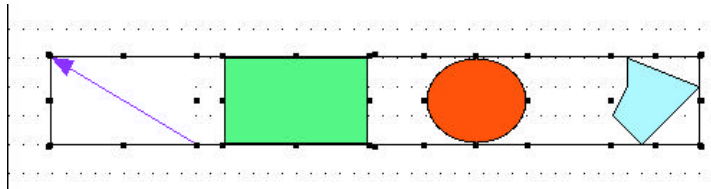
The gallery painters are based on the basic painter types, which are shown inside the 'Painter' toolbox.



Static Painter Objects

Rectangle, Ellipse, Polygone

These painters are 'static' because they do not have a direct link to a process point or a container variable. Still, they can be linked to a variable with a condition, e.g. to change the color or to disappear.



Line



The line painter has the same properties as rectangle, ellipse and polygon. Additionally you can specify in its own properties dialog tab, if you want to have the line with an arrow at the beginning, the end or at both sides of the line.

Image



This Image view painter displays and updates a GIF or XBM formatted image file. The user can define up to three different images that are displayed as a function of a condition. The condition input is read from the web-server (as a process point) and compared to a value, which must be defined as a private property. More than one image can be displayed at the same time if the images have a transparent background. The conditions can refer to the same or different process points. The images will be scaled to the size indicated in the common properties. The image files could reside on the web-server this applet is loaded from or in the local directory `..\webpages`. The interior color will be visible as a background if the images have a transparent background.



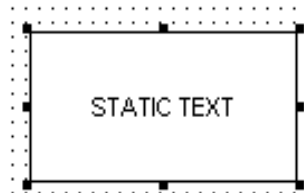
Dynamic Painter Objects

The following painters have an interactive behaviour and their properties can be edited in property tab (double click on painter). Some of these painters have the capability to write or modify process points on the server. Any modification, which is done by the user, will be written instantaneously to the target. It can take a short time until the new value is updated in the browser view because the new value has to be reloaded from the target first. This mechanism makes sure that the values, which are displayed inside the browser, are indeed identical to the corresponding variables in the PCD System.

Static text



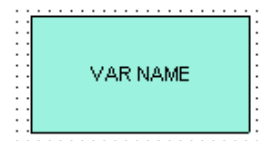
The 'static text' object can be used to display either regular text defined as a string, but also dynamic data from a container or PPO. The text may optionally have a frame, but cannot be edited. There are specific options to right- or left align, etc.



Edit Field



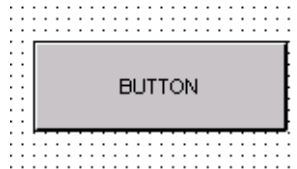
This painter displays and updates a process point from the webserver it was loaded from. When you click on it inside the browser the value can be modified and be written back to the target by pressing "Enter". The process point can be both, text or a number. The 'Text Position Advanced' boarder advanced is available, see 'Static text' painter for the description.



Button



You can change the displayed string on the button in the property window (double click on the painter) and 'Edit a Source' in the 'Repaints' tab true and type your string in the field 'Name'.



With the function URL Jump causes the browser to jump to a HTML page. The page to jump to is indicated with an URL. This URL can be located on the same web-server as well as on any other webserver, which is currently available. To jump to a local HTML page, it is sufficient to indicate the (path and) file name. e.g. 'help.html'. To jump to another server, it is necessary to indicate the complete URL. e.g. 'http://192.9.225.60/help.html' or 'http://www.ininet.ch/index.html'. Set the 'URL Jump' checkbox true and type the URL in the Field 'URL'.

With the function View Jump the browser to jump to another view inside the same applet. The page to jump to is indicated with an integer. This integer refers to the teq file number. For details on the numbering scheme inside an applet, see the section 'Multiple Views' inside this manual. Set the 'View Jump' checkbox true and type the index number of the teq view where the browser should jump to, in the field 'view index'.

Bar Graph



The 'Bar Graph' painter is able to display a process point in a graphical representation. To enable the desired functionality, the user must indicate the range minimum and maximum of the respective process point.

See chapter 'Property window' for more information's about the 'Bargraph Advanced' tab.

