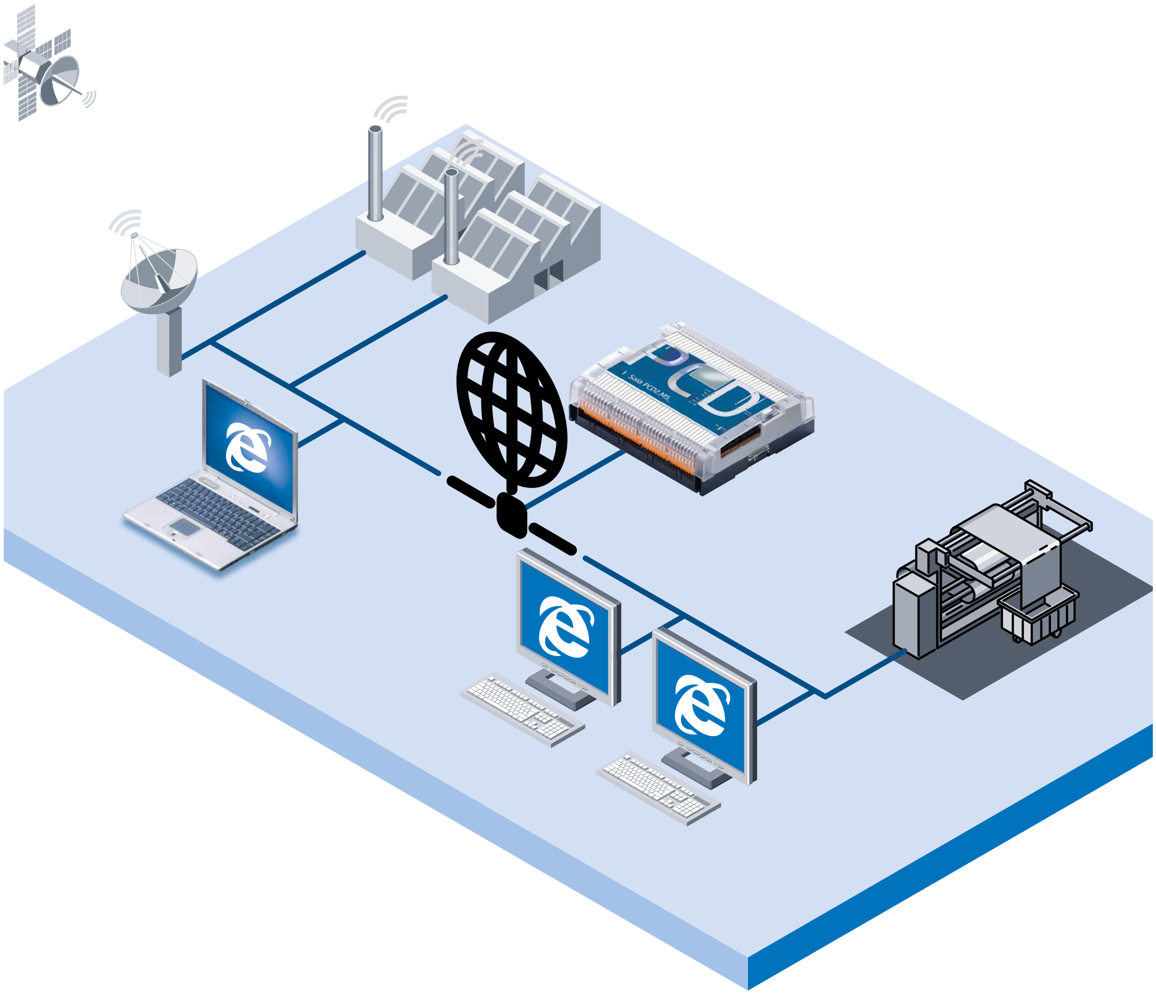
SAIA BURGESS CONTROLS



Manual Web-Server Classic

Document 26/790; Edition EN05 │ 2013-10-18

0 Content

0.1 Document History

Date	Version	Changes	Remarks
2002-10-31	pEN01	-	Preliminary Edition
2003-02-28	EN01	-	Published Edition
2003-05-18	EN02	-	Annex:HTML pages w/o editor, Icons, Doc. conc. etc.
2003-12-31	EN03	-	General revision
2007-03-13	EN03	2007-03-14	Chapter 7 : Renamed the names of the 4 CGI files
2008-06-26 2008-06-26 2010-09-06	EN04	-	Chapter 3.2 and 4.2 : hint to 26/800 Chapter 5.6 and 6.1 deleted Chapter B adapted the addresses
2013-10-18	EN05	-	New logo and new company name

0.2 About this manual

See the section in the appendix in relation to some of the terms, abbreviations and the references used in this manual.

0.3 Source list

Parts of this manual (in particular the Appendix) have been copied from TeamOne’s SELFHTML file.
SELFHTML can be downloaded from Internet under the www address:

http://www.teamone.de/selfhtml/....

0.4 Brands and trademarks

Saia PCD® and Saia PG5®
are registered trademarks of Saia-Burgess Controls AG.

Windows 95/98, Windows NT, Word, Excel, PowerPoint, FrontPage and Micro soft Internet Explorer are registered trademarks of The Microsoft Corporation.

Netscape Navigator is a registered trademark of The Netscape Communications

Technical modifications are based on the current state-of-the-art technology.

Published in Switzerland

1 Introduction

1.1 Minimum requirements for the Web-Server

The following minimum requirements are necessary for the Web-Server functionality:

Saia PCD® System	PCD2.M170, PCD4.M170	PCS1
Saia PCD® FW	010	on request
Saia PCD® serial port	Any one	
Saia PG5®	Version 1.2 or higher (includes the Web-Server)	
PC	• IBM PC or compatible • minimum 133 MHz Pentium with 32MB RAM • Microsoft Windows 95/98/2000/XP or Windows NT 4.0. • Winsock 1.1 API compliant networking package • Telephony Application Programming Interface (TAPI) • Microsoft Internet Explorer 5.5 or Netscape Navigator 6 • Java Machine V1.4 for MonitoRIO	

1.2 Features

Within the Saia PCD®, the integrated Web-Server allows the user to create an own control and monitoring function quick and efficient with the help of HTML pages.

Standard Microsoft programs (such as Word, Excel, PowerPoint or FrontPage) are used to create these control and monitoring functions.

Traditional Web-Servers require a large amount of resources, e.g. in memory or computing capacity. PCs do not have unlimited resources in their processors and they may be fully utilised for controlling machines or processes. Therefore, control technology uses only Web-Servers in the form of additional PCs or expensive add-on components (co-processors).

1.3 Web-Server without hardware add-on components

Is it possible to have a Web-Server without hardware add-on components? Yes it is. To define the precise function of add-on components it is necessary to examine the structure of a Web-Server more closely:

The job of a Web-Server is to prepare and administer files (HTML pages, image files, Java Applets, etc.) and despatch those files on request. Whereas file administration takes a simple form, this is not the case for file despatch – since all communication must take place with standard browsers via the TCP/IP protocol. However, TCP/IP places heavy demands on memory resources and CPU power.

SBC Web-Server functions on the Saia PCD® Classic series, are splitted: The file administration part is located in the Saia PCD®'s CPU and the part treating the request from TCP/IP communications is installed on the PC that is used to call control and monitor-ing functions. It's then possible to communicate between PCs and Saia PCD®s via a simple, efficient protocol that uses significant less resources than TCP/IP. Reduced to the essential transfer function, the protocol also guarantees deterministic behaviour and does not impact the real-time behaviour of the Saia PCD®.

Standard browsers, like Microsoft Internet Explorer or Netscape Communicator, request files from Web-Servers under the TCP/IP protocol. To enable communication with the Web-Server in the Saia PCD®, the Web-Connect software must be installed on that PC, which works as the communication part of the Web-Server. Web-Connect receives requests from the browser in the TCP/IP protocol and forwards them with an efficient protocol to the Saia PCD®, this by the communication configured, it could be serial direct (PCD3.RIO also), modem or also Ethernet-TCP/IP. Saia PCD® answers are converted back into TCP/IP and conveyed to the browser.

Splitting the Web-Server dispenses with the need for additional TCP/IP co-processors or TCP/IP add-on components for the Saia PCD®s. It's then possible to connect and use the Saia PCD® Web-Server by a simple serial line or also by a modem. If necessary, it is also possible to communicate with an Ethernet/TCP-IP card, but it's not a must.

Advantages of the SBC Web-Server:

- Simple, low-cost creation of control and monitoring interfaces with familiar Microsofttools, such as Word, Excel, PowerPoint, FrontPage, etc.
- No Run time license necessary. In fact no license are necessary to operate or connect to the Web-Server.
- Operation of the control and monitoring interface with standard browsers, like Microsoft Internet Explorer or Netscape Communicator.
  Browsers have developped a standard user interface (with the familiar "Forward", "Back", "Home" and "Refresh" buttons) so that untrained staff can immediately operate a control and monitoring interface via a browser.
  Navigation by simply clicking on "Left" is intuitive and presents no problems.
- No expensive infrastructure required on Saia PCD® or PC, such as Ethernet networks, Internet providers, Interface modules, etc.
- With an integrated modem, ideal for Tele servicing of machines and processes over great distances.
- With the 4 built-in variable pages, the status of each process data point can be watched and, if necessary modified.
- The access to the control and monitoring interfaces is protected with 4, individual definable passwords.

1.4 HTML pages integrated in the Firmware

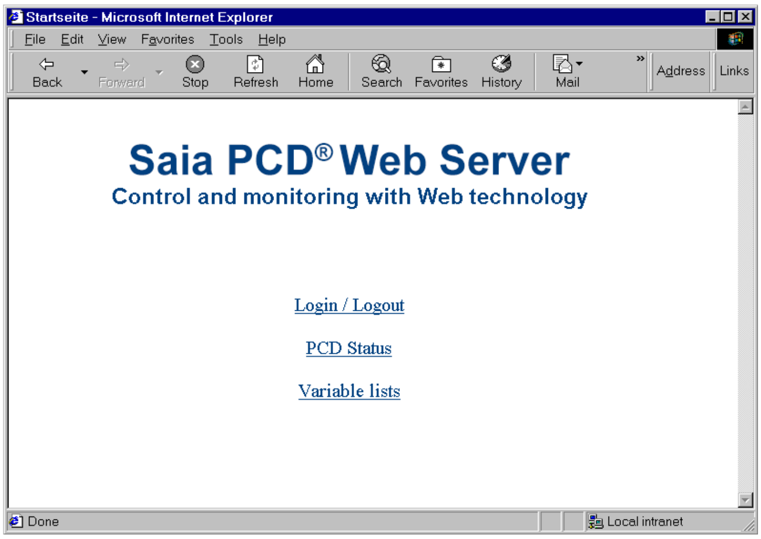
The Web-Server comes from the factory with 4 stored HTML pages that can be displayed with any standard browser. To acces these pages no special actions are neccessary. The 4 pages stored in FW are as follows:

start.htm	Standard start page: If no page is referenced on the URL when connecting to a station, the start.htm page is always the first file to be sent to the Web-Browser. A page with this name always have to be included in the web project. Starting from this page, links are then made to other HTML pages.
pwdform.htm	Standard password entry page: The password can be entered on this page. The user's own pwdform.htm page can be defined.
status.htm	Page where you can see the type of Saia PCD®, its actual status and the operating system version (firmware).
varlist.htm	Standard variable list: Under this menu item 4 variable pages are stored that can be edited as desired by the user. All Saia PCD® data points can be accessed from these pages (if permitted by password level). The user's own varlist.htm page can be defined.

Rules governing the 4 HTML pages already contained in the Web-Server from factory:

- If HTML pages are stored in the user program and they have the same name as the ones mentioned (start.htm, pwdform.htm, etc…), when the Web-Browser invokes these pages it will be loaded with the ones stored in the user program.
- If the HTML pages listed above are invoked in the Web-Browser but do not exist in the user program, the default standard pages will be loaded into the Web-Browser.

Page: **start.htm**


Startseite - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Address Links

Saia PCD® Web Server

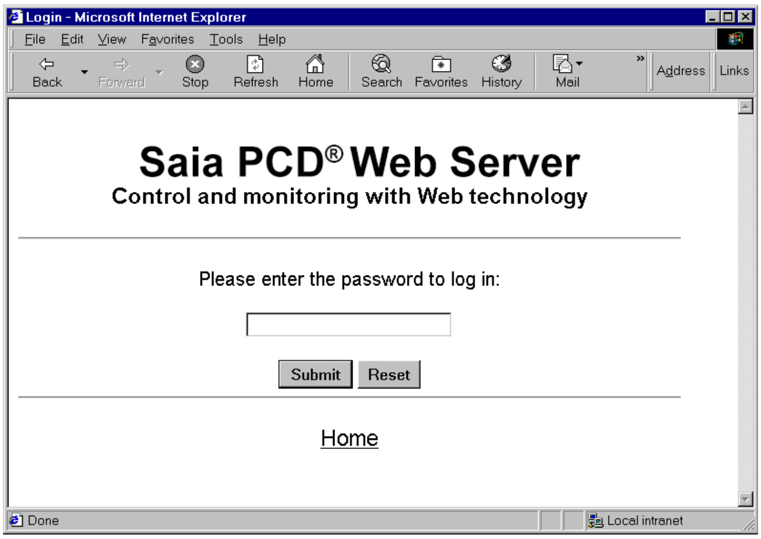
Control and monitoring with Web technology

Login / Logout

PCD Status

Variable lists

Page: **pwdform.htm**


Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Address Links

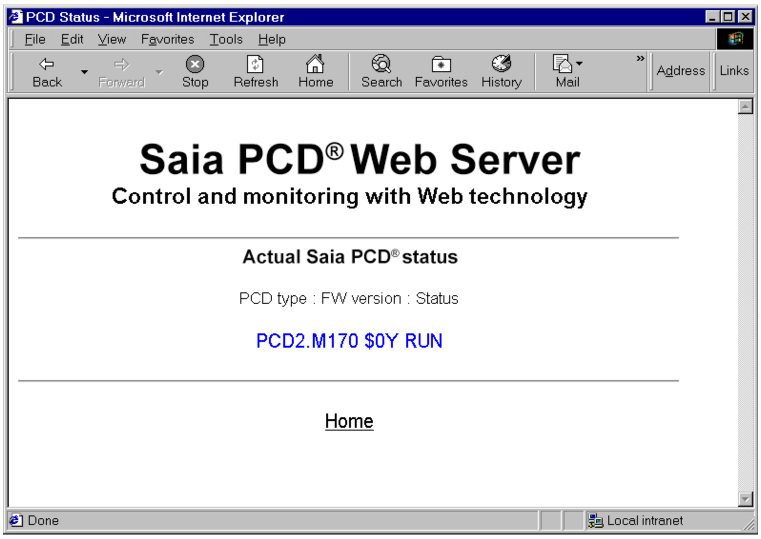
Saia PCD® Web Server

Control and monitoring with Web technology

Please enter the password to log in:

Submit Reset

Home


PCD Status - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Address Links
Done
Local intranet

Saia PCD® Web Server

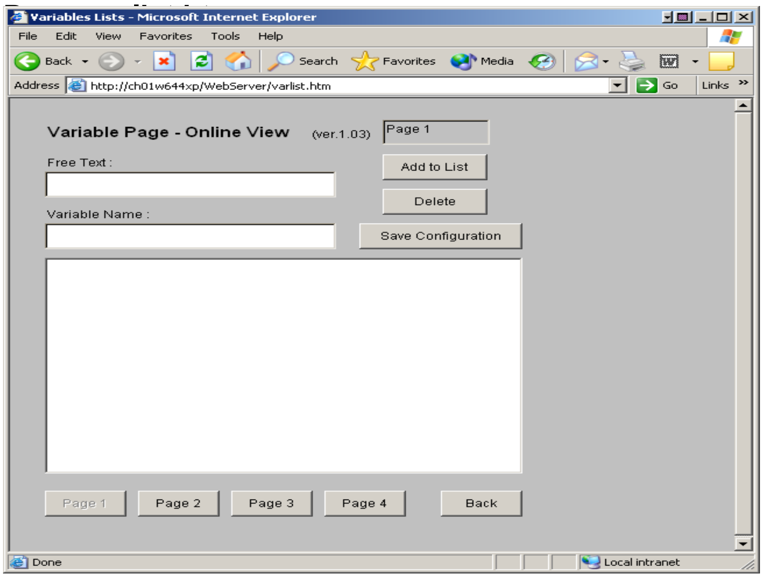
Control and monitoring with Web technology

Actual Saia PCD® status

PCD type : FW version : Status

PCD2.M170 $0Y RUN

Home


Variables Lists - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://ch01w644xp/WebServer/varlist.htm
Go Links
Page 1
Add to List
Delete
Save Configuration
Page 1 Page 2 Page 3 Page 4 Back
Done
Local intranet

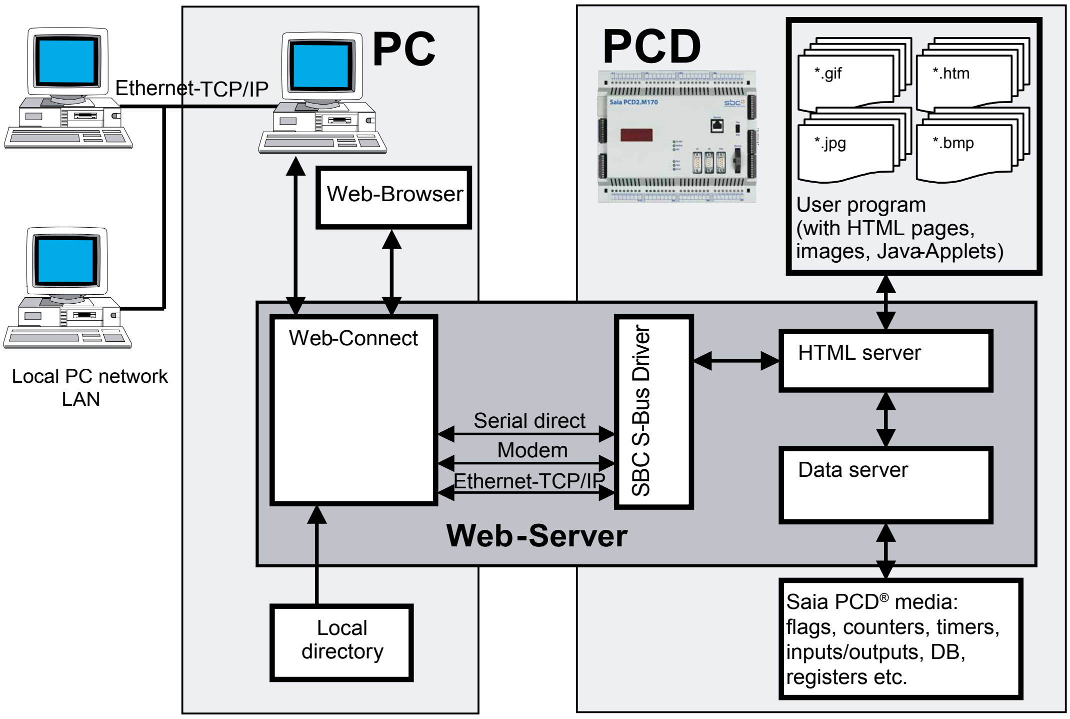
Variable Page - Online View (ver.1.03)

Free Text :

Variable Name :

2 Structure and function

The Saia PCD®'s integral Web-Server comprises the following individual elements:



2.1 HTML-Server

The HTML-Server is the heart of the entire Web-Server. It sends HTML pages requested by the Web-Browser (and any necessary images/files) across the S-Bus Driver to the PC.

HTML pages, images or files are stored in the user program memory of the Saia PCD®, it belongs to the user program and so it's downloaded from the Saia PG5®.

In the HTML server, HTML pages are also checked for their process data point identification key (PDP key). As soon as the HTML-Server finds a PDP key in a HTML page, it sends that PDP key to the Data-Server.

The HTML-Server checks only files with suffix *.HTML or *.HTM for a possible PDP key to be transferred to the data server.
This means that it is compulsory for all files containing a PDP key to have the suffix *.HTML or *.HTM.

2.2 Data-Server

The Data-Server processes PDP keys received from the HTML-Server and transfers them in the requested Saia PCD® data directly from the Saia PCD®'s memory to the HTML-Server.
The Data-Server can access DB, flags, register, inputs, outputs, timers, counter, texts and Saia PCD® status.

2.3 HTML pages, images, graphics etc

One of the Web-Server's essential advantages is the creation of a control and monitoring interface using Microsoft standard tools. Any software tool that generates

HTML can be used to create HTML pages. For example, individual pages of a control and monitoring project can conveniently be edited as a text file using Word.

Any tables, graphics, diagrams, photographs, etc. that can be displayed by the Web-Browser may be inserted. The full functionality of the relevant Microsoft application can be used for this. No prior knowledge is required to link any number of pages into complex menu structures.

To integrate current Saia PCD® data into the control and monitoring function, simple text commands in the form of a PDP key are inserted in the text file. In this way Saia PCD® media can be displayed during run-time in various formats.

All HTML pages, images or files belonging to a control and monitoring application are stored in the Saia PCD® as data blocks (DBs) of the user program. The user is able to store any file in any preferred format and with whatever suffix (*.htm, *.html, *.bmp, *.gif, *.txt, *.zip or any other) in the Saia PCD®, as long as the file does not exceed **100 kByte** in size.
The HTML-Server only checks files with suffix *.HTML or *.HTM for the PDP key.

If the control and monitoring application needs to display large HTML pages, images or files that doesn't fit in the Saia PCD® memory then there is the possibility to store this HTML pages, images or files on the hard disk of the PC, where the Web-Connect software is running. The HTML pages, images or files have to be stored in the Web-Connect directory.

2.4 Local directory

The local directory, is a directory on the PC, where the Web-Connect software is running. By default its name and location is "C:\WebPages", this location and name can be changed in the Web-Connect option and during the Software installation.

This directory can contain files, which from now on will be called "the local files".

It is possible to save files in this directory which we don't want to save in the Saia PCD®. This option is very useful when the files we want to save are quite huge. If we would save these files in the Saia PCD® it could take a long time to load them, not so if they are saved in the local directory. The mechanism will be the following, the file is requested to the Saia PCD® and if the file doesn't exist there, the Web-Connect software will check then the local directory to find and return it. This operation is totally transparent for the user.

Here you also save the local web-site, like the default.html page and all files going with it, this will be treated in the chapter " Web-Connect".

2.5 Web-Builder

Saia PCD® classic controllers are programmed with Saia PG5® programming software. Control and monitoring files are stored in the Saia PCD®'s user program memory.

All files belonging to the control and monitoring function (HTML pages, images, etc.), are stored in the user program of Saia PG5®. To do so, the web files must be converted into source files to be integrated in the Saia PG5® project. Conversion takes place with the easy-to-use add-on software tool "Web-Builder". This add-on tool is called automatically, when a file of the type Web-Server Project (.wsp) is clicked in the Saia PG5® project.

This tool is used to select HTML pages and images which have to be integrated in the

Web-Server, and also to define some Web-Server settings. Then it is only necessary to generate a source file.

Back in the Saia PG5® project, the Web-Server content and the settings will be compiled in the same time as the Saia PG5® software and also downloaded in the same time into the Saia PCD®.

The Web-Builder can run in two modes:

Demo mode:

When no license has been found in the Saia PG5® for this tools.

- Only 2 HTML pages can be placed into the project (Web-Server)
- It is not possible to select the option VarList Sites.

Full version:

- The software needs to be registered in Saia PG5® Licenses system
- No limitation of the number of HTML pages, images, etc., that can be placed into the project.

2.6 Web-Connect

To display in the Web-Browser the control and monitoring functions stored in the controller, it is necessary to start up the Web-Connect, PC software.

This software receives the request coming from the clients (Browser) and forwards them to the right Saia PCD® station using the adequate way and vice-versa.

This software is doing the link between the PC-Browser and the communication to the Saia PCD®s.

The Web-Connect program enables the user to :

- Configure the station that will be accessed from this PC.
- Configure the channels that will be used to access the Station.
- Forward request from the PC to the Saia PCD®s
- Forward request from other PC on the same LAN to the Saia PCD®s.

As you can see there are few different ways to access to a Saia PCD®, we can make two categories of connection from that, direct access to the Saia PCD® and indirect access to the Saia PCD®.

2.6.1 Connection possibilities

Direct mode:
The Web-Browser which is making the request and the Web-Connect software are running both on the same PC. But then depending which is the Saia PCD® concerned by the request, it can take different connection types, here they are:

① Direct connection to the serial port from the PC.
The Saia PCD® is directly connected to the PC with a serial line or by a gateway.

② Direct world-wide connection via modem
The Saia PCD® is equipped with a modem and communicates with the PC via that modem.

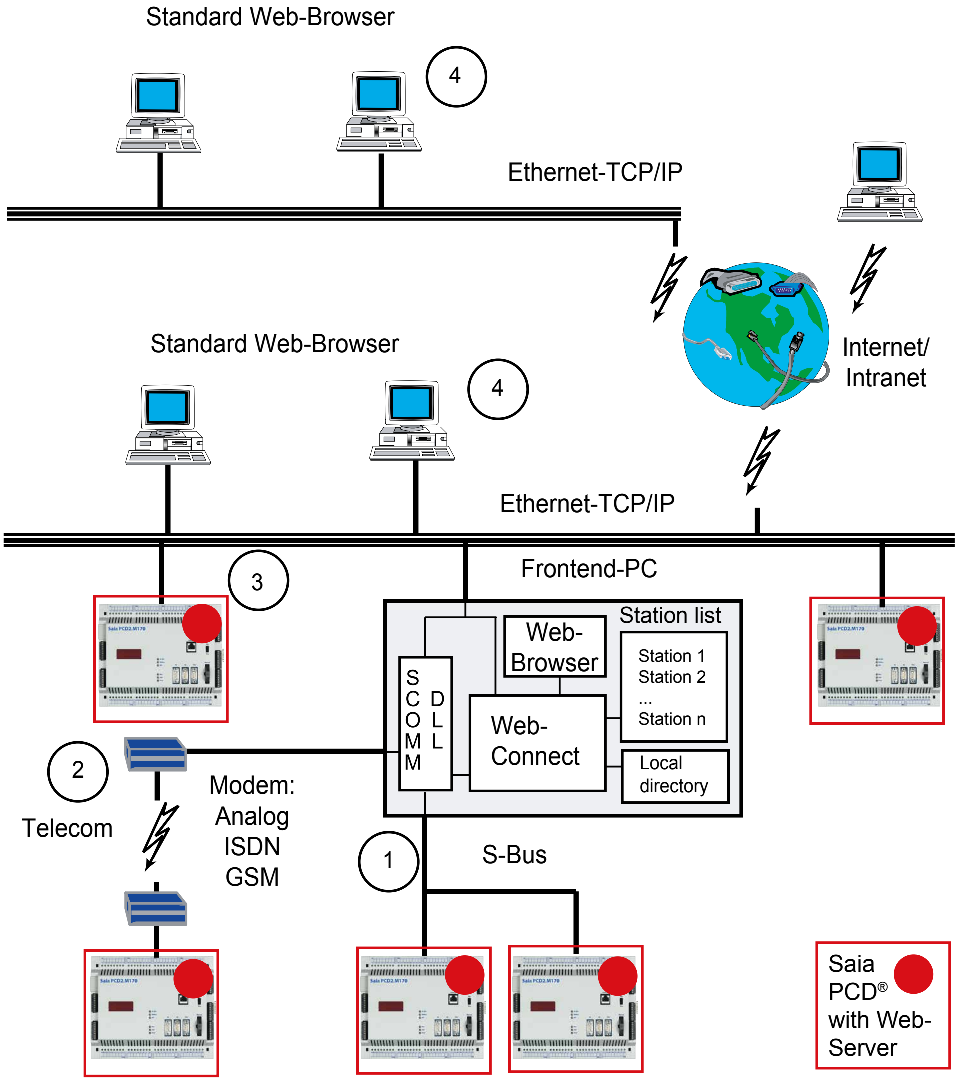
③ Direct by an Ethernet-TCP/IP module.
The Saia PCD® is equipped with a module Ethernet-TCP/IP and the PC is on a LAN.

Indirect mode:

④ Indirect, via LAN network
Other PCs, where no Web-Connect software is running, can also make re-

quests to the Saia PCD® with the Web-Server. To do so they have to do an indirect access. In fact the request will go first to the PC where Web-Connect software is running and then, this PC will send the request to the Saia PCD®. The answer will be sent back to the request originator on the same way.

This both ways of working are possible, because the Web-Connect software, doesn't make a difference if the request is coming from the local Web-Browser or if it is coming from a Web-Browser running on an other PC of the LAN.



2.7 Web-Browser

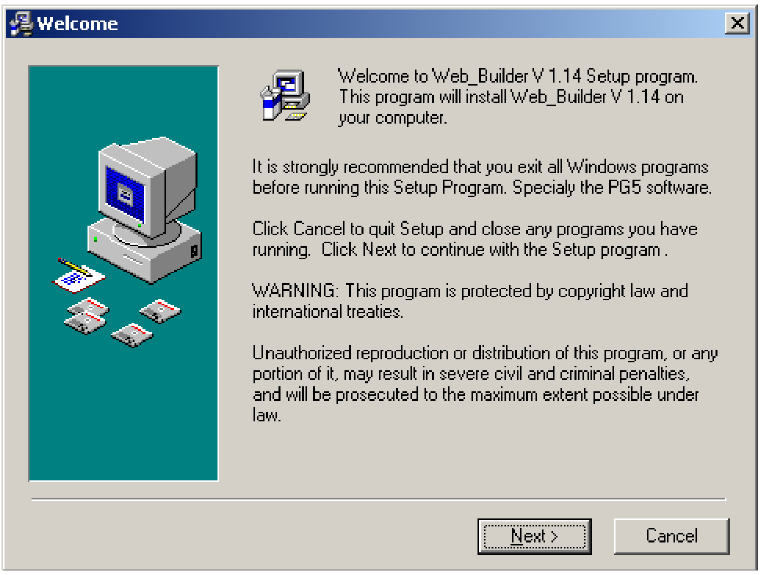
We recommend to use Netscape Navigator or Microsoft Explorer as Web-Browser to display HTML pages, images or files stored in the Saia PCD®, which belong to the control and monitoring application.

3 Installation and Settings

Prior to install the Web-Builder, make sure that a Saia PG5® version 1.2 or higher is installed on your PC. If not, install the Saia PG5® first.

3.1 Web-Builder

Web-Builder is installed with a setup program:
Make sure, that you have a license for the Web-Builder. In Windows, select the "Start", "Execute" menu and select the path used to save the Web-Builder. The "Setup.exe" file must be selected and started up in this path. Web-Builder will then be installed on the PC.



Welcome to Web_Builder V 1.14 Setup program. This program will install Web_Builder V 1.14 on your computer.

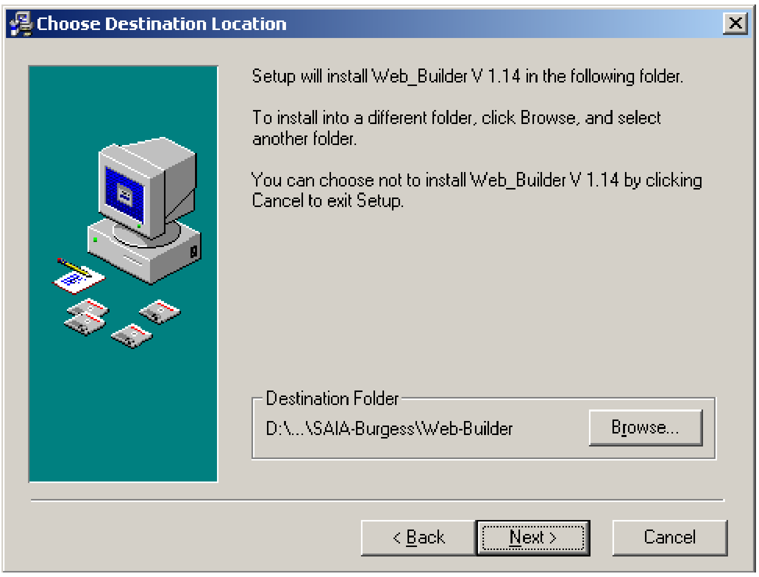
It is strongly recommended that you exit all Windows programs before running this Setup Program. Specialy the PG5 software.

Click Cancel to quit Setup and close any programs you have running. Click Next to continue with the Setup program .

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Web-Builder is installed by default in path: C:\Programme\SBC\Classic\Web-Builder.



Setup will install Web_Builder V 1.14 in the following folder.

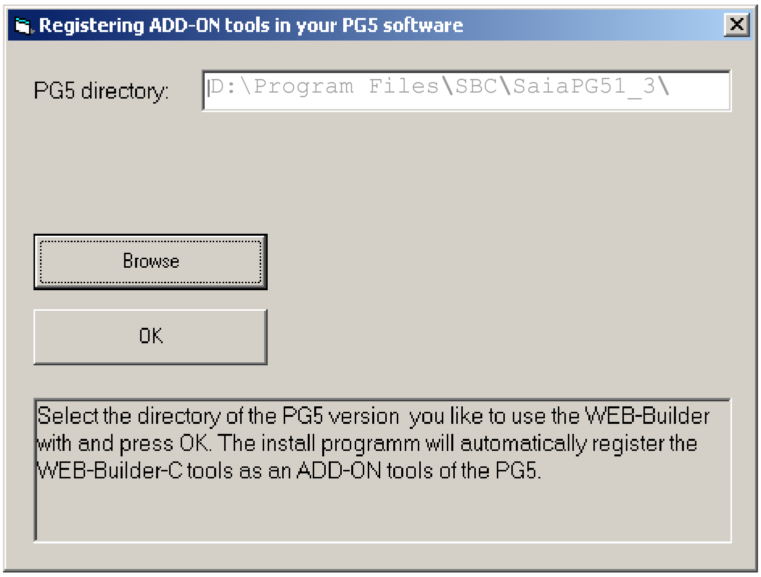
To install into a different folder, click Browse, and select another folder.

You can choose not to install Web_Builder V 1.14 by clicking Cancel to exit Setup.

Destination Folder
D:\...\SAIA-Burgess\Web-Builder

At the end of the installation, the software will seek on your computer the Saia PG5®, the result will be dispalyed in a window like the one shown below. If the path found for Saia PG5® is not correct, you can change it.

If you agree with the path, confirm with “OK”. This will generate a file and place it into the Saia PG5® directory, next time you start the Saia PG5® software, the add-on tools “Web-Builder” will auto-register.

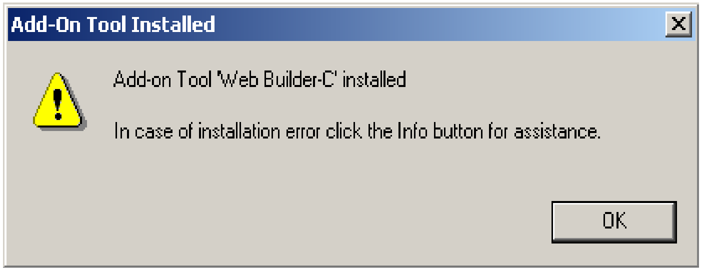


Registering ADD-ON tools in your PG5 software

PG5 directory: D:\Program Files\SBC\SaiaPG51_3\

Select the directory of the PG5 version you like to use the WEB-Builder with and press OK. The install programm will automatically register the WEB-Builder-C tools as an ADD-ON tools of the PG5.

Start the Saia PG5® software, a window like this one will appear and confirm that the add-on is installed.



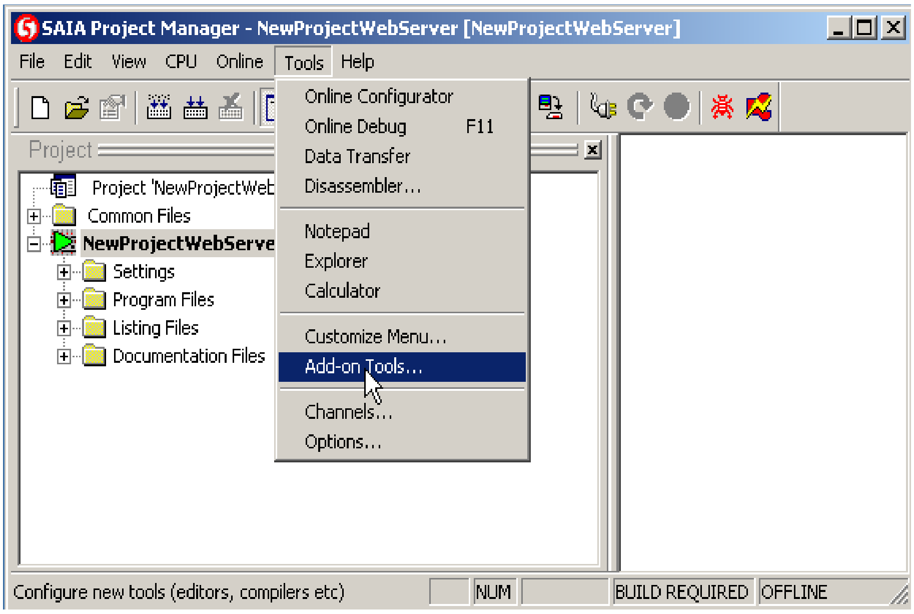
Add-On Tool Installed

Add-on Tool 'Web Builder-C' installed

In case of installation error click the Info button for assistance.

If a problem occured during registering the add-on tool, you can now register it manually:

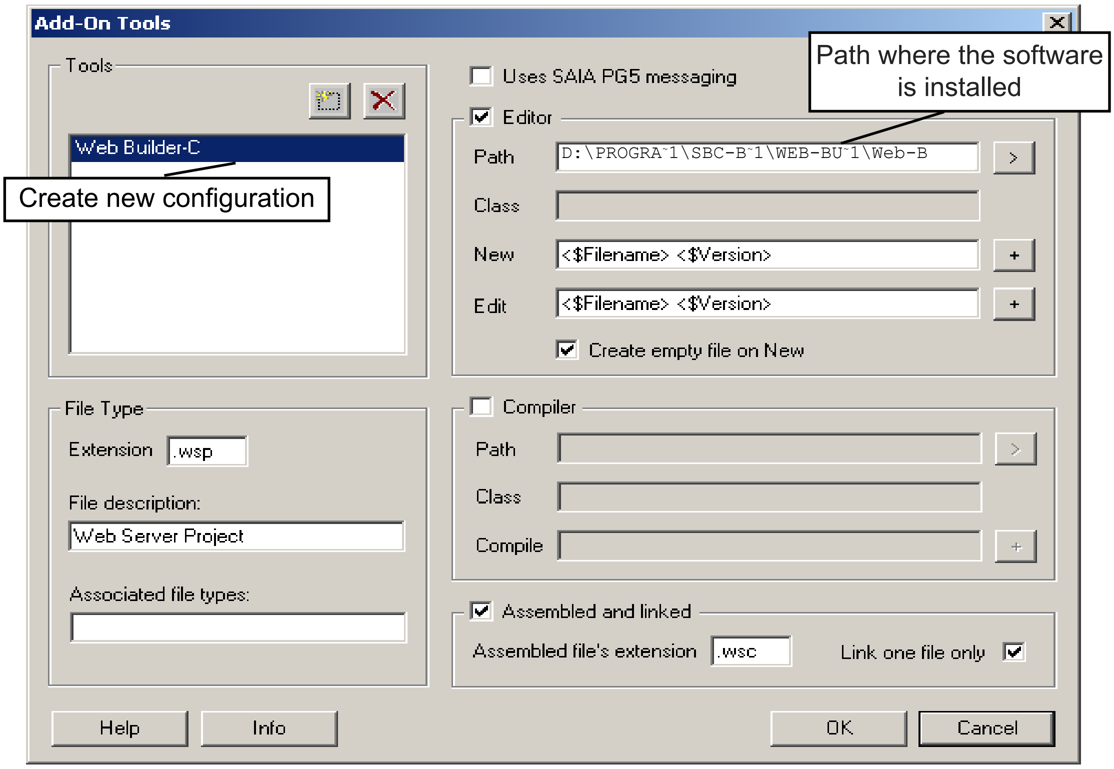
1) Go into the menu Tools of the Saia PG5® and select the “Add-on tools” command.



SAIA Project Manager - NewProjectWebServer [NewProjectWebServer]

Configure new tools (editors, compilers etc)

2) Create a new configuraton for an add-on tool.
3) In the Editor path, select the path where you installed the Web-Builder software.



Add-On Tools

Tools: Web Builder-C

Create new configuration

Path where the software is installed

☐ Uses SAIA PG5 messaging

☑ Editor

Path: D:\PROGRA~1\SBC-B~1\WEB-BU~1\Web-B

Class:

New: <$Filename> <$Version>

Edit: <$Filename> <$Version>

☑ Create empty file on New

File Type

Extension: .wsp

File description: Web Server Project

Associated file types:

☐ Compiler

Path:

Class:

Compile:

☑ Assembled and linked

Assembled file's extension: .wsc

Link one file only ☑

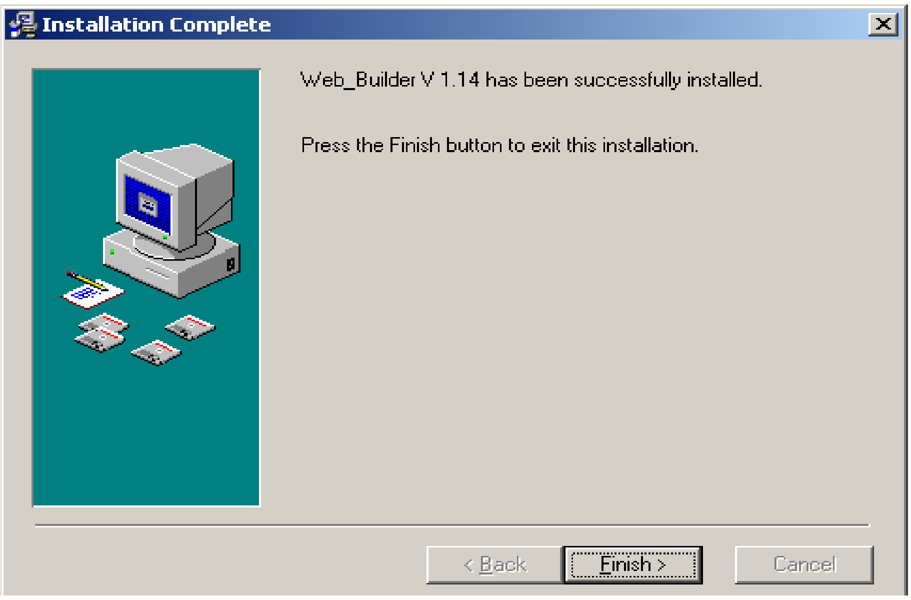
Then fill up all the other fields as shown on this picture.

<u>File Type:</u>
File type extension : **.wsp**
File description: **Web-Server Project**

<u>Editor:</u> ☑
Path: where the Web-Builder is installed
New: select **<$Filename><$Version>**
Edit: select **<$Filename> <$Version>**
Create empty file on New: ☑.

<u>Assembled and linked:</u> ☑
Assembled files extension: **.wsc**
Link one file only: ☑.

After a successful installation the following screen appears:



Installation Complete

Web_Builder V 1.14 has been successfully installed.

Press the Finish button to exit this installation.

3.2 Web-Connect

For installation and settings please refer to the manual 26-800 Web-Connect. This document can be found on SBC support site www.sbc-support.com under Software.

3.3 Web-Browser and Network settings

If no Web-Browser has yet been installed on the PC, this must be done in accordance with the installation information for each individual Web-Browser.
Microsoft Virtual Machine must be installed.
Installation of Microsoft Virtual Machine takes place during the installation of the Web-Browser.

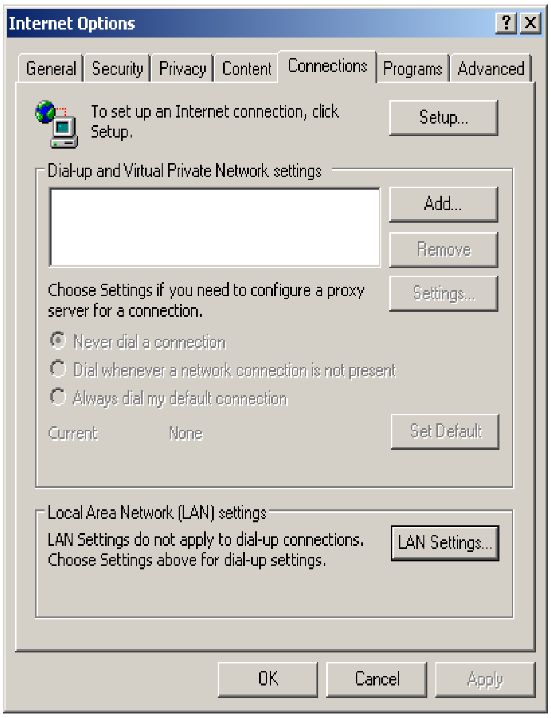
MS Internet Explorer or Netscape Navigator must be installed (version see **1.2 Minimum requirements**).

Web pages from which the Web-Browsers can be downloaded.

MS Internet Explorer in German:	http://www.eu.microsoft.com/germany/
MS Internet Explorer in English:	http://www.microsoft.com/
Netscape Navigator in German:	http://home.de.netscape.com/de/
Netscape Navigator in English:	http://home.netscape.com/

3.3.1 The PC's proxy server

If the PC is connected to Internet via a proxy server, the proxy server must be bypassed for local addresses. Taking the Internet Explorer as an example, this can be done with the following option:



Internet Options

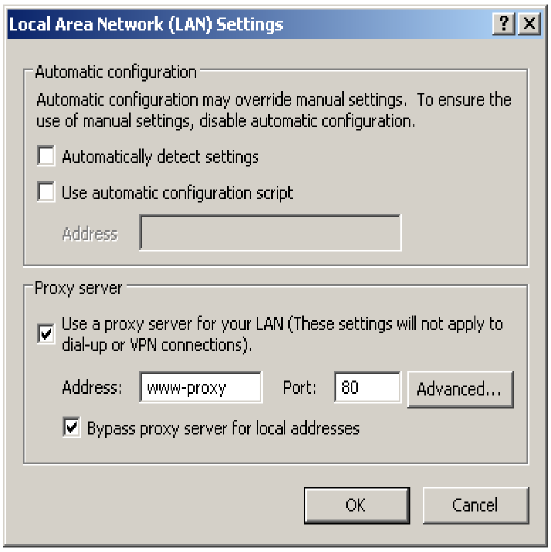
General | Security | Privacy | Content | Connections | Programs | Advanced

To set up an Internet connection, click Setup. [Setup...]

Dial-up and Virtual Private Network settings
[Add...] [Remove] [Settings...]
Choose Settings if you need to configure a proxy server for a connection.
● Never dial a connection
○ Dial whenever a network connection is not present
○ Always dial my default connection
Current None [Set Default]

Local Area Network (LAN) settings
LAN Settings do not apply to dial-up connections. Choose Settings above for dial-up settings. [LAN Settings...]

[OK] [Cancel] [Apply]



Local Area Network (LAN) Settings

Automatic configuration
Automatic configuration may override manual settings. To ensure the use of manual settings, disable automatic configuration.
□ Automatically detect settings
□ Use automatic configuration script
Address

Proxy server
☑ Use a proxy server for your LAN (These settings will not apply to dial-up or VPN connections).
Address: www-proxy Port: 80 [Advanced...]
☑ Bypass proxy server for local addresses

[OK] [Cancel]

Tools→Internet Options..., select option **Connections** and select button **LAN Settings** by selecting option **Bypass proxy server for local addresses**.

3.3.2 Modem connection

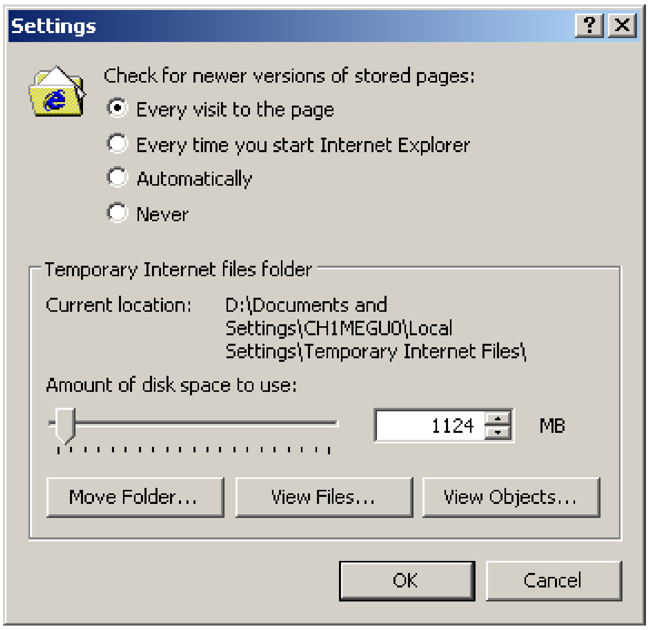
If the PC communicates with Internet via a modem connection, the **Never dial a connection** option must be selected. This option appears in Internet Explorer under **Tools→Internet Options**..., tab **Connections**.

It is necessary to select this mode, to avoid that every time the PC tries to connect to internet just to get access to the local software Web-Connect.

3.3.3 The PC’s cache memory

In order to refresh HTML pages periodically, the Web-Browser’s cache memory must be switched off.

This is possible with Internet Explorer under **Tools→Internet Options**..., **General**, **Settings...**, selecting the option “Check for newer versions of stored pages“ **Every visit to the page**.



Check for newer versions of stored pages:

● Every visit to the page

○ Every time you start Internet Explorer

○ Automatically

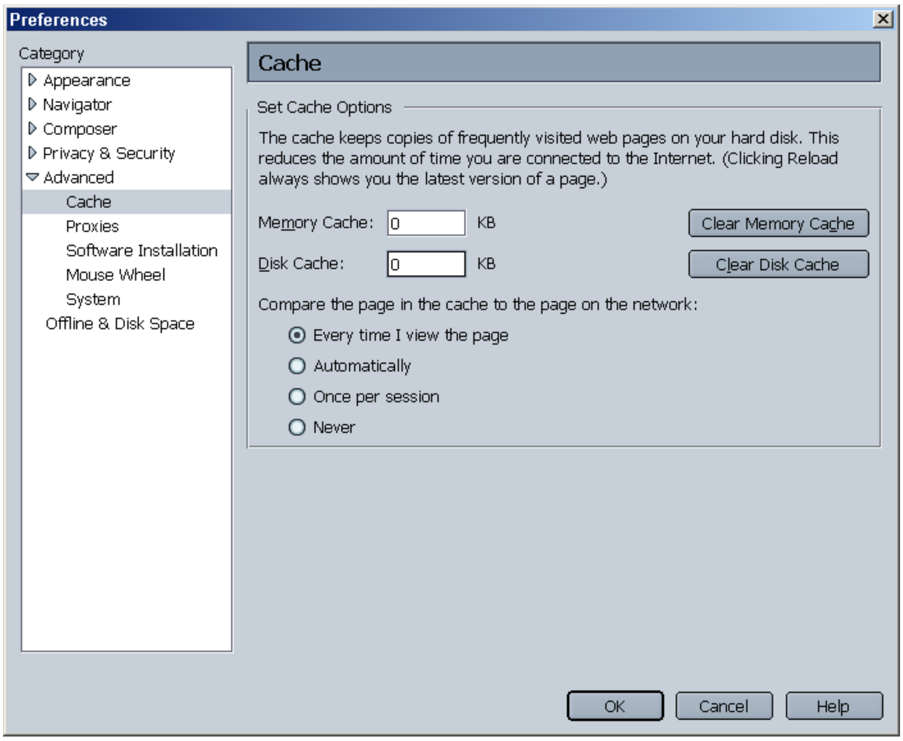
○ Never

Temporary Internet files folder

Current location: D:\Documents and Settings\CH1MEGU0\Local Settings\Temporary Internet Files\

Amount of disk space to use: 1124 MB

With Netscape Communicator – menu: **Edit -> Preference**:



Set Cache Options

The cache keeps copies of frequently visited web pages on your hard disk. This reduces the amount of time you are connected to the Internet. (Clicking Reload always shows you the latest version of a page.)

Memory Cache: 0 KB

Disk Cache: 0 KB

Compare the page in the cache to the page on the network:

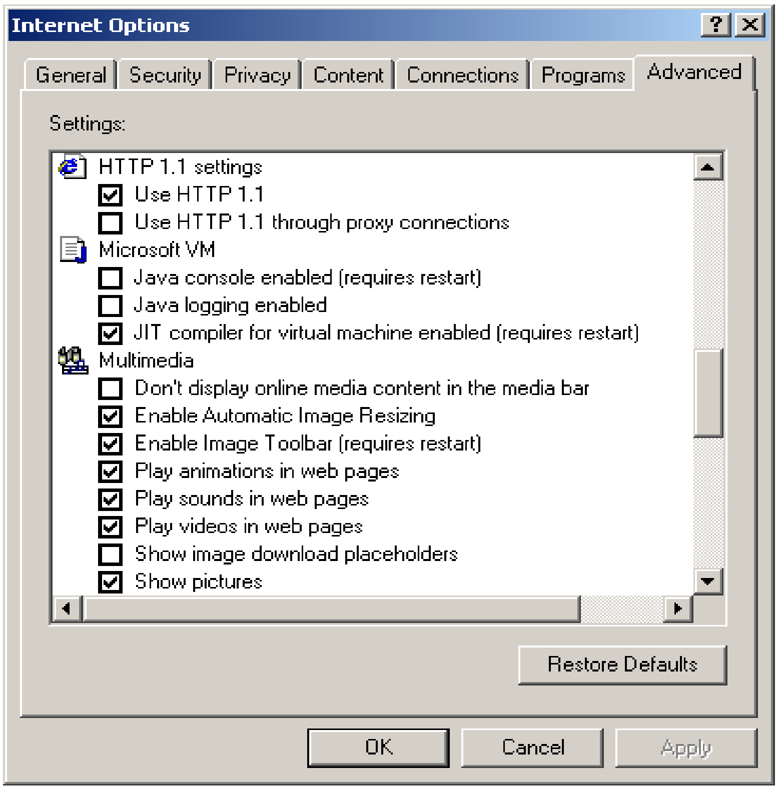
● Every time I view the page

○ Automatically

○ Once per session

○ Never

To optimise the execution of Java Applets under Internet Explorer, the Java Console option must be disabled.



Settings:

HTTP 1.1 settings
- ☑ Use HTTP 1.1
- □ Use HTTP 1.1 through proxy connections

Microsoft VM
- □ Java console enabled (requires restart)
- □ Java logging enabled
- ☑ JIT compiler for virtual machine enabled (requires restart)

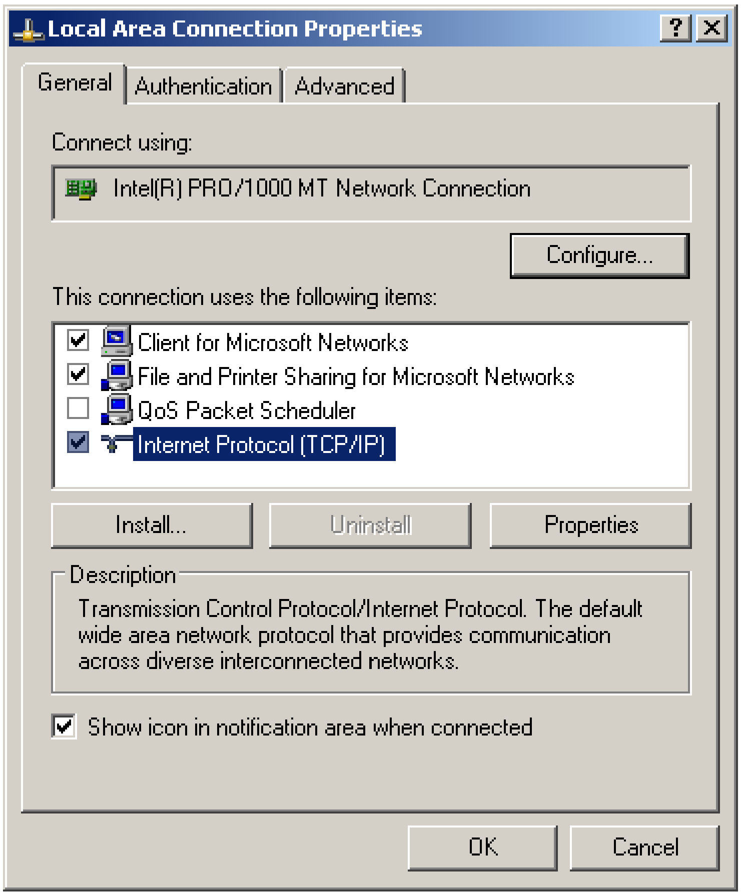
Multimedia
- □ Don't display online media content in the media bar
- ☑ Enable Automatic Image Resizing
- ☑ Enable Image Toolbar (requires restart)
- ☑ Play animations in web pages
- ☑ Play sounds in web pages
- ☑ Play videos in web pages
- □ Show image download placeholders
- ☑ Show pictures

To check Java Console settings, go into Internet Explorer and select **Tools, Internet Options, Advanced, Microsoft VM** (Virtual Machine).
This involves ensuring that the Java Console Enabled option is **not** selected.

3.4 Settings and Test of TCP/IP

3.4.1 TCP/IP

The PC, on which the Web-Browser is supposed to display the Saia PCD®'s control and monitoring functions must have TCP/IP.
It is possible to test if the TCP/IP protocol is installed on the PC by checking the 'Protocol' option under **Start, Settings, Control Panel, Network Connections,** e.g. **Local Area Connection**.
The TCP/IP protocol is installed on the PC if a TCP/IP protocol is defined under the tab **General, Properties**.



Local Area Connection Properties

Connect using:

Intel(R) PRO/1000 MT Network Connection

This connection uses the following items:

☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☐ QoS Packet Scheduler
☑ Internet Protocol (TCP/IP)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

☑ Show icon in notification area when connected

3.4.2 IP address for PC

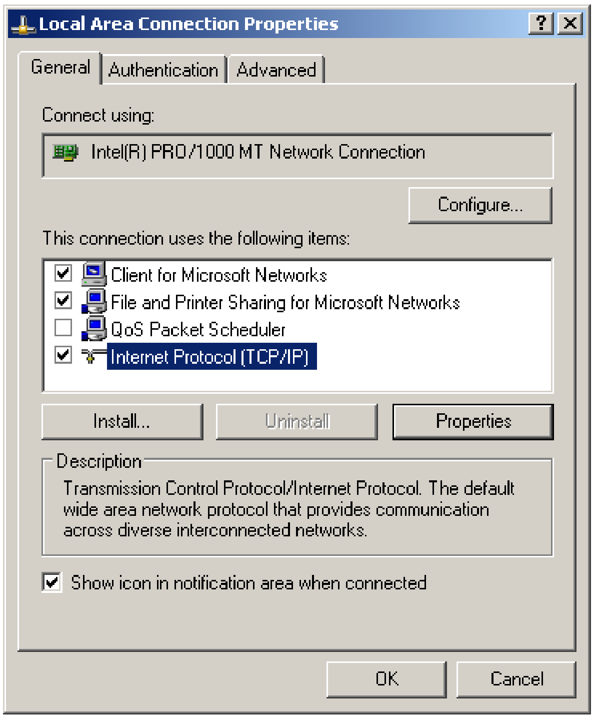
You will see later, that access to the Saia PCD® Web-Server, is made with a Web-Browser. In this Web-Browser a valid assignment to the PC, where the Web-Connect SW is running has to be defined. This assignment is done through the URL which has to include the IP address or the PC name.

Now, if you don't want to use the PC name and prefer to use the IP address, this has to be a fixed one.You first need to have, at your disposition, a unique and valid IP address.

Having its own, unique address means the option **Obtain an IP address from a DHCP server** under the PC's Network TCP/IP settings must **not** be selected. This option appears in WIN NT under **Start, Settings, Control Panel, Network:** button **Protocols**. The IP address of the PC can also be defined under the same menu item.

Every PC has the same default IP address as a factory setting.
This is: 127.0.0.1
Therefore, if no IP address has been specified, this address can be used in the Web-Browser to access to the PLC's HTML pages.

For WIN 2000



Local Area Connection Properties

General | Authentication | Advanced

Connect using:

Intel(R) PRO/1000 MT Network Connection

Configure...

This connection uses the following items:

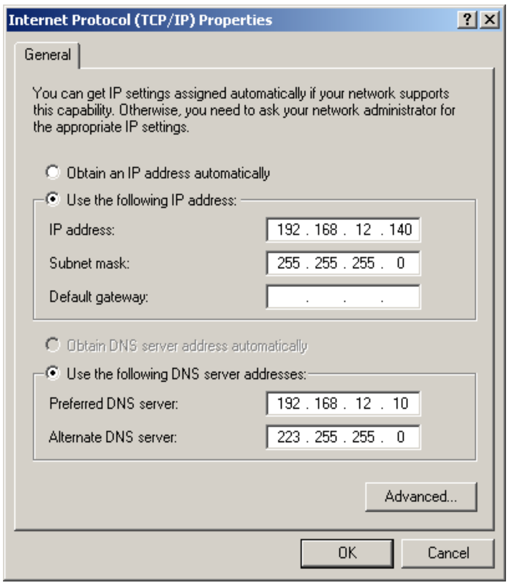
☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☐ QoS Packet Scheduler
☑ Internet Protocol (TCP/IP)

Install... | Uninstall | Properties

Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

☑ Show icon in notification area when connected

OK | Cancel



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address: 192 . 168 . 12 . 140
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: 192 . 168 . 12 . 10
Alternate DNS server: 223 . 255 . 255 . 0

Advanced...

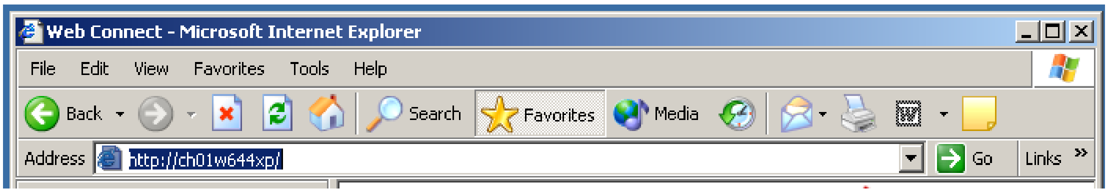
OK | Cancel

An IP address can be specified in the above fields.

3.4.3 PC name

As said before, it is also possible to work with the PC name instead of the IP address. So in the URL you can enter **localhost** and you will find your PC name.

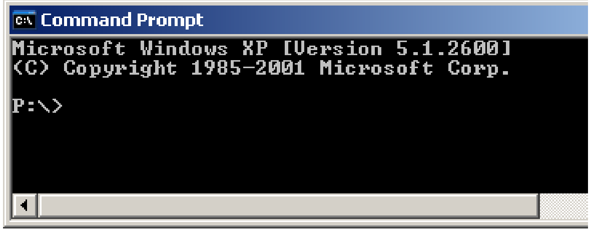
In our case the name is ch01w644xp.



Address: http://ch01w644xp/

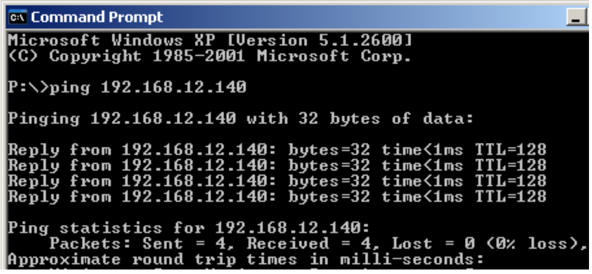
3.4.4 Test of the IP address and the TCP/IP protocol

To check if a TCP/IP protocol has been installed on the PC and if the IP address is correct, the following test can be carried out:



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

P:\>
```

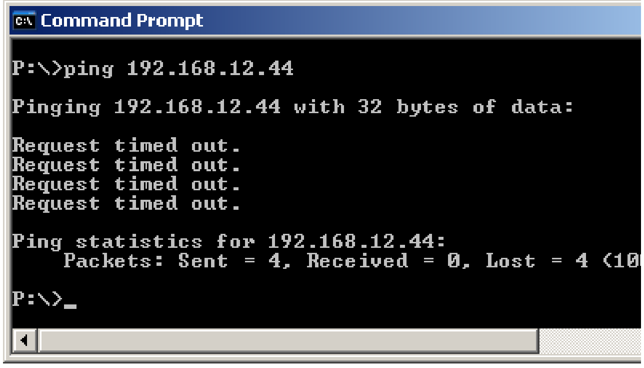
Start the DOS console with **Start, Programs, Accessories, Command Prompt.**



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

P:\>ping 192.168.12.140

Pinging 192.168.12.140 with 32 bytes of data:

Reply from 192.168.12.140: bytes=32 time<1ms TTL=128
Reply from 192.168.12.140: bytes=32 time<1ms TTL=128
Reply from 192.168.12.140: bytes=32 time<1ms TTL=128
Reply from 192.168.12.140: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.12.140:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

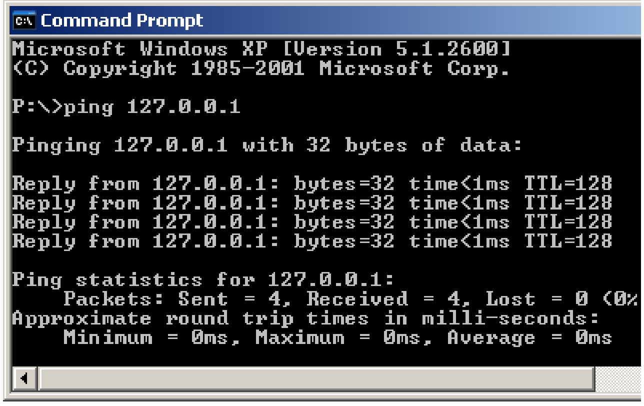
Enter the command ping “IP Number of PC“. e.g. ping 192.168.12.140

If the IP address is correct and TCP/IP is installed, the following reply is shown on the screen.



```
P:\>ping 192.168.12.44

Pinging 192.168.12.44 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.12.44:
    Packets: Sent = 4, Received = 0, Lost = 4 (10

P:\>_
```

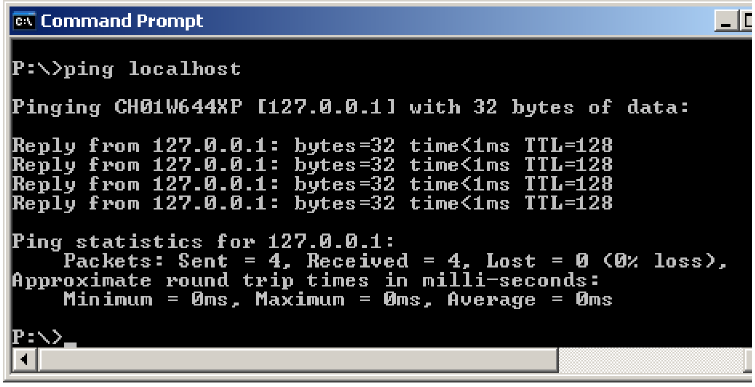
If the IP address doesn’t exist, the following reply is shown on the screen.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

P:\>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0%
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

If the PC’s IP address is not known, an attempt can be made by using the address 127.0.0.1.

It is also possible to check the PC’s name you are using with the ping function, look for **localhost**:



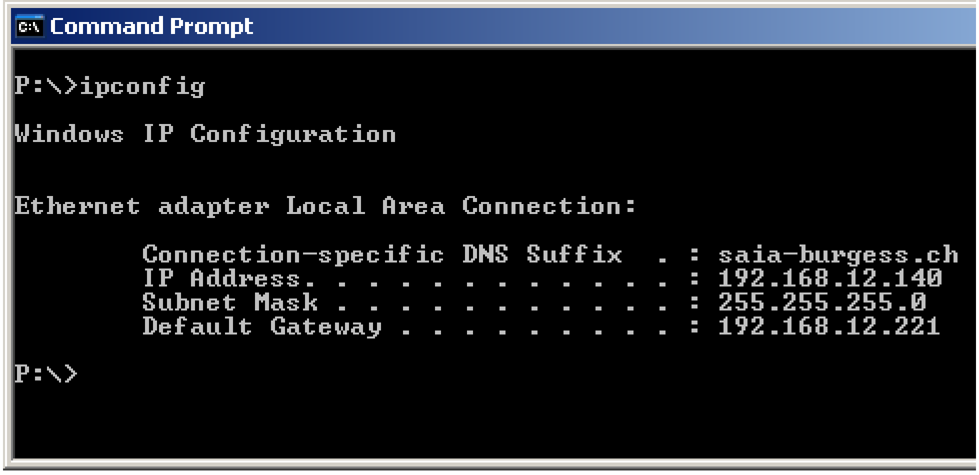
```
P:\>ping localhost

Pinging CH01W644XP [127.0.0.1] with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

P:\>_
```

ping localhost

In this example the PC name is: ch01w644XP.

3.4.5 Finding IP address

The function “**ipconfig**“ is another possibility to find out what is configured on your PC.

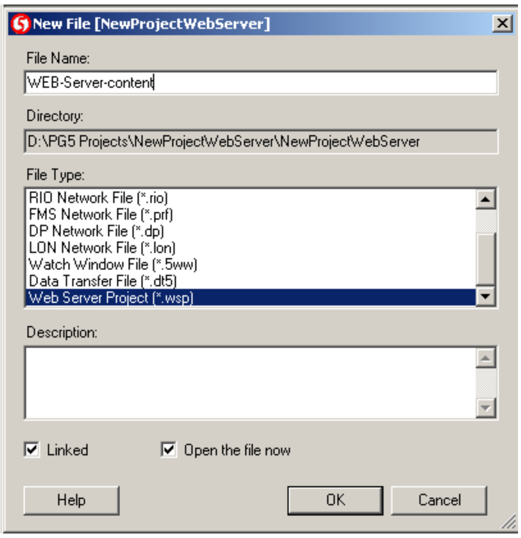


```
P:\>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : saia-burgess.ch
        IP Address. . . . . . . . . . . . : 192.168.12.140
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.12.221

P:\>
```

4 Configuration and Administration

4.1 Web-Builder

It is necessary to use the Web-Builder software to select the content of the Web-Server and also the settings of it. As a Web-Server belongs to a Saia PCD® project, they are also integrated in the Saia PG5® project. So to integrate the Web-Server into your Saia PCD®, you need to add a Web-Server project file into your Saia PG5® project.



New File [NewProjectWebServer]

File Name: WEB-Server-content

Directory: D:\PG5 Projects\NewProjectWebServer\NewProjectWebServer

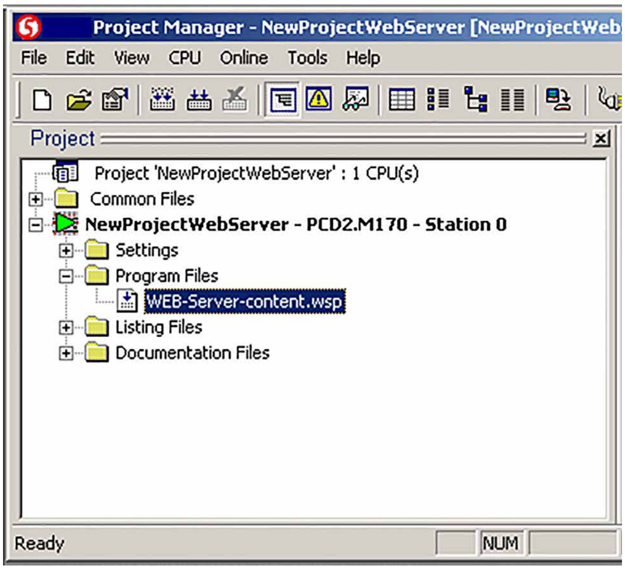
File Type:
- RIO Network File (*.rio)
- FMS Network File (*.prf)
- DP Network File (*.dp)
- LON Network File (*.lon)
- Watch Window File (*.5ww)
- Data Transfer File (*.dt5)
- Web Server Project (*.wsp)

Description:

☑ Linked ☑ Open the file now

Select the Saia PCD® concerned and in the “**File**“ menu select the function “**New…**“, this window will appear.

Then select the file type as “Web Server Project (*.wsp)“ and type the name of the project Web-Server.



Project Manager - NewProjectWebServer [NewProjectWeb

- Project 'NewProjectWebServer' : 1 CPU(s)
- Common Files
- NewProjectWebServer - PCD2.M170 - Station 0
  - Settings
  - Program Files
    - WEB-Server-content.wsp
  - Listing Files
  - Documentation Files

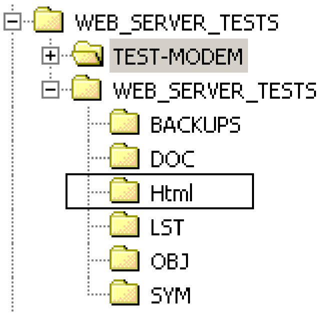
When it’s done, you should have a new file of the type .wsp into your Saia PCD® “Program Files“ subdirectory.

Just double-click on the file to open the Web-Builder software and edit the Server content.

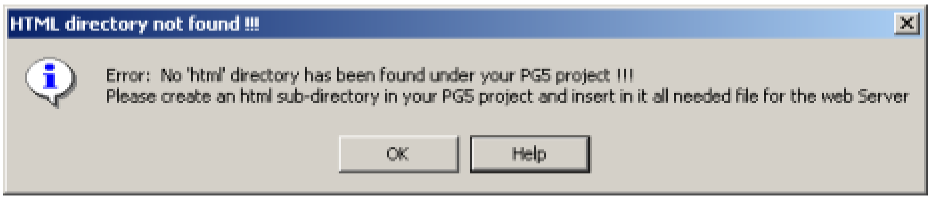
4.1.1 Location of the Web-Server files

The files which are listed in the left window of the Web-Builder are from a specific folder. It is not possible in the Web-Builder to select another directory. In fact the window will display all the files contained in the folder called “Html“, this “Html“ subdirectory has to be created by yourself in the CPU folder of your Saia PG5® project.
In this folder, you need to put all the files you want to use in your Web-Server.
Let see an example :
You have a project called “WEB_SERVER_TEST“, in the project you have two CPUs. “TEST-MODEM“ and “WEB_SERVER_TESTS“.

!
The CPU WEB_SERVER_TEST will run with the Web-Server functionality, so in this CPU you will create the html folder, as shown on the following picture. Then you will copy in it all the html, .jpg, .bmp and .gif files you want to use. This has to be done with a file managing tool like “Win Commander”, “Win Explorer” etc. and **not with the Saia PG5®.**



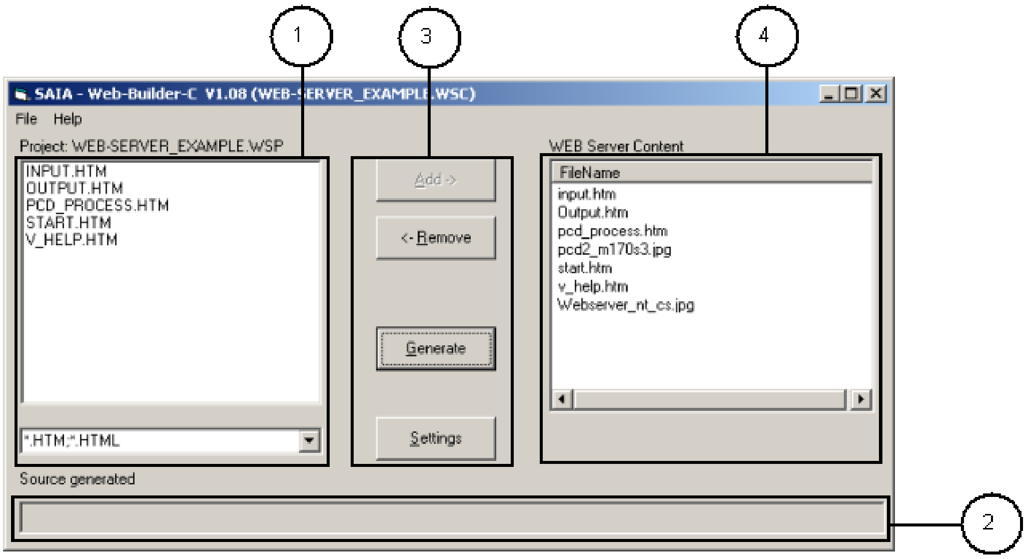
If the folder is not created before you run the Web-Builder software, this one will refuse to run and will show you an error message, before you exit.


HTML directory not found !!!

Error: No 'html' directory has been found under your PG5 project !!!
Please create an html sub-directory in your PG5 project and insert in it all needed file for the web Server

4.1.2 Web-Builder structure

The following screen will be displayed:


SAIA - Web-Builder-C V1.08 (WEB-SERVER_EXAMPLE.WSC)

Project: WEB-SERVER_EXAMPLE.WSP

INPUT.HTM
OUTPUT.HTM
PCD_PROCESS.HTM
START.HTM
V_HELP.HTM

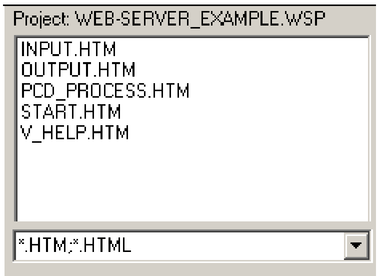
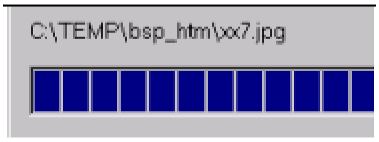
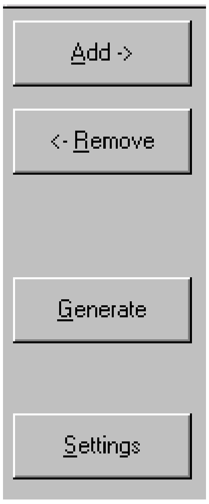
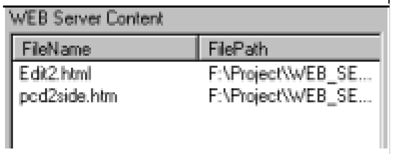
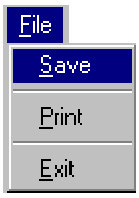
.HTM;.HTML

WEB Server Content

FileName
input.htm
Output.htm
pcd_process.htm
pcd2_m170s3.jpg
start.htm
v_help.htm
Webserver_nt_cs.jpg

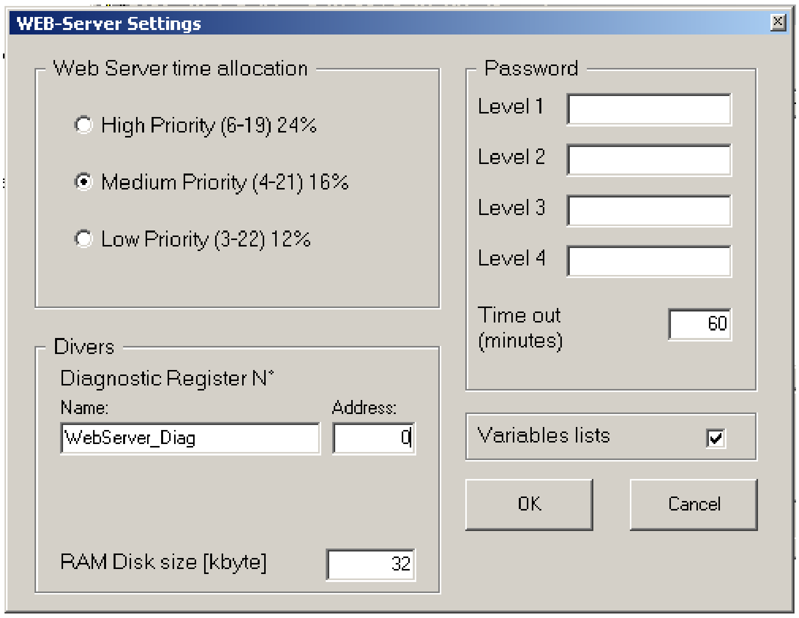
Source generated

Individual screen components are:

1	**Select window:** Only the files contained in the folder “html“ of your CPU, are displayed here. See next chapter “location of the Web-Server files“. The following display masks can be selected: *.HTM, *.HTML: to show HTML files *gif, *jpg, *.bmp:to show graphics and im age files *.* to show all files	 Project: WEB-SERVER_EXAMPLE.WSP INPUT.HTM OUTPUT.HTM PCD_PROCESS.HTM START.HTM V_HELP.HTM *.HTM;*.HTML
2	Progress Bar While the files get treated and the source file in the Saia PG5® project is generated, the progress bar will show the status of the generation.	 C:\TEMP\bsp_htm\xx7.jpg
3	Function buttons, which can be used to add files or images from the select window to the translation window, or to remove them. • **Add->** Files marked in the select window are copied to the Web-Server content window. • **<-Remove** Files marked in the Web-Server content window are deleted from it. • **Generate** All files and images entered in the Web-Server content window are converted into a source file of the Saia PG5® project. • **Settings** This button will open a new window, where you can choose the settings of the Web-Server .	
4	Web-Server content window All files and images to be translated must be listed in this window. All files listed in this window will be converted into source files of the Saia PG5® project.	 WEB Server Content FileName | FilePath Edit2.html | F:\Project\WEB_SE... pcd2side.htm | F:\Project\WEB_SE...
5	**Menu Structure File** • **Save** The name and the path of the files, which were selected in the Web-Server are saved in the Saia PG5® project. • **Print** Print information belonging to current project. • **Exit** Shuts down the Web-Builder	

4.1.3 Settings in the Web-Builder

Another very important point are the settings of the Web-Server.



WEB-Server Settings

Web Server time allocation

○ High Priority (6-19) 24%

● Medium Priority (4-21) 16%

○ Low Priority (3-22) 12%

Password

Level 1

Level 2

Level 3

Level 4

Time out (minutes): 60

Divers

Diagnostic Register N°

Name: WebServer_Diag

Address: 0

RAM Disk size [kbyte]: 32

Variables lists ☑

Web-Server Time allocation
In the Saia PCD®, only the Web-Server or the Saia PCD® user software can run at the same time, this means that when the Saia PCD® user program is running, the Web-Server will stay in stand by and vice-versa. So this could generate some delay in the response time of the Web-Server. Now depending what has more priority, you can set here the priority of the Web-Server and the user program execution.

- High priority, 24 % of the Saia PCD® time will be spent for the Web-Server, so 76 % for the user program.
- Medium priority, 16 % of the Saia PCD® time will be spent for the Web-Server , so 84 % for the user program.
- Low priority, 12 % of the Saia PCD® time will be spent for the Web-Server , so 88 % for the user program.

Diagnostic Register
The Diagnostic Register will actually count the number of requests treated by the Web-Server. It is a live counter. Value different than zero in this register will indicate that the Web-Server has been correctly started. Here, in the configuration, you can either specify a symbol and the register number or only the symbol, then the register number will be allocated automatically. If specifying the register number, be careful not to specify it in the range of the auto-allocation area.
See the **Saia PG5®->CPU->Software Settings**…

RAM Disk
The size of the Ram Disk is related to the maximum number of PDP requests supported by the web server. Increase the size in case of Ram Disk overflow history entry.

Passwords

The passwords can be used to limit the access to the Web-Server or to some functionality. In fact, 4 levels of access have been defined. It’s possible to protect each level by a password.
Here are the levels and their possibilities.

- Password level 1
  This level allows to access html pages, without the possibility to read the Saia PCD® data point (media as Register, Flag, etc…..)
- Password level 2
  This level allows to access html pages and the possibility to read the Saia PCD® data point (media as Register, Flag, etc…..)
- Password level 3
  This level allows to access html pages and the possibility to read and write the Saia PCD® data point (media as Register, Flag, etc…..)
- Password level 4
  This level gives access to all the possibilities of the level 3, but also the Variable lists.
- Password Time out
  Defines the time after which the password will have to be re-entered, if no activity has been detected on the Web-Server.

If a password is defined for a certain level, it will protect this level until the next level which has a defined password, e.g.:
A password is defined in level 2 and 4. Then the password protects the level 2 and 3. If then the user enters password level 2, the Web-Server gives access up to level 3.

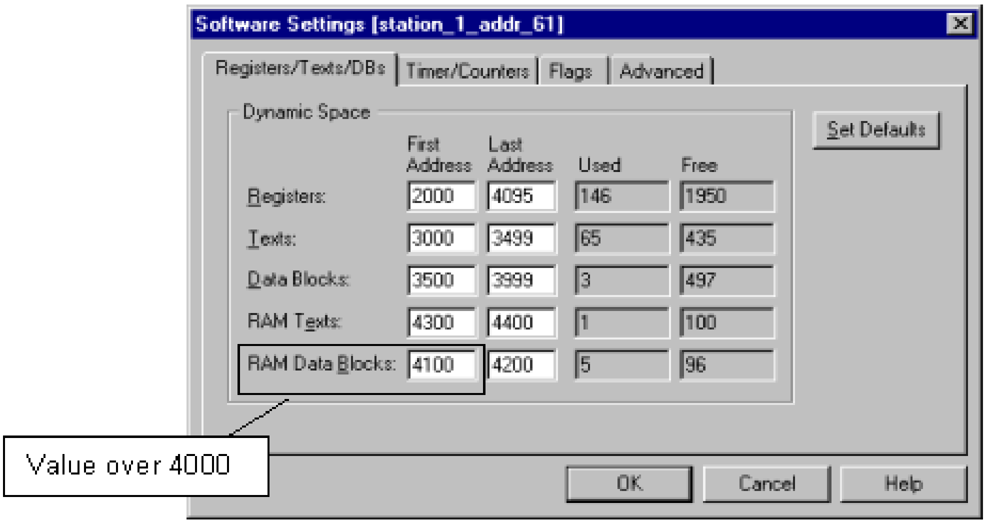
Variables lists

Enable or disable the functionality to save the list of media selected in the varlist in the run time. If this box is not checked it will not be possible at any time to save the media list.

Checking this box will create dynamically Data Block in the PLC software, this is done during the Build in Saia PG5®. Only Data block over 4000 are supported by the Web-Server. So make sure that in the Saia PG5® the area of dynamic allocation is over 4000 !

In the picture next page, in the Saia PG5® software:

- Select the CPU of your project concerned by the Web-Server.
- In the menu “**CPU**“, select the function “**Software Settings…**“, this window will appear.
- RAM data blocks have to be set over 4000 !



Software Settings [station_1_addr_61]

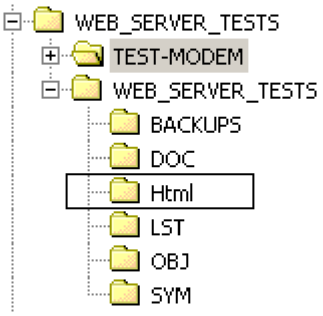
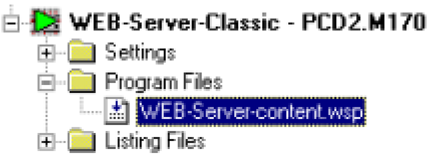
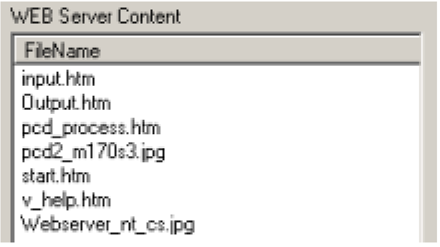
Dynamic Space

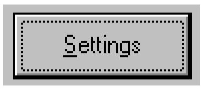
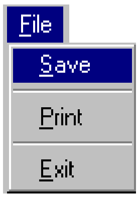
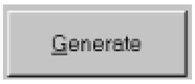
	First Address	Last Address	Used	Free
Registers:	2000	4095	146	1950
Texts:	3000	3499	65	435
Data Blocks:	3500	3999	3	497
RAM Texts:	4300	4400	1	100
RAM Data Blocks:	4100	4200	5	96

Value over 4000

4.1.4 Using Web-Builder

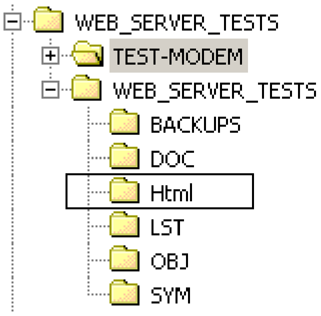
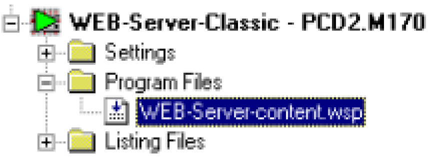
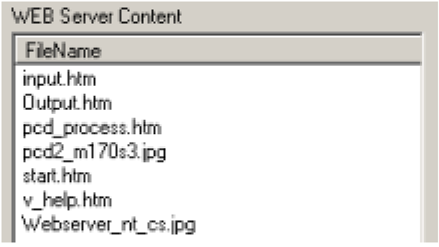
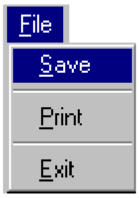
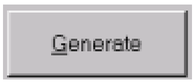
Files and images are converted and saved using the Web-Builder.
The following procedure can be used for this purpose:

1	With a file utility software, like the "windows commander" or the "Explorer", create a sub-folder, called "html", inside the CPU concerned by the Web-Server.. Usually the default path, where all the Saia PG5® projects are placed, is : "**C:\Saia PG5® Projects**" Inside this folder you will find all the Saia PG5® projects you have made. Go into your project and also go inside the CPU concerned by the Web-Server. Like here in this example it is the Project : WEB_SERVER_TESTS . CPU: WEB_SERVER_TESTS. In this folder we create the subfolder "html" as shown in the drawing. Then you recopy inside this new sub-folder, all the files we will use inside the Saia PCD®, e.g. html files and pictures files.	 WEB_SERVER_TESTS TEST-MODEM WEB_SERVER_TESTS BACKUPS DOC Html LST OBJ SYM
2	In the Saia PG5® project, create a **.wsp** file and then open the file. This will open the Web-Builder	 WEB-Server-Classic - PCD2.M170 Settings Program Files WEB-Server-content.wsp Listing Files
3	Select on the left side then the files you like to load into the Saia PCD® and "add" them, they will be then displayed on the right side "Web-Server Content" window.	 WEB Server Content FileName input.htm Output.htm pcd_process.htm pcd2_m170s3.jpg start.htm v_help.htm Webserver_nt_cs.jpg

4	Go into the settings Window and set, if necessary, the Passwords, the Diagnostic register number, enable Varlist function, etc..	
5	Save the configuration.	
6	Generate the source files for Saia PG5®.	

4.1.5 Web-Builder, after change is made in HTML page

After an HTML file or any file, which will be contained in the Saia PCD® have been changed, it is necessary to use again the Web-Builder to create a new source file for the Saia PG5®. The following procedure can be used for this purpose:

1	Be sure that the file you modified is in the “html“ folder of your Saia PG5® user project.	 WEB_SERVER_TESTS TEST-MODEM WEB_SERVER_TESTS BACKUPS DOC Html LST OBJ SYM
2	In your Saia PG5® project, double click on the **.wsp** file, it will open the file with the Web-Builder.	 WEB-Server-Classic - PCD2.M170 Settings Program Files WEB-Server-content.wsp Listing Files
3	If new files have to be added, then select them on the left side and add them to the Saia PCD® content.	 WEB Server Content FileName input.htm Output.htm pcd_process.htm pcd2_m170s3.jpg start.htm v_help.htm Webserver_nt_cs.jpg
4	Save the configuration.	
5	Generate the source files for Saia PG5®®.	

4.2 Web-Connect

For details please refer to the manual 26-800 Web-Connect. This document can be found on SBC support site www.sbc-support.com under Software.

4.3 Variable pages

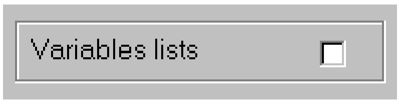
The Web-Server includes 4 variable pages that can be used to access and write to any PLC data point during run-time. Variable pages are suitable for service and debug purposes and correspond to the Display and Refresh value of the Saia PG5® debugger.

Each variable page can have up to 16 free definable process data points. For each process data point a free text can also be defined. Passwords protect the functionality of reading and writing process data points and of saving variable pages.

4.3.1 Before use the Variable list

Data Blocks

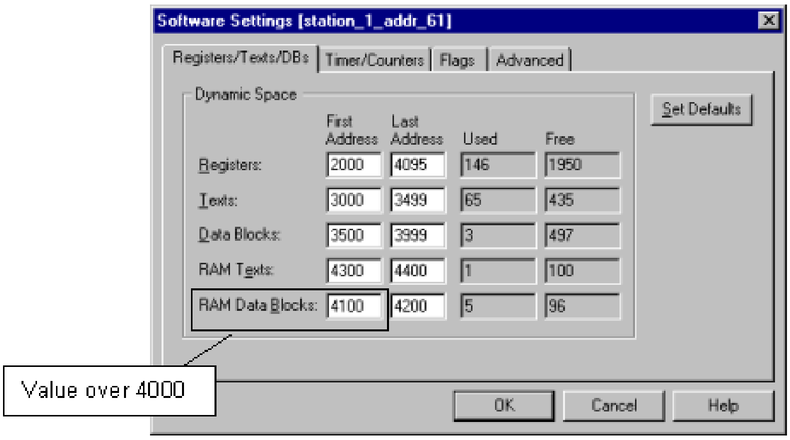
When Variable list pages are stored, they are stored in Data Blocks of the user program. So in order to be able to use this functionality, you first need to create those Data Blocks, this is done in the Web-Builder Settings.
By checking the Box "Variables Lists"



Variables lists □

If this box is checked, Data Blocks will be created during the build of the Saia PG5® project. But be aware that only Data Blocks over 4000 are suitable for the Web-Server, so make sure that in the Saia PG5® settings, the DB RAM area is over 4000.

In Saia PG5® Menu **CPU, Software Settings…**



Software Settings [station_1_addr_61]

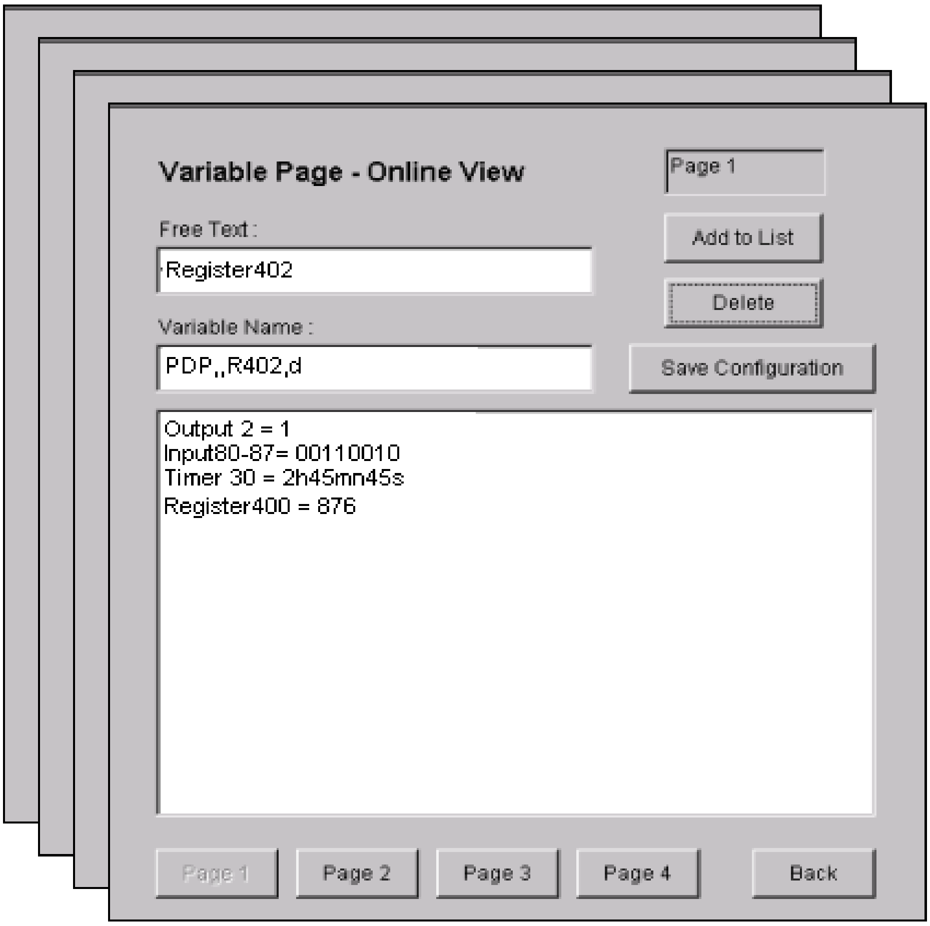
Dynamic Space

	First Address	Last Address	Used	Free
Registers:	2000	4095	146	1950
Texts:	3000	3499	65	435
Data Blocks:	3500	3999	3	497
RAM Texts:	4300	4400	1	100
RAM Data Blocks:	4100	4200	5	96

Java Applet

The Variable list is using an Java Applet, so before using it, this need to be available. The file name is “**varlist.jar**“. This file will be copied in the local directory (the default path is “C:\WebPages”), when installing the Web-Connect software.

- If Variable list pages are to be stored in the PLC’s user memory, a memory range must first be reserved. This takes place in Web-Builder, Settings window, and the check box Variables lists.
- DB RAM area in the Saia PG5® should be over 4000.
- Variable list applet “varlist.jar” should be in the local directory of the PC



Variable Page - Online View

Free Text : Register402

Variable Name : PDP,,R402,d

```
Output 2 = 1
Input80-87= 00110010
Timer 30 = 2h45mn45s
Register400 = 876
```

Free Text field	Free text about the process data point
Variable Name field	Process data point with PDP key (without % character)
Add to List button	Free text and PDP key are added to the list
Delete button	Selected list entry is deleted
Save Configuration button	Current variable page entries are saved to the PLC (only possible for password level 4 and if the appropriate space has been reserved in PLC memory).
Page1…Page 4	Switch to the other variable pages
Back	Back to start page.

Here is the description of the tag addressing for Saia PCD® media.
It starts with the PDP identification, which means Saia PCD® Data Point, then the characters “,,“. After these comes the Media area and the address of the media. Depending on the media type it is possible to have a range of them, so numbers of elements can be specified, separated with the address by “/“. At the end, separated with a coma, comes the representation format.
The tag is not case sensitive, it can be either in minuscule or capital letters.

PDP,,[Media type][element][/count][,format]

Media type	ID of the media	Element	Count (optional)	Possible format
Input or Output	I O	Address of the input or the output	Number of elements, max 32	D,Y,H
Flag	F	Address of the flag	Number of elements, max 32	D,Y,H
Register	R	Address of the register		D,Y,H,F
Timer	T	Timer number		D,Y,H,T
Counter	C	Counter number		D,Y,H
Text	X or TEXT	Text number		A
Saia PCD® Status	STATUS			A

For the Data Blocks it is different, because we need one more information. Which is the element number in the DB. Here is how the syntax is for this tag.
PDP,,[DB][**Address**][**.element**][**,format**]

Media type	ID	address	Element	Possible format
Data Block	DB	DB number	Element number	D,Y,HF

The formats are:

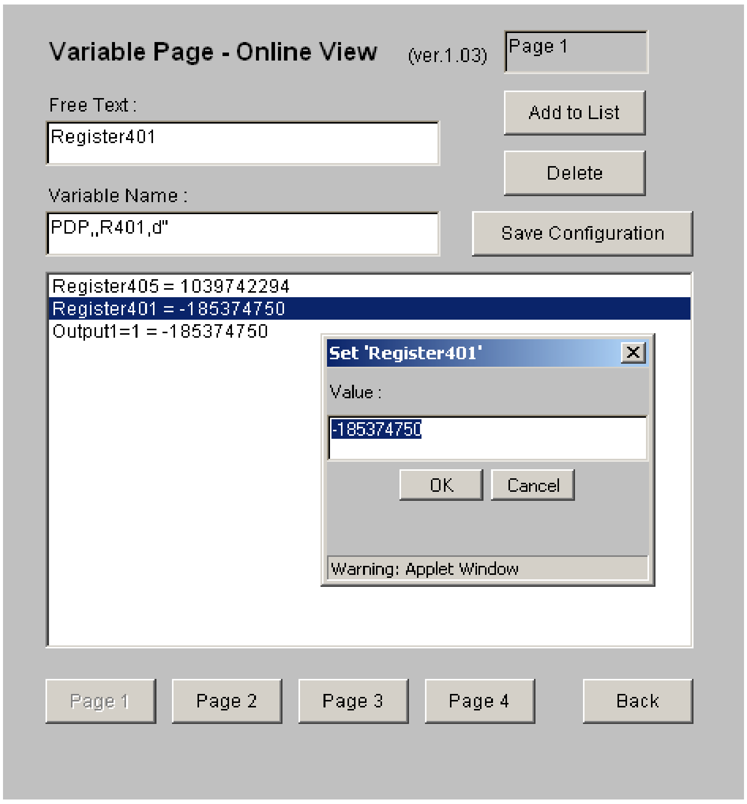
Format ID	description
Y or B	BINARY
c	Decimal with Coma, all numbers are devided by 10 (304 is displayed as 30,4)
D	DECIMAL SIGNED
H or X	HEXADECIMAL
A or S	Text
i	Interpreted Text, only if it needs to be interpreted
k	Clock FBox, format from the FBox CVC. Only for registers.
F	Floating point
T	Timer **[..h..mn..s]** ..hours..minutes..secondes

Examples:

- PDP,,I16,B is the
  same as PDP,,I16,Y addressing one input I16 and format is binary.
- PDP,,O32/8,D addressing 8 outputs from the O32, format is decimal
- PDP,,F10/16,H addressing 16 flags from the F10, format is hexadecimal
- PDP,,R300,F addressing the register 300, format is in floating point
- PDP,,T50,T addressing Timer 50, format in hours, minutes, seconds
- PDP,,C99,D addressing Counter 99, format is decimal
- PDP,,TEXT40,A addressing the TEXT40, format is always string
- PDP,,X40,A addressing the TEXT40, format is always string
- PDP,,DB10.20,D addressing DB10, element 20, format is decimal
- PDP,,STATUS,A actual status of the Saia PCD®, it will return the status, firmware version and Saia PCD® type.

4.4 Modifying entries in a variable page

Process data points in variable pages can be modified by double-clicking on the specific process data point.



Variable Page - Online View (ver.1.03)

Page 1

Free Text : Register401

Variable Name : PDP,,R401,d"

Register405 = 1039742294
Register401 = -185374750
Output1=1 = -185374750

Set 'Register401'

Value : -185374750

Warning: Applet Window

4.5 Access to Saia PCD® data

If you wish to access PLC process data points from your HTML page, these process data points must be defined with a special process data point key (PDP key) in the HTML page. PDP keys can be defined in HTML pages or in control functions.
A PDP key is as described in the above chapter, but the tag is surrounded now by the character “%”. Two characters “%%“ at the beginning of the TAG and one “%%“ at the end.

- In HTML pages, PDP keys must be delimited by “%”.
  In control functions, “%“ delimiters are not needed.
- If the PDP key has been incorrectly defined, a read operation will result in the display of an error message rather than the data.
  For a write operation, the value will not be written to the PLC.
- The PDP key is only evaluated for pages with the suffix *.html or *.htm.
  If a PDP key is stored in any other type of file, they are not considered and so no value will be displayed.
- The PDP key is not case sensitive for the media information, but the PDP has to be in capital letters.

Examples:

• %%PDP,,I16,B% is the same as %%PDP,,I16,Y%	addressing one input “I16” and format is binary.
• %%PDP,,R300,F%	addressing the register 300, format is in floating point
• %%PDP,,DB10.20,d%	addressing DB10, element 20, format is decimal

4.5.1 Access with Symbols in the tags

It is also possible in the tags to use directly the Saia PG5® symbols. But this only in the HTML pages, it's not possible for the Variable list functions.

In fact this is possible, thanks to the Web-Builder and the Saia PG5® compiler, which will replace at the compilation time, the symbols by the absolute address of the media. The only condition is that the symbols are declared as **Global** in the Saia PG5®.

Then, in the tags, the media area identification and the address can be replaced by the symbol. To identify that it is a symbol, it has to be preceded by the character "&".

Here are some examples.

Let's have the following symbols:

Register 1000	R1000 Motor_1_temperature	
Input 0	I0	Motor_1_alarm
Timer 50	T50	Motor_1_time_on

So if we would use it in the html page, the tag will be as follows:

Media	Normal tag	Tag with symbol
Register 1000	%%PDP,,R1000,D%	%%PDP,,&Motor_1_temperature,D%
Input 0	%%PDP,,I0,Y%	%%PDP,,&Motor_1_alarm,Y%
Timer 50	%%PDP,,T50,T%	%%PDP,,&Motor_1_time_on,T%

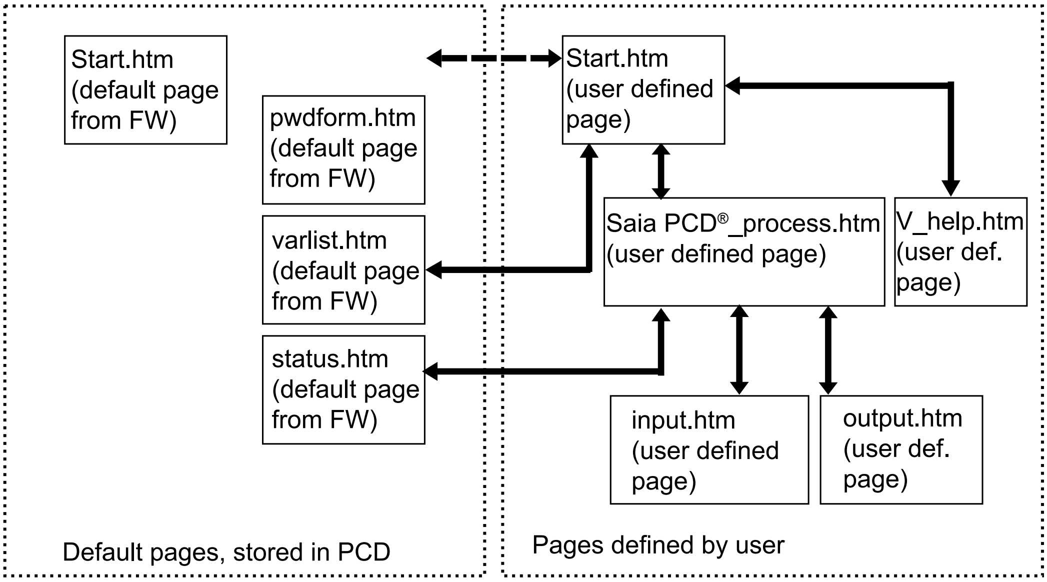
It is also possible to specify an array of inputs, lets say we like to read the input 0 to input 7.

Input 0 to 7	%%PDP,,I0/8,Y%	%%PDP,,&Motor_1_alarm/8,Y%

5 Example Web-Project

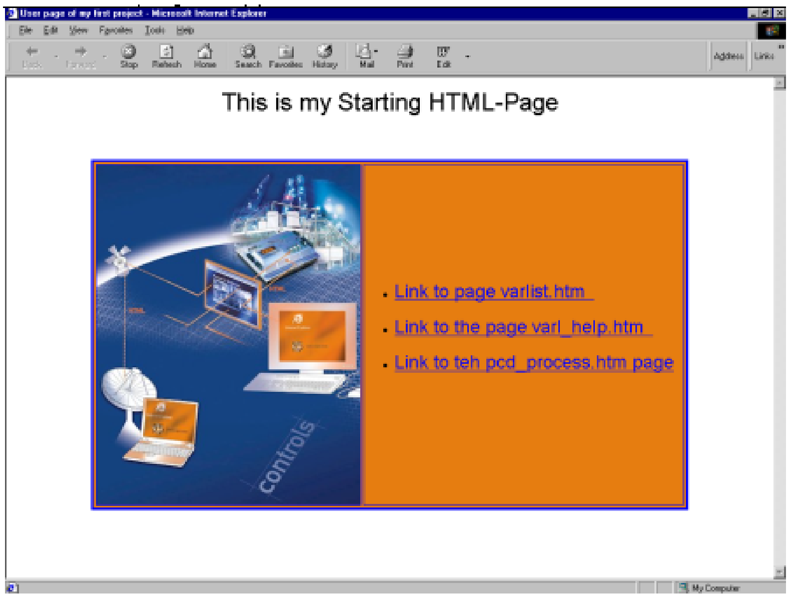
5.1 Structure of pages

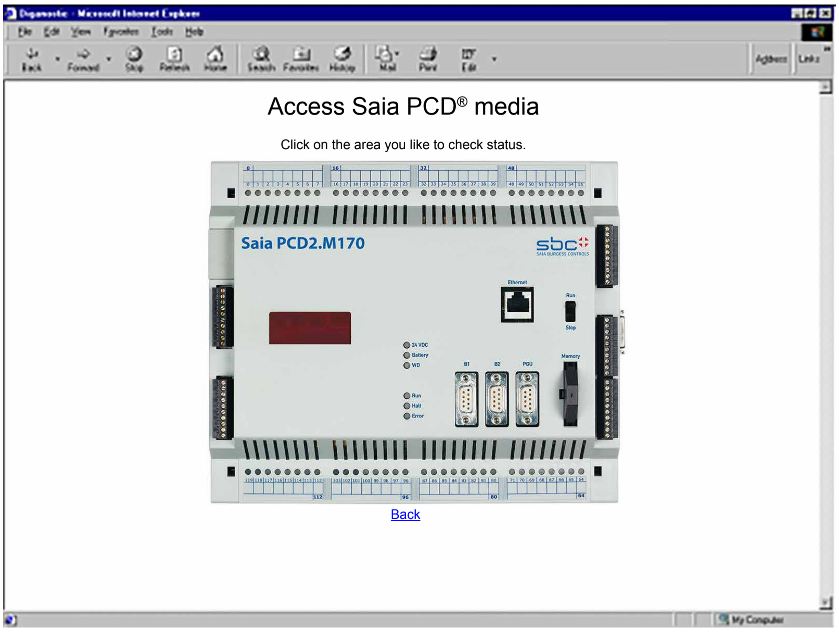
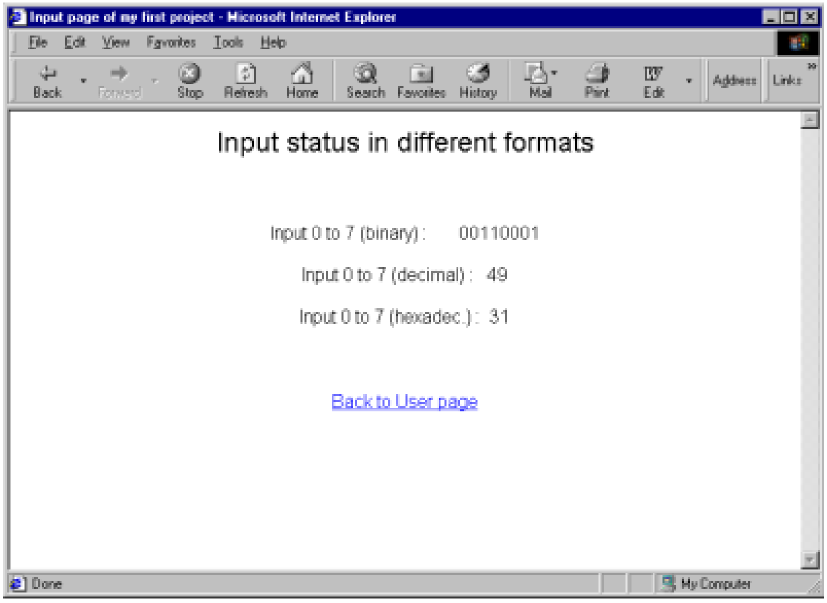
In the following example four HTML pages are to be defined and downloaded to the PLC. The structure is as follows:

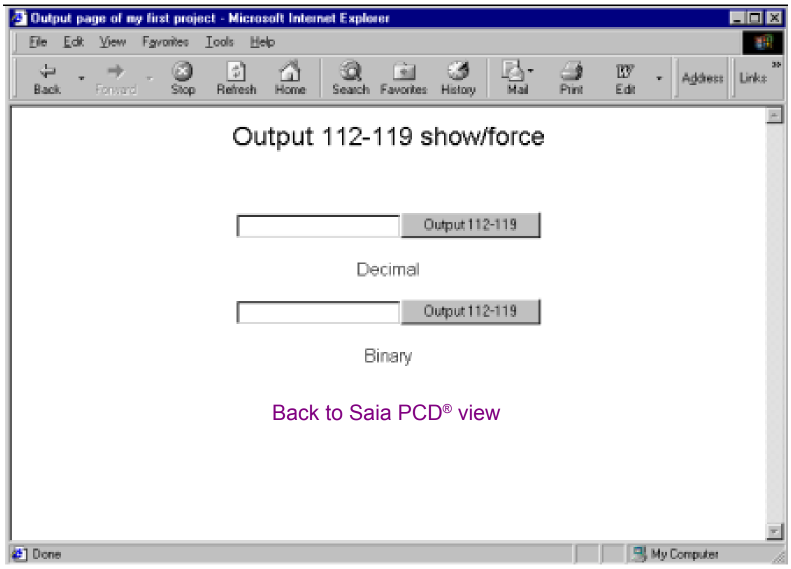

Start.htm (default page from FW)
pwdform.htm (default page from FW)
varlist.htm (default page from FW)
status.htm (default page from FW)
Default pages, stored in PCD
Start.htm (user defined page)
Saia PCD® _process.htm (user defined page)
V_help.htm (user def. page)
input.htm (user defined page)
output.htm (user def. page)
Pages defined by user

We will not show here, how to create all the HTML pages, but only the interesting ones. We mean by interesting, the ones which have Saia PCD® media in it.
So we will describe in detail how to do the input.html and the output.html pages. All the pages of this project can be found in the Saia PG5® project "WEB_doc_project".

The four pages defined in the user program appear as following in the Web-Browser:

Page: start.htm Start page, with a picture and links to the pages: varlist.htm, pcd_ process.htm and V_help.htm	 User page of my first project - Microsoft Internet Explorer This is my Starting HTML-Page • Link to page varlist.htm • Link to the page varl_help.htm • Link to teh pcd_process.htm page

Page: pcd_process.htm is a visual selection page, which allows access to the pages: input.htm output.htm status.htm	 Access Saia PCD® media Click on the area you like to check status. Back
Page: input.html Displays the actual status of the inputs 0 to 7	 Input page of my first project - Microsoft Internet Explorer File Edit View Favorites Tools Help Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Address Links Done My Computer Input status in different formats Input 0 to 7 (binary) : 00110001 Input 0 to 7 (decimal) : 49 Input 0 to 7 (hexadec.) : 31 Back to User page

Page: output.htm Display and modification of outputs 112 to 119	 Output page of my first project - Microsoft Internet Explorer Output 112-119 show/force Output 112-119 Decimal Output 112-119 Binary Back to Saia PCD® view Done My Computer

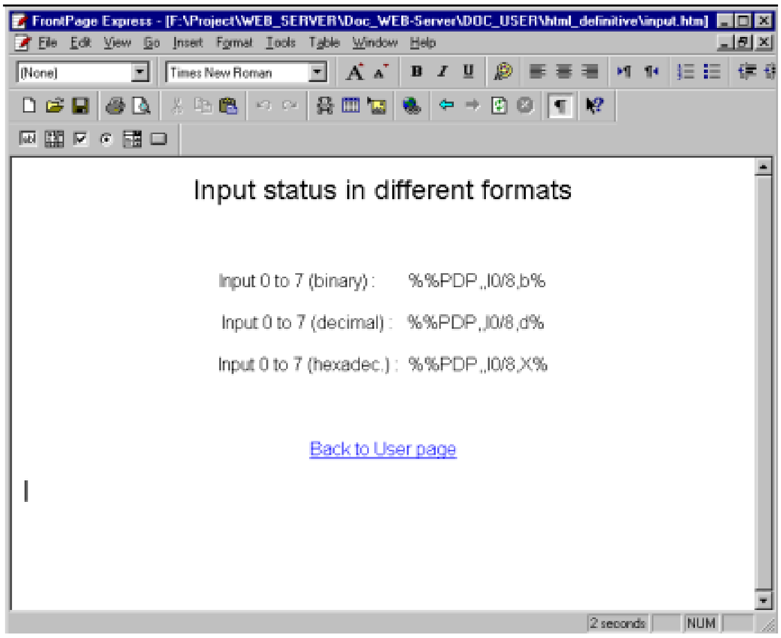
5.2 Saia PCD® data point into the HTML Pages

We will show here how to introduce in your html page a dynamic area, where value of the Saia PCD® media can be displayed. We will show it with the FrontPage express editor, but of course you can use any HTML editor.

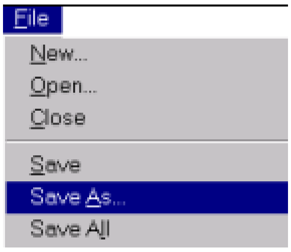
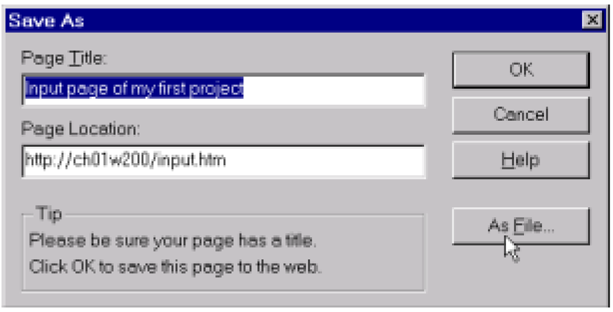
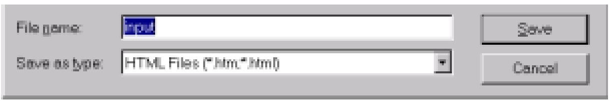
Start up an HTML editor, for example FrontPage Express.

5.2.1 Page: input.htm

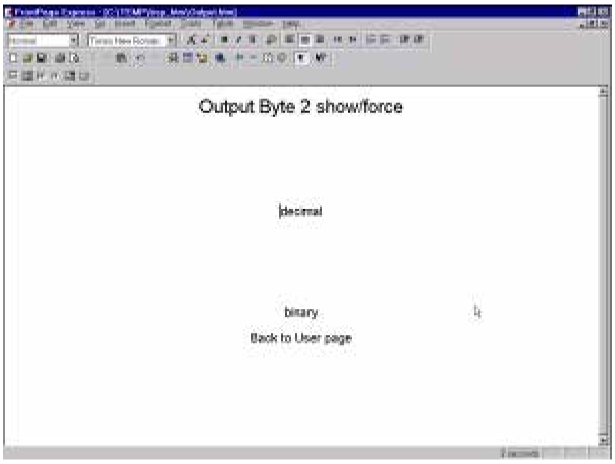
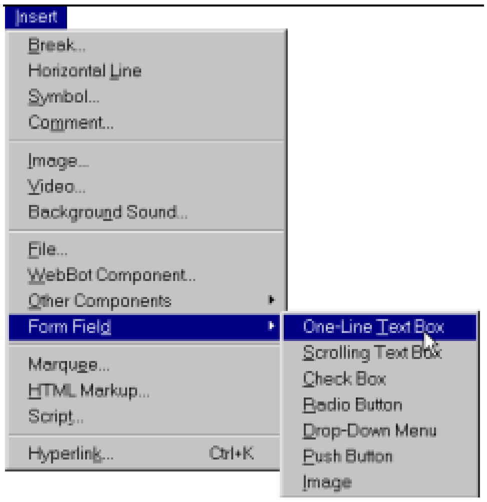
In this page we demonstrate how easy it is to put some fields in a page to show the value of a media, at the time the HTML page is loaded from the Saia PCD®.
The example shows inputs, but any area can be displayed like this, see the chapter “Variable pages“, to check the right syntax.

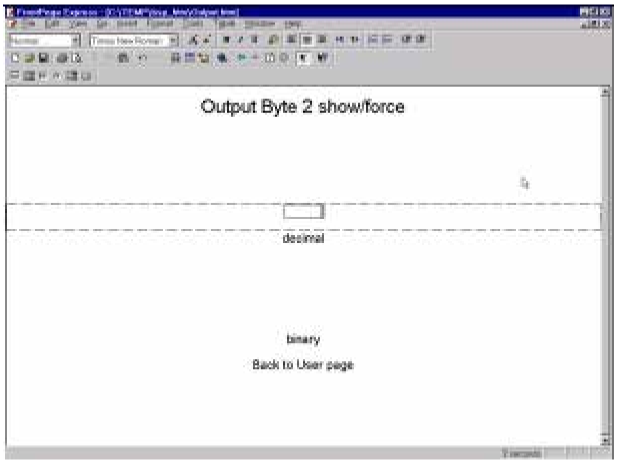
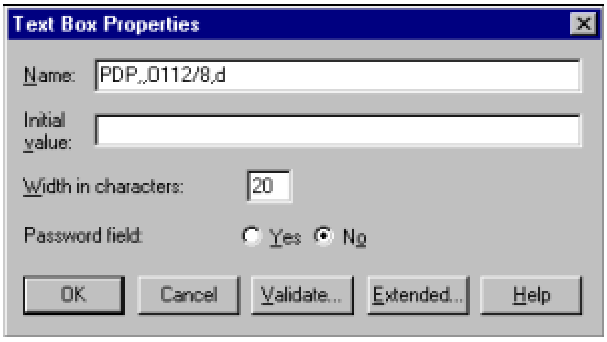
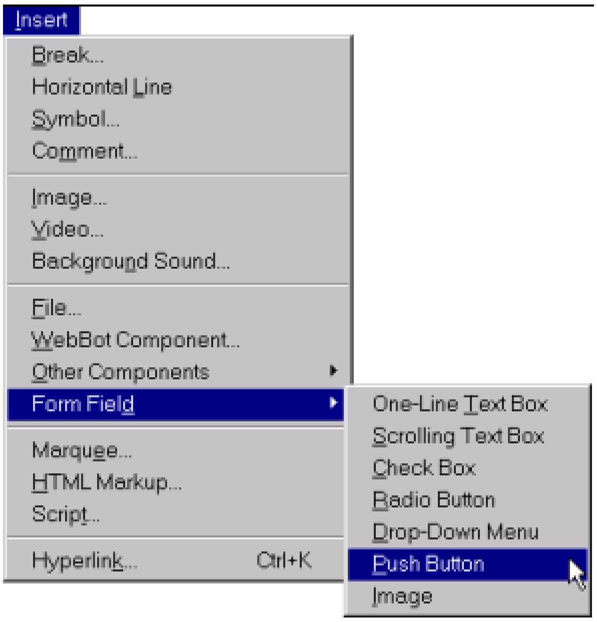
1	Open a new, empty page. Insert text and position text. Note syntax for PDP key: %%PDP,,I0/8,b% %%PDP,,I0/8,d% %%PDP,,I0/8,x%	 FrontPage Express - [F:\Project\WEB_SERVER\Doc_WEB-Server\DOC_USER\html_definitive\input.htm] Input status in different formats Input 0 to 7 (binary) : %%PDP,,I0/8,b% Input 0 to 7 (decimal) : %%PDP,,I0/8,d% Input 0 to 7 (hexadec.) : %%PDP,,I0/8,X% Back to User page 2 seconds NUM

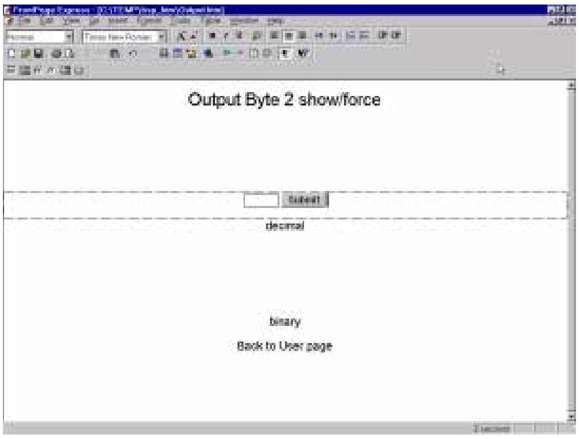
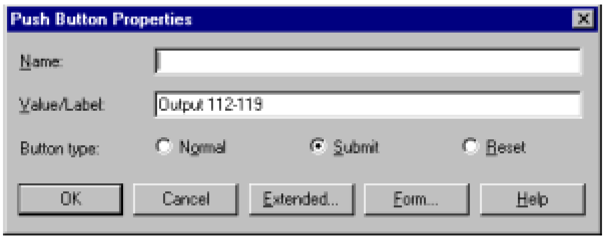
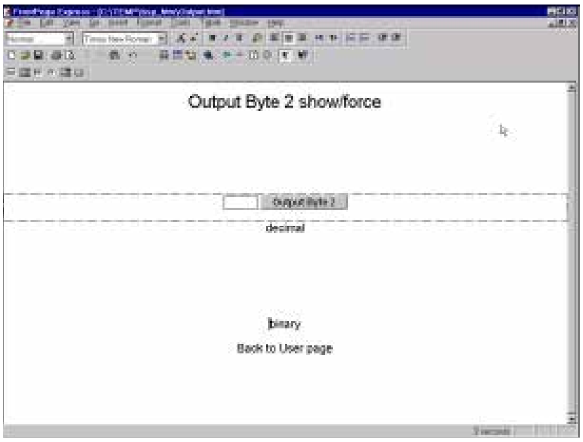
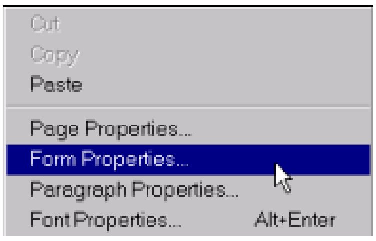
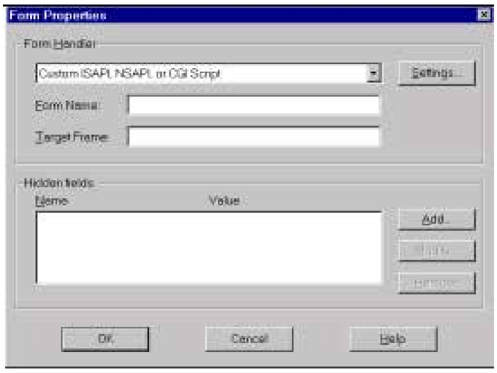
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Address Links
File Edit View Go Insert Format Tools Table Window Help
(None) Times New Roman

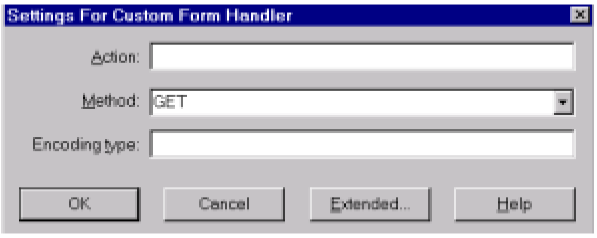
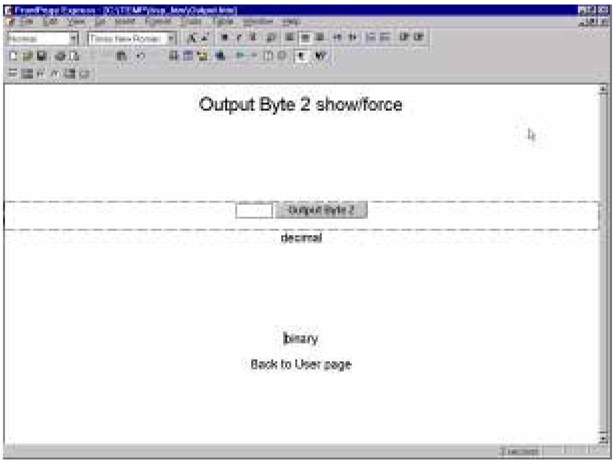
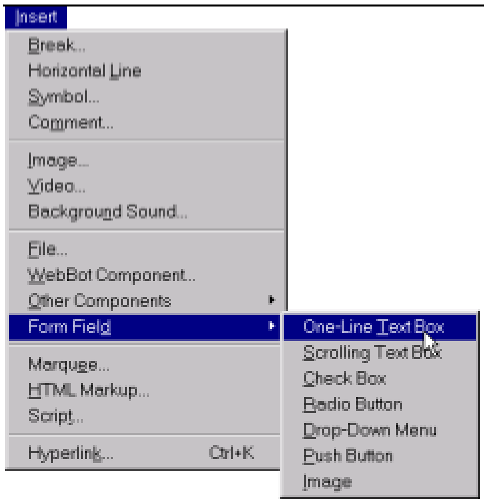
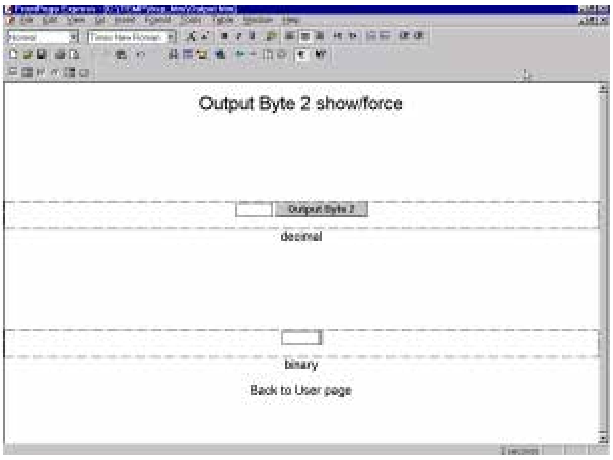
2	Save file with the Save option.	 File New... Open... Close Save Save As... Save All
3	Enter page title correctly. Select the **As File** option	 Save As Page Title: Input page of my first project Page Location: http://ch01w200/input.htm Tip Please be sure your page has a title. Click OK to save this page to the web.
4	Enter file name: **input.** Then save.	 File name: input Save as type: HTML Files (*.htm;*.html)

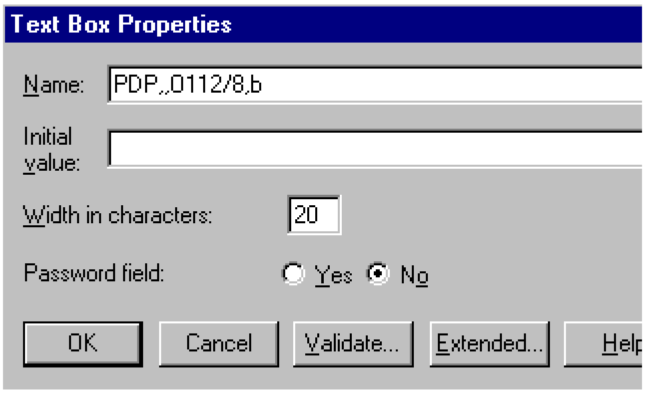
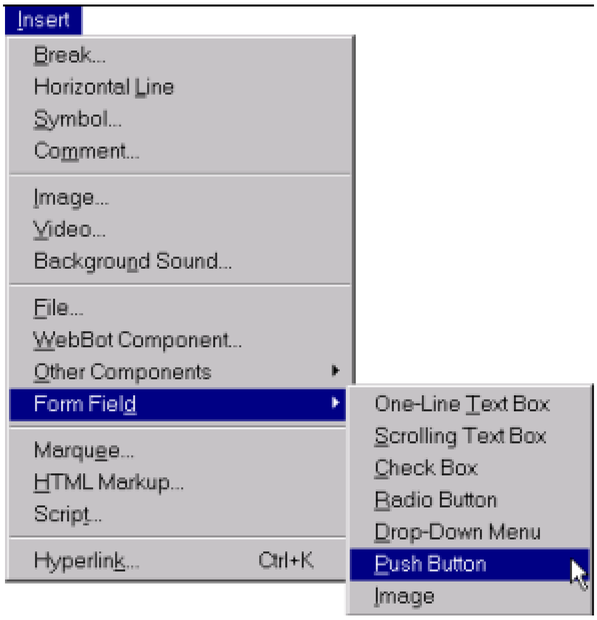
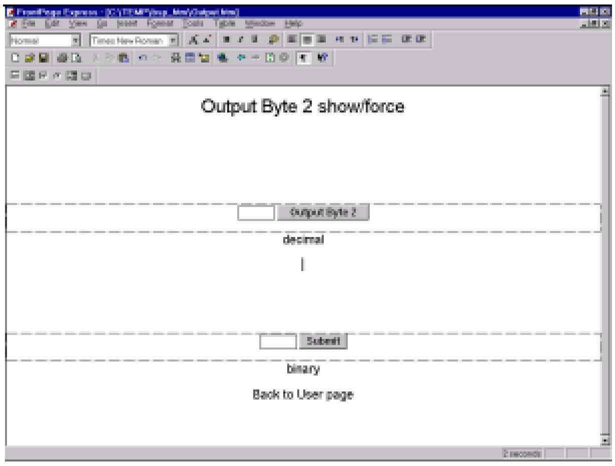
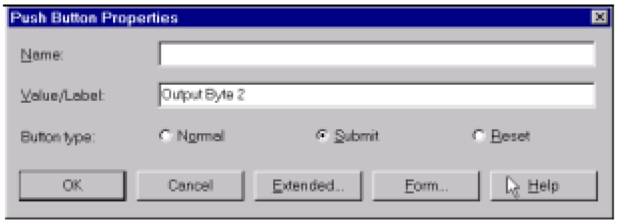
5.2.2 Page: output.htm

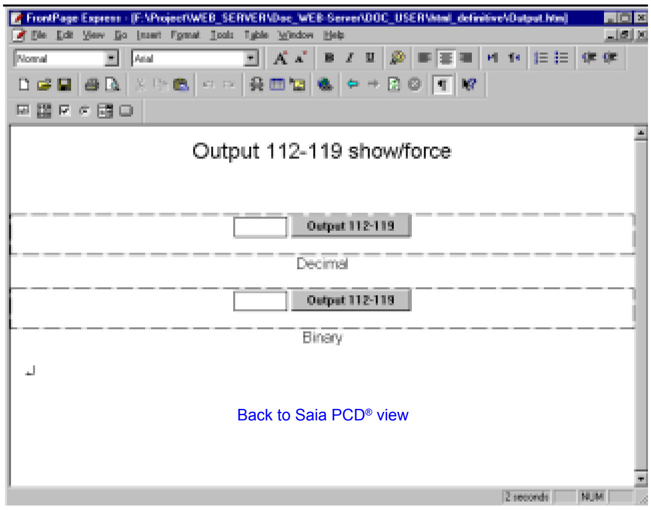
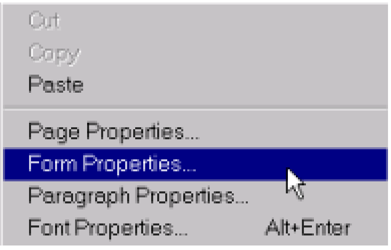
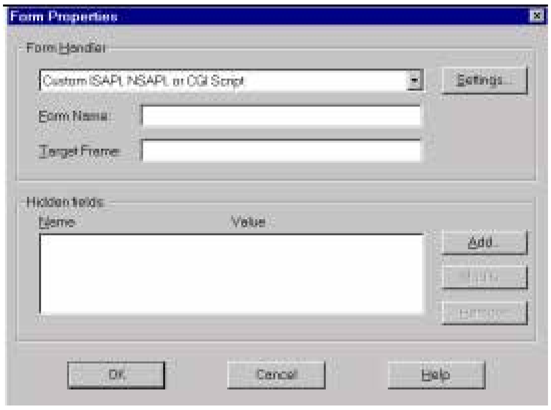
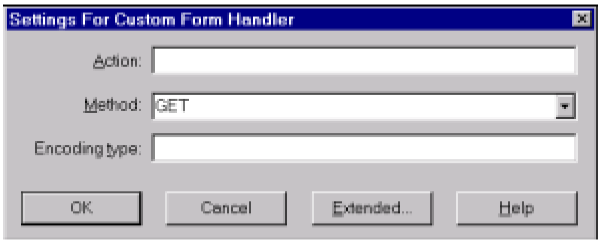
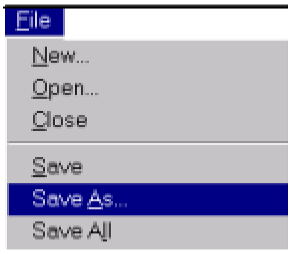
1	Open a new, empty page. Insert text and position text.	 Output Byte 2 show/force decimal binary Back to User page
2	Insert input/output fields: Position cursor over the text decimal. Insert field with: Insert, FormField, One-Line Text Box	 Insert Break... Horizontal Line Symbol... Comment... Image... Video... Background Sound... File... WebBot Component... Other Components Form Field Marquee... HTML Markup... Script... Hyperlink... Ctrl+K One-Line Text Box Scrolling Text Box Check Box Radio Button Drop-Down Menu Push Button Image

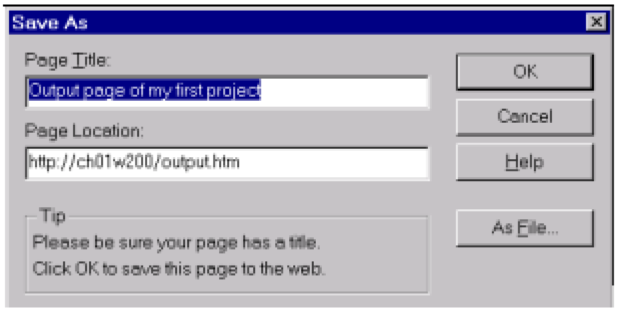
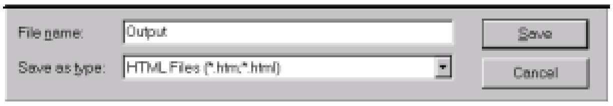
3	Position the field	 Output Byte 2 show/force decimal binary Back to User page
4	Double-click on the field. Enter the PDP key Note the correct syntax: PDP,,O112/8,d It means, from Output 112, take 8 outputs following.	 **Text Box Properties** Name: PDP,,0112/8,d Initial value: Width in characters: 20 Password field: ○ Yes ● No OK Cancel Validate... Extended... Help
5	Insert the Submit Button. Position cursor near entry field in the same frame (Form Field). Insert, Form Field, Push Button	 Insert Break... Horizontal Line Symbol... Comment... Image... Video... Background Sound... File... WebBot Component... Other Components Form Field Marquee... HTML Markup... Script... Hyperlink... Ctrl+K One-Line Text Box Scrolling Text Box Check Box Radio Button Drop-Down Menu Push Button Image

6	Double-click on button.	 Output Byte 2 show/force
7	Insert text which will appear on the button.	 Push Button Properties Name: Value/Label: Output 112-119 Button type: ○ Normal ● Submit ○ Reset
8	The page now appears as follows:	 Output Byte 2 show/force
9	Position cursor to right of the button and click the right-hand mouse button. Select the Form Properties option	 Cut Copy Paste Page Properties... Form Properties... Paragraph Properties... Font Properties... Alt+Enter
10	Select the Settings option	 Form Properties Form Handler: Custom ISAPI, NSAPI, or CGI Script Form Name: Target Frame: Hidden fields: Name Value

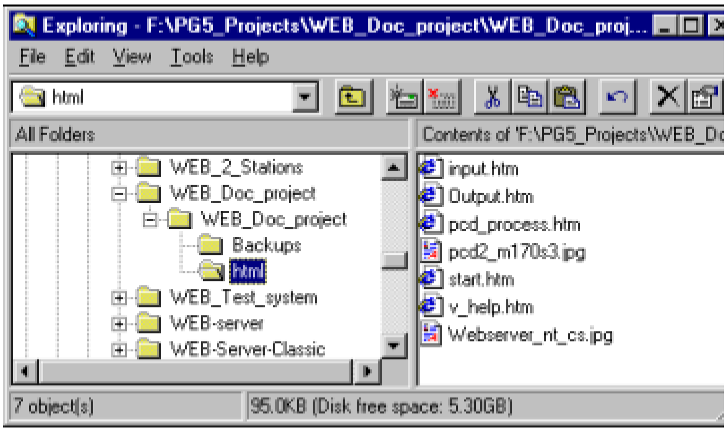
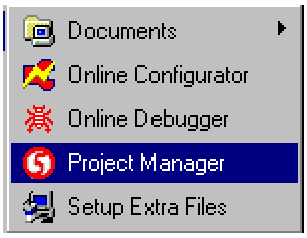
11	Method: GET Select Close window	 Settings For Custom Form Handler Action: Method: GET Encoding type:
12	The page now appears as follows: The same procedure as set out in steps 2 to 11 should now follow for the binary value entry field.	 Output Byte 2 show/force Output Byte 2 decimal binary Back to User page
13	Insert input/output fields: Position cursor over the text decimal. Insert field with: Insert, FormField, One-Line Text Box	 Insert Break... Horizontal Line Symbol... Comment... Image... Video... Background Sound... File... WebBot Component... Other Components Form Field Marquee... HTML Markup... Script... Hyperlink... Ctrl+K One-Line Text Box Scrolling Text Box Check Box Radio Button Drop-Down Menu Push Button Image
14	Position field	 Output Byte 2 show/force Output Byte 2 decimal binary Back to User page

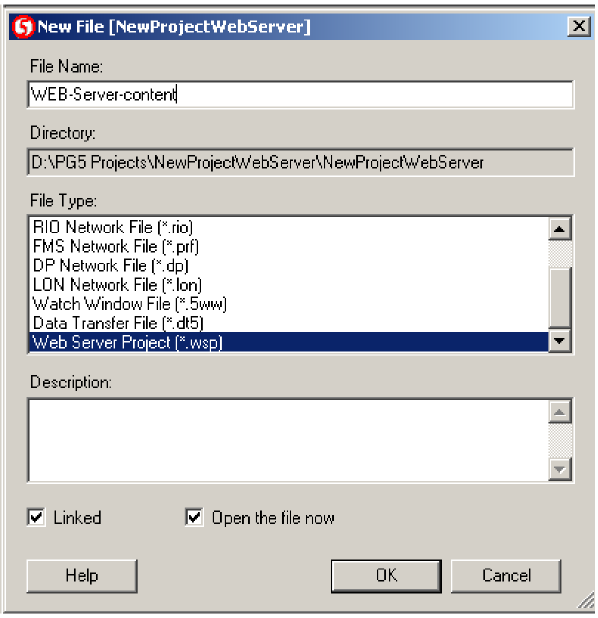
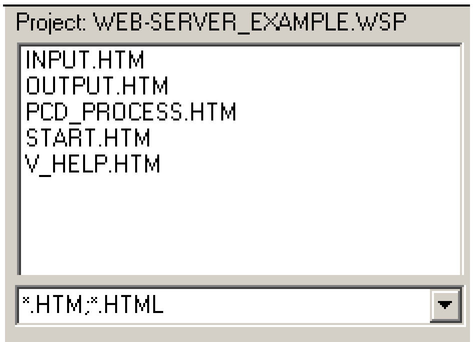
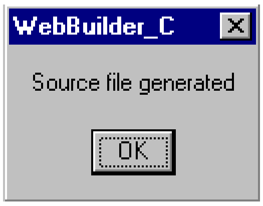
15	Double-click on the field. Enter the PDP key. Note correct syntax: PDP,,O112/8,b	 Text Box Properties Name: PDP,,0112/8,b Initial value: Width in characters: 20 Password field: ○ Yes ● No
16	Insert Submit Button. Position cursor near the entry field in the same frame (Form Field). Insert, Form Field, Push Button	 Insert Break... Horizontal Line Symbol... Comment... Image... Video... Background Sound... File... WebBot Component... Other Components Form Field Marquee... HTML Markup... Script... Hyperlink... Ctrl+K One-Line Text Box Scrolling Text Box Check Box Radio Button Drop-Down Menu Push Button Image
17	Double-click on button	 Output Byte 2 show/force Output Byte 2 decimal Submit binary Back to User page
18	Insert text to appear on button	 Push Button Properties Name: Value/Label: Output Byte 2 Button type: ○ Normal ● Submit ○ Reset

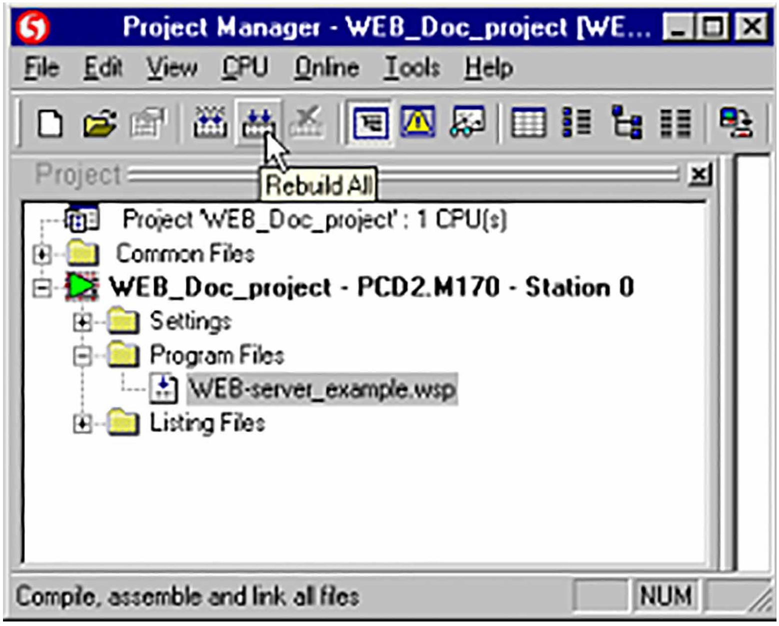
19		 FrontPage Express - [F:\Project\WEB_SERVER\Doc_WEB-Server\DOC_USER\html_definitive\Output.htm] File Edit View Go Insert Format Tools Table Window Help Output 112-119 show/force Output 112-119 Decimal Output 112-119 Binary Back to Saia PCD® view 2 seconds NUM
20	Position cursor to right of the button and click right-hand mouse button. Select the Form Properties option.	 Cut Copy Paste Page Properties... Form Properties... Paragraph Properties... Font Properties... Alt+Enter
21	Select the Settings option.	 Form Properties Form Handler Settings... Form Name: Target Frame: Hidden fields Name Value Add... Modify... Remove OK Cancel Help Custom ISAPI, NSAPI, or CGI Script
22	Method: Select GET Close window	 Settings For Custom Form Handler OK Cancel Extended... Help Action: Method: GET Encoding type:
23	Save file with the Save option.	 File New... Open... Close Save Save As... Save All

24	Enter page title correctly. Select the **As File** option	 Save As Page Title: Output page of my first project Page Location: http://ch01w200/output.htm Tip Please be sure your page has a title. Click OK to save this page to the web.
25	Enter file name: **output.** Then save.	 File name: Output Save as type: HTML Files (*.htm;*.html)

5.3 Importing the Web-Site into the Saia PCD®

1	Copy the HTML pages and pictures you created into the Saia PG5® project, create in the right CPU, a sub-directory with the name “**html**”. See chapter **4.1.1 Location of the Web-Server files**	 Exploring - F:\PG5_Projects\WEB_Doc_project\WEB_Doc_proj... File Edit View Tools Help html All Folders WEB_2_Stations WEB_Doc_project WEB_Doc_project Backups html WEB_Test_system WEB-server WEB-Server-Classic Contents of 'F:\PG5_Projects\WEB_Dc input.htm Output.htm pcd_process.htm pcd2_m170s3.jpg start.htm v_help.htm Webserver_nt_cs.jpg 7 object(s) 95.0KB (Disk free space: 5.30GB)
2	Start the Saia PG5®	 Documents Online Configurator Online Debugger Project Manager Setup Extra Files

3	Open the Saia PG5® project. Then create a new “program file“ of the “Web-Server Project (.wsp)“ type. Then open it by double-click on this new file.	 New File [NewProjectWebServer] File Name: WEB-Server-content Directory: D:\PG5 Projects\NewProjectWebServer\NewProjectWebServer File Type: RIO Network File (*.rio) FMS Network File (*.prf) DP Network File (*.dp) LON Network File (*.lon) Watch Window File (*.5ww) Data Transfer File (*.dt5) Web Server Project (*.wsp) Description: ☑ Linked ☑ Open the file now Help OK Cancel
4	Select all the files concerned and don’t forget the pictures. To see them, change the filter at the bottom, from html to GIF, JPG, BMP. Add the files.	 Project: WEB-SERVER_EXAMPLE.WSP INPUT.HTM OUTPUT.HTM PCD_PROCESS.HTM START.HTM V_HELP.HTM *.HTM;*.HTML
5	Select the settings if necessary.	
6	Convert files and images	
7	When this is done, confirm the generation	 WebBuilder_C Source file generated OK

8	Make a “Rebuild All” of your Saia PG5® project and then download the Saia PG5® software into the Saia PCD®.	

Project Manager - WEB_Doc_project [WE...
File Edit View CPU Online Tools Help
Rebuild All
Project
Project 'WEB_Doc_project' : 1 CPU(s)
Common Files
WEB_Doc_project - PCD2.M170 - Station 0
Settings
Program Files
WEB-server_example.wsp
Listing Files
Compile, assemble and link all files
NUM

6 Troubleshooting

6.1 Variable List (varlist)

Q: How to access the varlist function ?

A: The URL to access the varlist function is the following :
http://pc_name/station_name/varlist.htm

Q: The varlist is not running, nothing happens !

A: Check that the Java settings of the browser are correct and that the browser version is supported for our product. See chapter : **3.3.3 The PC's cache memory**.

Q: I can't save the media configuration in the varlist page, it loses all configuration every time I change pages.

A:

1) Make sure, if you had made a change in the media list, that you saved the me dia list before you changed the page.
2) Make sure you selected the option varlist in the Web-Builder settings and that in the Saia PG5®, the dynamic allocation area for RAM DB is over 4000, see chapter : **4.1.3 Settings in Web-Builder**.

6.2 Saia PG5® error

Q: Why do I get this error while compiling a project in Saia PG5®:

```
1){Error 42: measure_perf.wsc: Line 1741: Symbol not defined: LLigth_on_off_
web
Error 165: measure_perf.wsc: Line 1748: The symbol LLigth_on_off_web is a DB
and not the type you are using int the file TAG.htm}
```

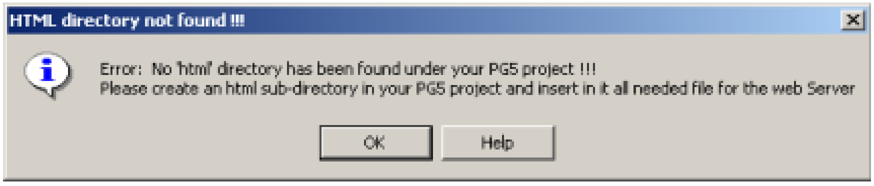
A1: A symbol that you use inside a TAG of the HTML page specified, is not declared in the Saia PG5® as a global symbol. Be also careful in this case, in fact Saia PG5® symbols are case sensitive.

```
2) Error 165: measure_perf.wsc: Line 1744: The symbol Ligth_on_off_web is not
a TEXT type as you are using it in the file TAG.htm
```

A2: There is a syntax error inside the TAG you wrote in the HTML page specified in the error message. Be careful of the type format compare to the media type, like a flag can't be displayed in string format !

6.3 Web-Builder

Q: When I create a ".wsp" file in my Saia PG5® project and try to open it, I get this window error message:



HTML directory not found !!!

Error: No 'html' directory has been found under your PG5 project !!!
Please create an html sub-directory in your PG5 project and insert in it all needed file for the web Server

A: Before you open a .wsp file, make sure you created the folder html under the Saia PG5® project, see chapter : **4.1.4 Using Web-Builder**

A Appendix A

A.1 How to create html pages without editor

This description is intended for people who wish to write html pages without editor, have more dynamic html pages with automatic update of values or manipulating the media Saia PCD® with Java applets.

Of course a good knowledge of HTML, JavaScript and Java programming is expected to understand the following chapters.

All examples have been made and tested with the Browser IE 6.0

A.2 CGI (Common Gateway Interface) of the Web-Server

In the Web-Server (inside the Saia PCD®), there are already 4 CGIs (executable) implemented:

- readVal.exe
- writeVal.exe
- OrderValues.exe
- ReadFile.exe

All these CGIs are intended to manipulate values of the Saia PCD® through the tags.

1) All these CGIs, have the restraints of the Web-Server access level, in fact if the Web-Server is in access level 2, it is only possible to read the values and not to write them, etc…
2) All this functions are based on the GET method of the HTTP protocol.

A.2.1 readVal.exe

Read value is used to read a single Saia PCD® media.
The syntax of it, is the following:

http://pc_name/station_name/cgi-bin/readVal.exe?TAG

pc_name: the PC where the Web-Connect is running.
Station_name: The name assigned to a Saia PCD®, corresponding to his
communi cation settings.
/cgi-bin/readVal.exe: the CGI
? : separation between the CGI and the parameters.
TAG : media tag, which has to be read.

The return answer will contain the value of the specified media.

Example
This is the request to read the register 1000, the value should be in decimal format:

```
http://pc_name/station_name/cgi-bin/readVal.exe?PDP,,R1000,d
```

The answer to that in the HTTP protocol will be:

```
HTTP/1.0 200 OK[0x0A]Content-Type:text/plain[0x0A]Content-
Length:4[0x0A][0x0A]3400
```

The value is: 3400

Access level of Saia PCD®

It is also possible to read in which level actually the Saia PCD® is, this in the purpose to limit some functionality when you are writing your own applet.

http://pc_name/station_name/cgi-bin/readVal.exe?PasswordLevel

This will return the actual access level of the Web-Server (between 0 to 4).

A.2.2 writeVal.exe

Write value is used to write a single Saia PCD® media.
The syntax of it, is the following:

http://pc_name/station_name/cgi-bin/writeVal.exe?TAG+xxx

pc_name: the PC where the Web-Connect is running.
Station_name: The name assigned to a Saia PCD®, corresponding to his
communi cation settings.
/cgi-bin/writeVal.exe: the CGI
?: separation between the CGI and the parameters.
TAG : media tag that has to be written.
+: separation between parameters
xxx: the value to be written, the format of the value was specified in the tag itself !

The answer to the request is the value you had sent.
HTTP/1.0 200 OK[0x0A]Content-Type: text/plain[0x0A]Content-Length: 3[0x0A][0x0A]xxx

xxx: is the value sent
[0x0A]: it is a hexadecimal value that can’t be represented as an ASCII character.

Example

This is the request to write the register 1000 with the value 2300 and the value is in decimal format.

```
http://pc_name/station_name/cgi-bin/
writeVal.exe?PDP,,R1000,d+2300
```

The answer to that in the HTTP protocol will be:

```
HTTP/1.0 200 OK[0x0A]Content-Type:text/plain[0x0A]Content-
Length:4[0x0A][0x0A]2300
```

So the value written is: 2300

A.2.3 OrderValues.exe

OrderValues is used when access to more than one Saia PCD® media is needed in the same time. In fact the functionality allows to define a list of media to be read, but this will just define the list of media and not read it. To read it, you have to use the “ReadFile.exe”, see next chapter.
The syntax is as follows:

http://pc_name/station_name/cgi-bin/OrderValues.exe?listname+dummy+ssss+TAG1+TAG2+ TAG3+TAG4+TAG5+TAG6+TAG7

pc_name: the PC where the Web-Connect is running.
Station_name: The name assigned to a Saia PCD®, corresponding to his
com munication settings.
/cgi-bin/OrderValues.exe: the CGI
?: separation between the CGI and the parameters.
Listname: list (or file) name of the media, that will be ordered (listed, referenced). This name is then used to read the media or to rewrite the list. It is possible to have in the same time more than one list in the Web-Server, this will de pend on the amount of memory allocated to the Web-Server RAM disk.
+: separation between parameters

dummy: this is a dummy parameter, not used today, but it must be in the request anyway, so the word “dummy” should be written in the request.

ssss: this has to be a number, which corresponds to the size of memory necessary to reserve in the RAM disk of the Web-Server, to save the actual sent list of media, and also to prepare the answer to this media list. Answer is the media list plus the media value. See ReadFile.exe. So the size here specified has to be large enough for the whole the list plus the answer. The number has to be specified in byte, so writing here 1000, means 1000 bytes reserved for the list specified. The default size of memory allocated for the RAM disk is 2000bytes, but it is possible to configure a bigger size.

TAGx: the media to be accessed.

The answer to this request is normally:
„HTTP/1.0 200 OK*[0x0A]*Content-Type: text/plain *[0x0A][0x0A]*done*[0x0A]*” , it just means, that the http request has been received correctly, but it doesn’t mean that the syntax of tags are correct.

[0x0A]: is it a hexadecimal value that can’t be represented as an ASCII character.

Example

We create a list of tags, this list of tags will be called room2, and will contain 3 tags which refer to the media Register 200, Flag 50, and Register 250.

```
http://pc_name/station_name /cgi-bin/OrderValues.exe?room2+
dummy+1000+PDP,,R200,d+PDP,,F50,b+PDP,,R250,d
```

the answer is:

```
HTTP/1.0 200 OK[0x0A]Content-Type:text/plain[0x0A][0x0A]Done.[0x0A]
```

A.2.4 ReadFile.exe

Read file is used to read the media values that have been predefined in the list with the “ordervalue” function. The function will return the list of the tags with their value.

The syntax is as follows :

http://pc_name/station_name/cgi-bin/ReadFile.exe?listname

pc_name: the PC where the Web-Connect is running.
Station_name: The name assigned to a Saia PCD®, corresponding to his communication settings.
/cgi-bin/ReadFile.exe:the CGI
?: separation between the CGI and the parameters.
Listname: list (or file) name of the media list that have been ordered before with „ordervalue”.

HTTP/1.0 200 OK.Content-Typecontent/unknown Content-Length: 343*[0x0A][0x0A]*TAG1=20689*[0x0A]*TAG2=61377*[0x0A]*TAG3=543*[0x0A]* TAG4=543*[0x0A]* TAG5=543*[0x0A]*

It will return the tags defined in the list and their values.

[0x0A]: it is a hexadecimal value that can’t be represented as an ASCII character.

Example

In this example we read the list created in the OrderValues chapter example.

So the list name is “room2”.

The request:

```
http://pc_name/station_name/cgi-bin/ReadFile.exe?room2
```

The answer:

```
HTTP/1.0 200 OK[0x0A]Content-Type: content/
unknown[0x0A]Content-Length: 47[0x0A][0x0A]PDP,,R200,d=3450[0x0
A]PDP,,F50,b=1[0x0A] PDP,,R250,d=1768[0x0A]
```

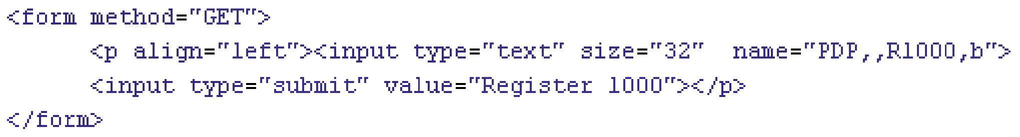
A.3 HTML coding

In this chapter, we do not explain the HTML coding, but just how to work with the Web-Server directly from the HTML code.

A.3.1 Writing values in the Saia PCD®

In the Web-Server documentation we have already described how to write a value in the Saia PCD®, the example is made with the FrontPage Express editor. Here we show how to do it directly in the HTML code.

1) first of all, create a form and use the method „GET”. In this form you will need a text input field to enter the value to write and a button to submit the form.
2) The text input field is defined by the keyword „input” and the type „text”. Very important is the name of it, which is the tag referring to the Saia PCD® media that has to be written. In this example we refer to the Register 1000 and the format is binary. Note that the tag is specified without the % at the beginning and at the end. Because we don’t want that the tag will be replaced with the value.
3) An input filed type „submit” is a button with the function to submit the form to the server.



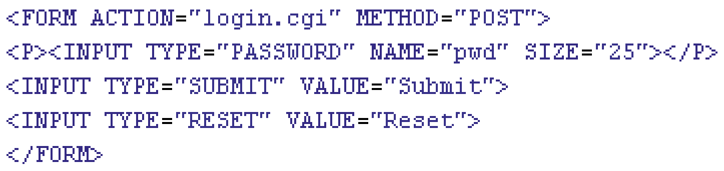
```
<form method="GET">
      <p align="left"><input type="text" size="32"  name="PDP,,R1000,b">
      <input type="submit" value="Register 1000"></p>
</form>
```

This will send one value to the Server to write (force) a media, it is of course, also possible to write more than one media in the same time, for it, just add as much as desired input filed as described in step 2.

A.3.2 Enter the Password

As you know, the access to the Saia PCD® Web-Server can be protected with passwords. To enter the password, a special page is available in the Saia PCD®, this page is named „pwd-form.htm”. But for some reasons you like to create you own html page, with a different presentation. Here is the code necessary for it.

1) A form needs to be created, but make sure to use this time the method „POST”.
2) Then an input filed, the TYPE in our case is „PASSWORD”, like this the entered data will be hidden”, but you can also use a TYPE „text” if you don’t want to hide the data entered. Important is that the name is „pwd”.
3) Then a button is needed, so input field of the type „submit” is necessary.
4) The Reset function is optional, it is just to reset the form if a mistake has been made while typing the password.



```
<FORM ACTION="login.cgi" METHOD="POST">
<P><INPUT TYPE="PASSWORD" NAME="pwd" SIZE="25"></P>
<INPUT TYPE="SUBMIT" VALUE="Submit">
<INPUT TYPE="RESET" VALUE="Reset">
</FORM>
```

A.4 Dynamic data update in the html page

As described in the documentation, you saw that it is possible to display the value of a Saia PCD® media into a html page, this by writing a tag in it. Then this tag is replaced by the media value when the Web-Server Saia PCD® is sending the file to the PC (and then into the Browser).

So the value in the Saia PCD® is changing, the only way to have an updated value is to upload the page again. This can work fine in some applications, but in some cases where you have pictures in the pages or large pages html. Loading the html page every time to have the value updated can have undesired side effect, which are: The page will disappear on the browser till the new page is there, uploading the whole page and eventually with pictures can take a long time.

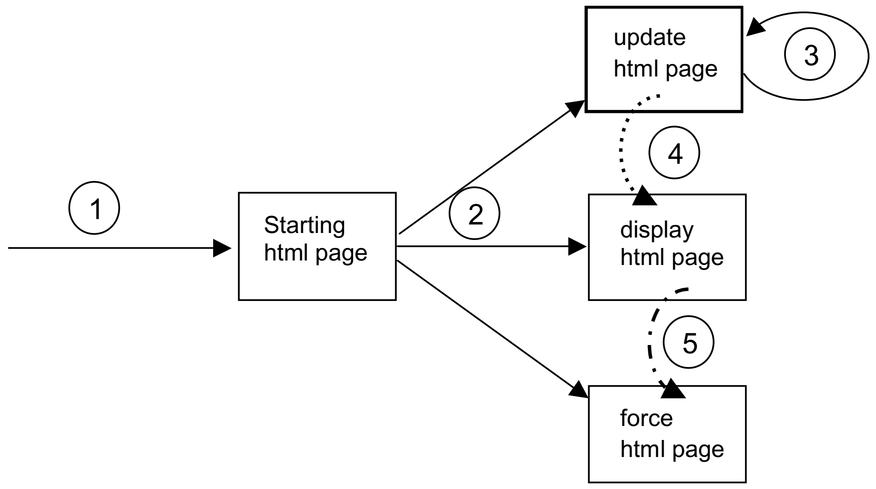
So, it is possible to avoid those problems with a little trick. It's described below how to do it.

Also the same problem can happen if you have a form in your html page to force a value inside the Saia PCD®, when you are submitting the form, it will reload the html page.

This trick to update automatically the values and to show them, is not an official feature of the Saia PCD® Web-Server, is it just presented here as an example of the html and internet possibilities. This example is working and based on the Internet Explorer Version 6. For future version of the Internet Explorer or other browsers, the example may need some adaptation. Saia Burgess Controls is not responsible for any problem that this example can cause or generate on your application.

A.4.1 The concept

The concept is to have more than one page to do all we want.
In fact, one page will be displayed to the user, but in the background (invisible) we will have another page that will update values and also write values to Saia PCD® when necessary. This is possible thanks to the java script and the communication between different html pages. One page will do the work and communicate it to the other pages. Or one page is working and transfers this work to another page.



To make this system working with all possibilities (update values and force values), we need 4 html pages. But finally only one page will be displayed.

The page **Starting** is called **(1)** by the user. This page will not do more than call **(2)** the other three pages and this by creating frames. One frame for each page, and the only visible frame will be the frame which contains the page **display** .

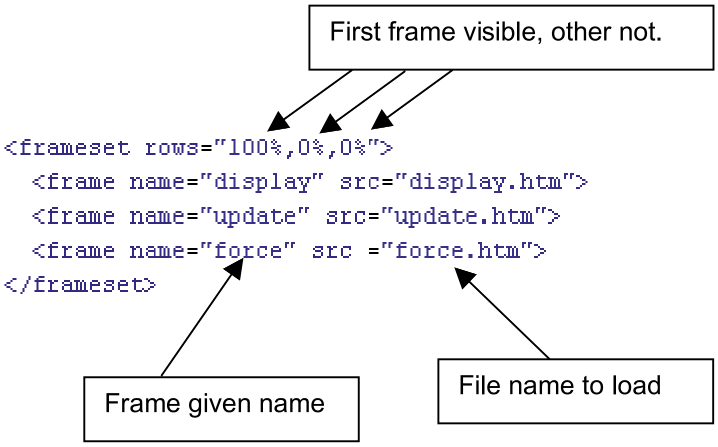
Now, the page **update** is the page that contains all the tags and which is regularly reloaded **(3)** from the Saia PCD®. This is done by an automated function in this page and a timeout. When the page is loaded, it is calling a function **(4)** in the display page to signal that new values are arrived, on the display page the function will then rewrite the value in each field that needs to be updated.

The page **force** is doing nothing till now than waiting. When the user like to force a value of the Saia PCD®, he will write a value in a field on the display page and /or click a button. This will call a function , that will take the value to be written, and transfer it **(5)**to the force page. Values are then introduced in a form like in the force page and submitted. This has the advantage, that only the force page will be reloaded after a submit of the force, the display page is not affected.

Now we saw the concept, let's see an example. This example is also delivered with the documentation. Project name is "**Dynamic_update**".

A.4.2 Starting page

As described before it is just making the frame and loading the 3 others pages. Now, for easier work with the different documents (from one to the other one), we give a name to each frame.



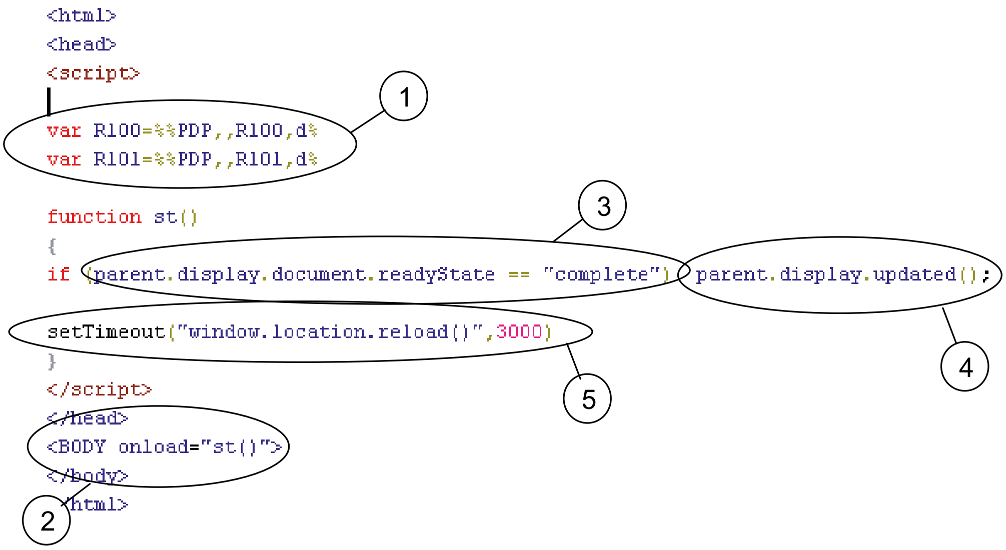
```
<frameset rows="100%,0%,0%">
  <frame name="display" src="display.htm">
  <frame name="update" src="update.htm">
  <frame name="force" src ="force.htm">
</frameset>
```

A.4.3 Update page

This page is the page collecting the Saia PCD® media and updating them automatically.

The Saia PCD® media are taken from the Saia PCD® always trough a simple tag and directly placed in **(1)** in Java script Object (variable). It will be then easy to communicate this value to different pages or functions.

The value and object described just before, are of course available only when this page has been totally loaded. So we use the onload event **(2)**. Then it means that when the page is totally loaded function st () is executed.



```
<html>
<head>
<script>

var R100=%%PDP,,R100,d%
var R101=%%PDP,,R101,d%

function st()
{
if (parent.display.document.readyState == "complete") parent.display.updated();

setTimeout("window.location.reload()",3000)
}
</script>
</head>
<BODY onload="st()">
</body>
</html>
```

This function will check if the display page exists **(3)** and if this condition is true, then it will call the function updated() **(4)** on the display page. We will see this function in the display page chapter. When the function has returned, it will start a timer **(5)**, for the reload of the page itself. When the timer has finished, the page upload is reloaded from the Saia PCD® and all is restarting as described from (**1**).

Define variable

The variable or Saia PCD® media that have to be accessed inside the Saia PCD®, has to be defined in this page on part **(1)**.

Refresh time

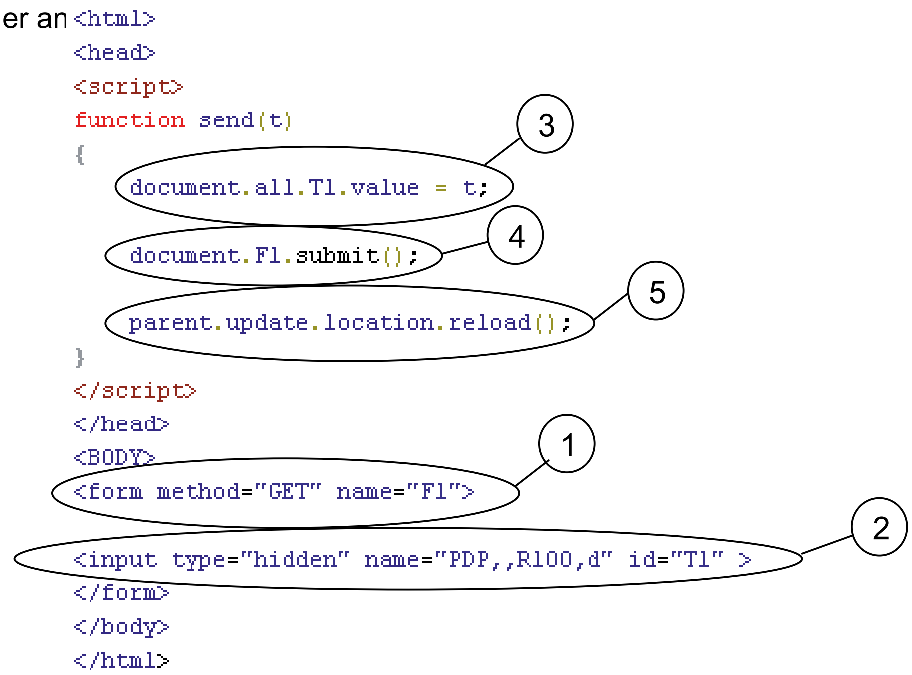
If the refresh time has to be redefined, then this one has to be done in this update page only. This is done on the setTimeout function call, in part **(5)**. In this example it is set to 3000 milliseconds = 3 seconds.

Do not put this value too small, problems can occur in the synchronization of the html pages.

A.4.4 Force page

This page is just used to send values down to the Saia PCD®. This just to avoid that the Display page is reloaded after a submit of a form.
So mainly this page contains a form **(1)**, which we gave the name „F1” to work easier with it. In this form we have an input field, the name of the input field **(2)** corresponds to the tag (media of Saia PCD®) that we want to write the value too. As this input field will not be displayed (because the force page is not visible), we can declare this input field as „hidden” type.

The display page is calling the send () function and one parameter is the value to be sent to the media. So the function send () does first affect **(3)** the receive value to the input field belonging to this page. Then, the form in this page has to be submitted **(4)** and send to the Saia PCD®.
Finally to have an update of new values displayed on the screen, we push the reload **(5)** of the update page to be made (this command is totally optional).
Then the function returns and the page is reloaded automatically from the Web-Serv-



er an
```
<html>
<head>
<script>
function send(t)
{
    document.all.T1.value = t;
    document.F1.submit();
    parent.update.location.reload();
}
</script>
</head>
<BODY>
<form method="GET" name="F1">
<input type="hidden" name="PDP,,R100,d" id="T1" >
</form>
</body>
</html>
```

A.4.5 Display page

The display page is the most important one, as it is the only one which will be displayed. In this page we can find some java script functions on the top of the page and the html body, where finally the content of the display will be made.

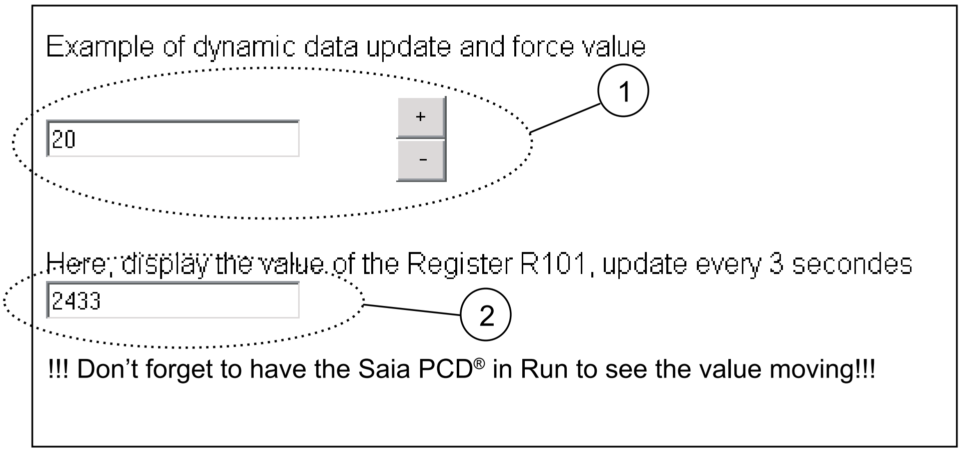
In our example the browser will show this:

(1) An input field and 2 buttons, that are used to increase or decrease the value in side the input field. But this field will be increased only when the value changed inside the Saia PCD®. So with that it is possible to be sure that the value has been increased or decreased.

(2) A simple text field that just displays a value and which is also steadily updated.

Now, here are 2 functions in javascipt:

(3) Change() function is called when a button is pressed and the value has to be changed inside the Saia PCD®. The parameter is the field that has the value and the other parameter is the value to add or subtract to it. Then the function is just making the calculation and calling the function in the force page to force the value inside the Saia PCD®.



Example of dynamic data update and force value

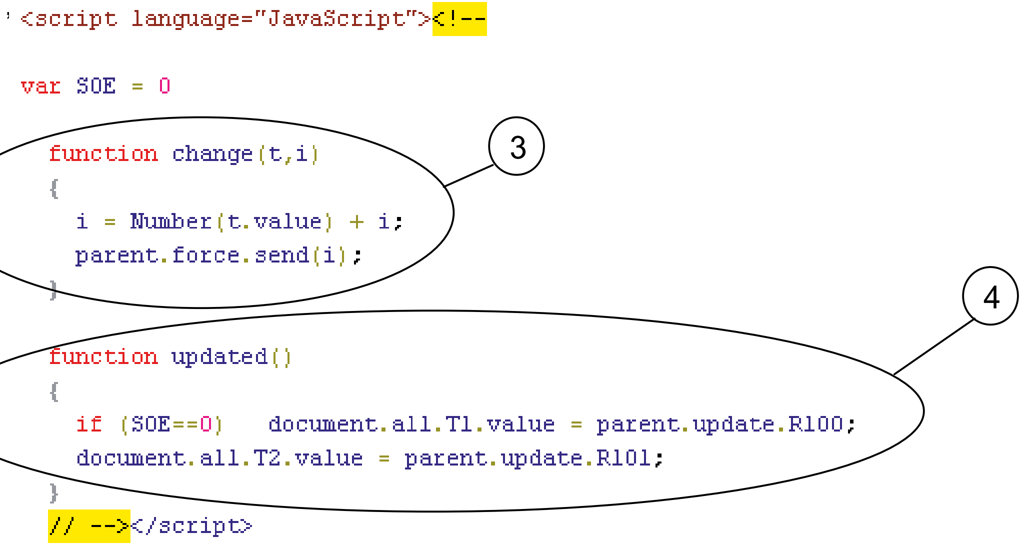
20

Here, display the value of the Register R101, update every 3 secondes

2433

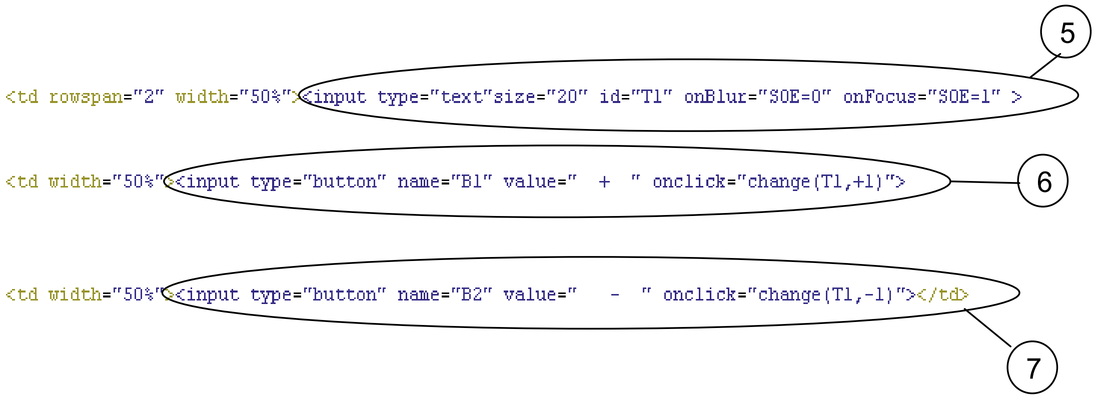
!!! Don't forget to have the Saia PCD® in Run to see the value moving!!!

(4) This function is the one called from the update page, when the new value ar rived from the Saia PCD®. Here we just make a test with the „SOE", to see if the user is actually editing the input field, then in this case we don't want to overwrite the value that is being entered. After that we take the value from the page "update" and we put it inside the input field called here „T1". And the second value in



```
<script language="JavaScript"><!--

var SOE = 0

  function change(t,i)
  {
    i = Number(t.value) + i;
    parent.force.send(i);
  }
  function updated()
  {
    if (SOE==0)   document.all.T1.value = parent.update.R100;
    document.all.T2.value = parent.update.R101;
  }
  // --></script>
```

Now in the body of the html page, we need to create the text field and the button. This is done as shown in the example below.
The input field is created **(5)** with the keyword "input" and type "text", then the id is given by id="T1". In this example we also detect when the user put the cursor in the field and when the cursor is leaving the filed, this is optional and this detection is done by OnFocus and onBlur. The buttons are 2, one to increment **(6)** and the other one to decrement **(7)**. The keyword is also input, but type is "button", respectively they have ID "B1" and "B2". Onclick is the event generated when the button is clicked. In this case we call the function described before in this document and we give the text field from which the value has to be modified and also give the value to add or subtract from it.



```
<td rowspan="2" width="50%"><input type="text"size="20" id="T1" onBlur="SOE=0" onFocus="SOE=1" >

<td width="50%"><input type="button" name="B1" value="  +  " onclick="change(T1,+1)">

<td width="50%"><input type="button" name="B2" value="   -  " onclick="change(T1,-1)"></td>
```

A

A.5 General principles of Web-Server technology

A.5.1 General notes concerning Web-Servers

The basis of Web-Server technology is the representation of web information (web = totality of different, associated HTML pages and their files, so that a menu structure emerges) using standard SW tools (Web-Browsers).

This includes the possibility of saving web information displayed directly to the PC on which the Web-Browser is located.
However, if web information is saved on a remote PC, this PC is called the Web-Server, only serving to store web information and transfer it on request to the Web-Browser.
The transfer of web information between the Web-Server PC and the Web-Browser PC can take place across any PC port (serial port, modem, LAN, Internet….) and normally occurs under the TCP/IP protocol.

Due to the limits of transmission capacity (e.g. on Internet) and the desire for fast loading times, the data structure of this web information has been very efficiently organized and needs little space in the Web-Server's memory.

The size of a normal HTML page is in the area of a few kBytes; image files are compressed with powerful algorithms, executable programs (e.g. Java-Applets) are processed interpretative in the browser (on the PC).

Since memory space is also limited in the controllers of industrial applications, Web-Server technology is ideally suited to this field too, for example to save images and files from control and monitoring interfaces directly to the industrial controller.
These small volumes of data place only a modest burden on the Saia PCD®'s CPU and therefore allow the majority of the CPU's capacity to be used for control purposes.

As a result, control and monitoring information is always stored decentrally, directly on the controller. Any user now accessing this control and monitoring information, regardless of the device used to access the data, always receives the same, up-to-date information.

This dispenses with the need for synchronization of control and monitoring information when- ever the application is updated (previously required by control and monitoring solutions that save information on the PC).

An additional advantage is the fact that web information can be edited with well known Microsoft tools, such as Word or FrontPage.
Web information is manipulated with standard browsers, designed for intuitive use and familiar to everyone.

A.5.2 TCP/IP protocol

TCP/IP is the lowest common denominator for overall data communication on Internet. From a historical point of view, it was only with this protocol that a limited network became a network of networks. Regardless of whether you call WWW pages, send e-mail, download files with FTP or work on a remote computer with Telnet: data is always addressed and transported in the same way. TCP stands for **Transmission Control Protocol;** IP stands for **Internet Protocol**.

When you send an e-mail or call a HTML file on WWW, data is broken up into little packages during transmission through the network. Each package contains information about the address to which it should be sent and the place of that package in the sequence of transmission.

The IP looks after addressing. An addressing plan exists for this purpose: the so-called IP addresses.

To ensure that packages of data actually reach their destination, and in the correct order, is the job of the TCP. The TCP uses sequential numbers for individual packages in a transmission. The transmission of data is not considered complete until all packages in a transmission have been received in full at their destination.

Every computer participating in the Internet is logged onto the network with an IP address. Computers that are connected to Internet are called **Hosts**. Therefore, when you use your PC to surf the WWW or collect new e-mails, you are logged onto Internet with an IP address. Your service provider, through whose host computer you dial up, can arrange fixed IP addresses for you. Major service providers, e.g. online services like CompuServe or AOL, also assign impersonal, dynamic IP addresses for every Internet dial-up. For a computer to participate on Internet, it must have software that supports the TCP/IP protocol. For example, under MS Windows this is the **winsock.dll** file in the Windows directory.

A.5.3 IP addressing

A typical IP address written in decimal looks like this: 149.174.211.5 – i.e. four numbers separated by stops. The stops have the job of addressing higher and lower ranking networks. Just as a telephone number in the international network has a country code, area code, subscriber number and sometimes also a direct dialling-in number, Internet also has an area code – the **network number**, and a direct dialling-in number - the **host number**.

The first part of an IP address is the network number and the second part is the host number. The location of the boundary between network number and host number is defined by a classification plan for network types. The following table explains this plan. In the ‘IP addressing’ and ‘Typical IP address’ columns, the network number (area code) is printed in bold characters. The rest of the IP address is the host number of a computer within that network.

Network type	IP addressing	Typical IP address
Class A network	**xxx**.xxx.xxx.xxx	**103**.234.123.87
Class B network	**xxx.xxx**.xxx.xxx	**151.170**.102.15
Class C network	**xxx.xxx.xxx**.xxx	**196.23.155**.113

Class A networks form the highest level of the hierarchy. Only the first number in an IP address is the network number. All other numbers are host numbers within the network. Network numbers in this kind of network can have figures between 1 and 126, i.e. there can only be 126 class A networks in the whole world. An IP address that belongs to a Class A network is therefore identifiable by having its first number between 1 and 126. The American military network is an example of such a class A network. Within a class A network the relevant network operator can allocate at will the second, third and fourth figures of individual IP addresses to participants in the network. Since all three figures can have values between 0 and 255, a class A network operator can assign up to 16.7 million IP addresses to host computers in that network.

Class B networks are the second-highest level in the hierarchy. Such networks have network numbers that extend across the first two figures of the IP address. For the first figure, class B networks can have values between 128 and 192. An IP address that belongs to a class B network is therefore identifiable by having its first number between 128 and 192. Values between 0 and 255 are allowed for the second number.

This means that around 16,000 such networks are possible. Since the third and fourth numbers in such networks may also have values between 0 and 255, any class B network can have up to 65,000 host computers connected. Class B networks are mainly assigned to large companies, universities and online services.

Class C networks occupy the lowest level of the hierarchy. The first number of a class C network's IP address lies between 192 and 223. The second and third numbers also form part of the network number. Over two million such networks can be addressed in this way. These addresses are mainly assigned to small and medium-sized companies with direct Internet connections, but also to the smaller Internet providers. Since only one number is left with a value between 0 and 255, the maximum number of host computers that can be connected in a C network is 255.

Some people doubt whether this addressing plan can meet future needs. Ideas already exist for restructuring network and host computer addressing.

A.5.4 Client-Server-Technologie

Any host computer intending to offer to other computers such Internet services as World Wide Web, Gopher, e-mail, FTP, etc. must have the appropriate **server** software running on it. A host computer can only offer an Internet service if the appropriate server software is active on the computer and if the computer is online.

Servers are programs that permanently wait to receive enquiries regarding their service. A WWW server, therefore, waits to receive enquiries that will call off WWW pages from the server computer.

In contrast, **clients** are software programs that typically request data from servers. Their WWW browser, for example, is a client. If you click on a reference that leads to an HTTP address, the browser (i.e. the WWW client) starts a request to the relevant server on the remote host computer. The server evaluates the request and transmits the data requested. Protocols exist to regulate communication between clients and servers. Client-server communication in the WWW is basically regulated by the HTTP protocol. This type of protocol runs above the TCP/IP protocol.

Normally, a client requests data and a server transmits data. However, exceptions also exist whereby a client not only can request data, but also send data to a server (e.g. when you use FTP to load a file on the server computer; when you send an e-mail or fill in and return a form in the WWW). These cases are also referred to as **Client-Push** (the client pushes data onto the server).

Another exception is when the server becomes active first the sends the client something without being asked. This is called **Server-Push** (the server pushes data onto the client). New technologies want to elevate this exception to a rule. They are called **push technologies**. These technologies are intended to enable a client to receive data regularly, without personally asking for it. They make it possible to realize broadcasting services like current news, etc. Netscape and Microsoft Internet Explorer (both from version 4.0) have the relevant interfaces to make use of such services.

A.5.5 DNS - Domain Name Service

Generally, computers work better with numbers and people work better with names. For this reason a system has been invented that translates the numeric IP addresses into address names that are clear for the final user.

The system created has a hierarchical structure like the IP addresses. An address name in this system belongs to a **top level domain** and within that to a **sub-level domain**. Each sub-level domain can in turn contain subordinate domains, but does not have to. The individual parts of these address names are separated from each other by stops, as with IP addresses. An example of such an address name is: **teamone.de**.

Top level domains occupy the final position of a domain name. As abbreviations, they more or less speak for themselves. The abbreviations for top level domains are either country codes or type codes. Examples are:

de =	Germany (Deutschland)
au =	Austria
ch =	Switzerland
it =	Italy
my =	Malaysia
com =	commercial proprietor
org =	organization
net =	general network
edu =	American educational establishments
gov =	American government offices
mil =	American military installations

Each of these top level domains represents an administrative area for which an “administrative authority“ also exists with responsibility for assigning the names of sub-level domains within its administrative area. For example, if you wanted to apply for a domain name like **MyCompany.de**, the application would have to be made to the DE-NIC (German Network Information Center). Commercial providers will do this for you if you use them for any corresponding service. However, you will only receive your chosen name if the address name has not been assigned elsewhere. Unscrupulous types therefore had the bright idea of reserving for themselves the names of large companies that had not yet applied for their own domain, to sell them on later at a high price, when even these large companies had recognized a sign of the times. Since then, measures have been taken to prevent such machinations. However, legal action is frequently resorted to in the cause of attractive domain names. If two companies happen to have identical names (but no other connection) and want to reserve the same domain name, only one application can be accepted. To minimize this type of conflict, new top-level suffixes have since been introduced.

The owners of two-part domain names can in turn assign sub-level domains. This has resulted, for example, in a domain name **seite.net**. The operators of this domain have in turn assigned sub-domains, producing domain addresses like **java.seite.net** or **javascript.seite.net**.

A.5.6 Routing and gateways

Internet, as the network of networks, cannot at present allow the direct transmission of data from one IP address to another within their own sub-network. In all other cases, when data is to be sent to another network number, computers that regulate traffic between networks come into the plan. These computers are called **gateways**. They route data from host computers in their own sub-network to gateways in other sub-networks and route incoming data from the gateways of other sub-networks to the host computer addressed in their own sub-network. Without gateways there would be no Internet.

Routing is the specific term for this activity and possible routes from a gateway computer’s own network to other networks are defined on the gateway computer in **routing tables**.

A gateway also has the job of finding an alternative route if the normal route does not work, perhaps due to a fault on the line or a data jam. Gateways constantly send themselves test packages, to check the connection is working and to find routes for data transfer where the traffic is light.

This means, therefore, that when data transfer takes place on Internet it is not at all clear from the outset which route data will take. Even the separate packages that make up a single transmission can take completely different routes. For example, if from Germany you call a WWW page that is located on a computer in the USA it may be that half the page comes over the Atlantic and the other half over the Pacific before your WWW browser can display it. Neither you nor your browser would realize it.

A.6 General principles of HTML

A.6.1 Mark up elements

HTML stands for **H**yper **T**ext **M**arkup **L**anguage. The language is defined with the help of SGML (Standard Generalized Markup Language). SGML was laid down under ISO standard 8879.

As a markup language, HTML has the job of describing the logical components of a document. It therefore contains instructions for marking typical elements of a document, such as headers, text spaces, lists, tables or graphical references.

The HTML model assumes a hierarchical division. HTML describes documents. Documents have global properties, such a title or a background colour. The actual content consists of elements, e.g. a first-order heading. Some of these elements in turn have sub-elements. A text space, for example, might contain a text position that is marked as bold, a list comprises separate list items and a table is divided into individual cells.

A fixed extent can be defined for most of these elements. In this way, a header extends from the first to the last character, a list from the first to the last list item, or a table from the first to the last cell. Marks indicate the start and end of elements. The following model applies for marking up a heading:

[Heading] *Text of heading* **[End of heading]**

For an element that in turn has sub-elements, a list perhaps, the same model applies:

[List]
[List item] *Text of list item* **[End of list item]**
[List item] *Text of list item* **[End of list item]**
[End of list]

WWW browsers for HTML files work out the markup instructions and then display the elements on screen in a form that is easily recognizable to the eye. However, screen representation is not the only conceivable form of output. For example, HTML can just as readily output computer-generated voices onto audio systems.

A.6.2 Cross-linking with hyperlinks

One of HTML’s most important features is the ability to define hyperlinks. Hyperlinks can lead to other places in the same project, but also to any other addresses on the world-wide web and even Internet addresses that are not part of the WWW

With this simple, basic feature, HTML opens up completely new worlds. Moving between computers that are physically far apart is reduced to a mouse-click with modern, graphical WWW browsers. In your own HTML files you can include hyperlinks that allow context-specific links between your content and that of other bidders. This fundamental idea is the basis of the entire world-wide web, to which it owes its name.

A.6.3 Software-independent plain text

HTML has a plain-text format. You can process HTML files with any text editor that can store data as straight text files. There is no specific software required for the creation of HTML files. Powerful programs that specialize in the editing of HTML may have existed for a long time, but this does not alter its crucial characteristic: HTML is not tied to any specific, commercial software product. This might even be HTML's single most important feature, any you should never lose sight of it when anyone tries to tell you that web publishing is only possible with specific software products.

The plain-text instructions of HTML have been designed for machines **and** humans. Anyone who is not prejudiced against visible instructions on the screen will find in HTML an astonishingly simple shell language. It is based on English words, but the number of instructions is so limited that even those without extensive knowledge of the English language can assimilate HTML.

Since HTML has a plain-text format, it is excellent for generating with the help of programs. CGI programs, for example, make use of this capability. Whenever you use a search service on the WWW and, after your search request, are presented with the results, what you see on the screen is HTML code that has been generated by a program.

A.6.4 Universal in application

HTML was actually invented as a markup language for creating WWW pages. However, HTML files do not just work on the WWW. It is no problem to open an HTML file locally on any computer with a WWW browser. This makes HTML files ideally suited for local documentation, CD-ROM interfaces, readme files, etc. With HTML and the languages that complement it directly (CSS and JavaScript, which also function locally) you can create sophisticated projects not intended for use on the WWW. Regardless of whether you want to get your diary in shape for the next millennium, add HTML-based online help to the next version of your software, or produce an informative CD - HTML is by far the most widely used file format in the world. Your HTML files will run on any computer with a WWW browser installed - and any computer that does not have a WWW browser can these days be called obsolete in every respect.

A.6.5 Formatting for HTML elements

HTML is a “logical“ language. With HTML, you define the basic structure of your WWW pages, the elements, structures, hyperlinks and referenced elements (such as graphics, multimedia, etc.). However, HTML was not really designed to specify exactly how an element should look. Therefore, you can define a heading in HTML and specify that it should be a first order heading. However, you cannot specify inside HTML how large the heading should be displayed, nor its font, etc. This is handled by the WWW browser during display. To do this it uses a mixture of basic settings, that the user can make, and fixed, programmed representations of individual HTML elements.

In the early days of HTML's great success, the language was “misused“ for all kinds of physical formatting. This meant that an HTML command for displaying flashing text would suddenly pop up, and later a command became popular that could be used to define the font, type size (in 7 relative stages) and type colour. However, in the end all these implementations in HTML are inconsequential.

At this point Cascading Style Sheets (CSS) come in. This is a language that directly complements and was specifically developed for HTML. It slots seamless into HTML and allows HTML elements to be formatted as desired. With the help of CSS style-sheets, for example, you can specify that all first order headings should be displayed as 24 points in height, written in red Helvetica, with a 16-point following space and a green double-line on the frame above. However, you can also just as easily specify for any text that it alone should be 3 centimetres high and should have the background colour yellow. In addition, the CSS language also contains instructions for the precise positioning of elements on the screen and for other output media, such as print or audio systems. CSS style-sheets therefore provide HTML with a powerful forward thrust.

The CSS language, just like HTML, has an official standard. Like HTML it is maintained and developed by the W3 consortium. In the meantime, more recent browser versions from Netscape and Microsoft also interpret the style-sheets.

A.6.6 A programming language for WWW pages

Among other things, forms can also be defined in HTML. Such forms can contain entry fields, selection lists, buttons, etc. The user can fill in a form and send it back via the WWW This invention is a blessing for many purposes. However, HTML does not allow you as the supplier of a form, for example, to check the user's entries for completion or plausibility before the form is sent back.

Or another case: although you can integrate a VRML file into HTML, when that HTML file is on the WWW you cannot know whether the user has a browser that can display VRML. It would be useful if the instruction to integrate the VRML file could be made dependent on the user's browser being able to display VRML.

For these and countless other useful but less used purposes, Netscape invented a programming language called JavaScript. You can include JavaScript instructions directly inside HTML files, or integrate them as a separate file.

JavaScript is increasingly gaining in importance for modern WWW pages. Recent developments in JavaScript are above all responsible for what has been called “Dynamic HTML”. Development has started and nothing can stop it now: JavaScript allows retrospective access to all elements of an HTML file during display. Fascinating new effects are therefore made possible, which normal HTML cannot do. A text can be replaced with another at the click of a mouse, a user event or time control can make elements disappear automatically, fade in, or change position on the screen.

If you want to create sophisticated WWW pages, you will find it hard to do without at least simple JavaScripts any more. For this reason SELFHTML contains extensive documentation on JavaScript. In any case, you should devote some attention to JavaScript as well as HTML

A.6.7 JavaScript and HTML

JavaScript is not a direct component of HTML, but a separate programming language. This language has, however, been specially created for the purpose of giving HTML authors a tool for optimizing WWW pages.

JavaScript programs can either be written directly in the HTML file, or in separate files. Unlike Java programs, they are not compiled, but interpreted as source text during run-time, i.e. like batch files or shell scripts. Modern WWW browsers such as Netscape or Microsoft Internet Explorer have the appropriate interpreter software for this.

A programming language such as JavaScript contains many confusing elements for the beginner: special characters, variables, if-then statements, loops, functions, methods, parameters, objects, properties and much more. To get properly to grips with these elements, you must learn to imagine what is happening in the computer when program code is executed. This is a prolonged learning process that users only master with much practice. However, JavaScript is excellently suited to this, because it is a comparatively simple language without many of the work areas found in a “major“ programming language, e.g. things like main memory management or file operations. In addition, JavaScript comes down to a specific environment, i.e. a WWW page that either has been displayed or will be.

A.6.8 JavaScript, JScript, ECMA-262, language versions

JavaScript is a script language introduced and licensed by Netscape. New versions of the Netscape browser therefore always offer rather more JavaScript than the competition, who can only wait to see what innovations Netscape has implemented.

Netscape 2.0 interprets JavaScript language standard 1.0, which was launched at the time. MS Internet Explorer mostly understands this language standard from version 3.0.
From version 3.0, Netscape interprets JavaScript standard 1.1. MS Internet Explorer mostly interprets this JavaScript language range from product version 4.0.
Netscape interprets JavaScript standard 1.2 from version 4.0. Some of the instructions in this language version are also interpreted by MS Internet Explorer 4.0.
Netscape and MS Internet Explorer also differ in the interpretation of individual object features. For example with Netscape, Cookies (a well known JavaScript feature) also function locally, whereas MS Internet Explorer, at least in version 3.x, requires a HTTP connection to support Cookies. Even with internal processes, such as loops that have to run through several times, there are differences between individual browsers and browser versions.

MS Internet Explorer 4.x does interpret JavaScript, but it also has Microsoft’s own language variant **JScript**, which provides special instructions for the operating system extensions of MS Internet Explorer.

Both Netscape and Microsoft stress that their language interpreters comply with the standard for Internet script languages: **ECMA-262**. The ECMA committee, which comprises various software manufacturers and includes Microsoft and Netscape, seeks to define a generally valid standard for the field of script languages, as is the case with HTML or CSS style sheets.

A.6.9 HTML editors

HTML pages can be developed with any preferred HTML editor. One can divide HTML editors fundamentally into either text-based editors or WYSIWYG editors (WYSIWYG = **W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et). With text-based editors you work directly with the HTML instructions. As a rule such editors have a toolbar with buttons and various menu commands to insert HTML tags in the text. Tags are displayed visibly in the text. WYSIWYG editors also offer a toolbar with buttons and menu commands to support the setting of HTML tags in the dialogue. However, the difference is that tags are not displayed (or only if requested). Instead, the text displayed on the screen is already as it will appear later, on the Web-Browsers.

At most of the following addresses you cannot download the relevant HTML editors. These relate mostly to shareware products, i.e. you are allowed to test the program, but must buy and register it before you can definitely use it.

This selection makes no claims to be comprehensive.

1-4-All	http://www.mmsoftware.com/14All/	HTML editor
4W Publisher	http://www.4w.com/4wpublisher/	Database tool that can automatically create reports in HTML form
AOL Press	http://www.aolpress.com/press/index.html	Wysiwyg HTML editor
Arachnophilia	http://www.arachnoid.com/arachnophilia/	HTML editor
Coffee Cup	http://www.coffeecup.com/editor/	HTML editor with ready-made JavaScripts, graphics, etc.
Claris Homepage	http://www.clarishomepage.com/	Wysiwyg editor
Frontpage	http://www.microsoft.com/frontpage/	Wysiwyg HTML editor by Microsoft
Horizon Web Text	http://www.gcnet.com/bw/horizon/	32-bit HTML editor
HTMLed (Pro)	http://www.ist.ca/	HTML editor
HTML Assistant	http://www.exit0.com/ez1/products/pro2000.html	HTML editor
SchemaText	http://www.schema.de/sitehtml/site-d/schemat5.htm	Professional authoring system for major web project
NetObjects	http://www.netobjects.com/	Powerful authoring tool
SuperHTML	http://www.superhtml.de/	HTML editor
Tarantula	http://www.indian-sites.com/nostrumindia/	Wysiwyg editor
WebMedia Publisher	http://www.wbmedia.com/publisher/	WYSIWYG HTML publisher

B Appendix

B.1 Icons

i	In manuals, this symbol refers the reader to further information in this manual or other manuals or technical information documents. As a rule there is no direct link to such documents.
⚡	This symbol warns the reader of the risk to components from electrostatic discharges caused by touch. **Recommendation:** Before coming into contact with electrical components, you should at least touch the Minus of the system (cabinet of PGU connector). It is better to use a grounding wrist strap with its cable permanently attached to the Minus of the system.
!	This sign accompanies instructions that must always be followed.
Classic	Explanations beside this sign are valid only for the Saia PCD® Classic series.
xx7	Explanations beside this sign are valid only for the Saia PCD® xx7 series.

B.2 Contact

Saia-Burgess Controls AG
Bahnhofstrasse 18
3280 Murten
Switzerland

Phone	+41 26 672 72 72
Fax...	+41 26 672 74 99

Email support:	support@saia-pcd.com
Supportsite:	www.sbc-support.com
SBC site:	www.saia-pcd.com
International Represetatives & SBC Sales Companies:	www.saia-pcd.com/contact

Postal address for returns from customers of the Swiss Sales office

Saia-Burgess Controls AG
Service Après-Vente
Bahnhofstrasse 18
3280 Murten
Switzerland