Embedded Web Server for FP Gateways Manual

Version: 2.2.1

www.inovolabs.com

Publication date: 23/02/2021

Table of contents

1 WEB SERVER 3

1.1 DISPLAYING PLC AND GATEWAY VARIABLES 3

1.2 SPECIAL TECHNIQUES FOR CREATING DYNAMIC WEBSITES 4
1.2.1 CGIs 4
1.2.2 HTML commands 5
1.2.3 Alias name for variable references 7

1.3 LOGALIASES TO FORMAT LOG DATA 7

1.4 UPLOADING A WEBSITE 8

1.5 TFTP FILE TRANSFER 8

1.6 WEB SERVER CONFIGURATION 10
1.6.1 Multiple websites, dividing websites 10
1.6.2 Web server directory access restrictions 11
1.6.3 Web server connection timeout 11
1.6.4 Web server TCP/IP port 12
1.6.5 Restricting the web server IP address range 13
1.6.6 TiXML TCP/IP port 14

1.7 WEB SERVER CONNECTIVITY SETTINGS 14
1.7.1 Dial-in access 14
1.7.2 Call back initiation service CBIS (only for 5th generation devices) 15
1.7.2.1 Changing the CBIS sender address 16

2 APPENDIX: PROJECT STRUCTURE AND COHERENCE 17

1 Web server

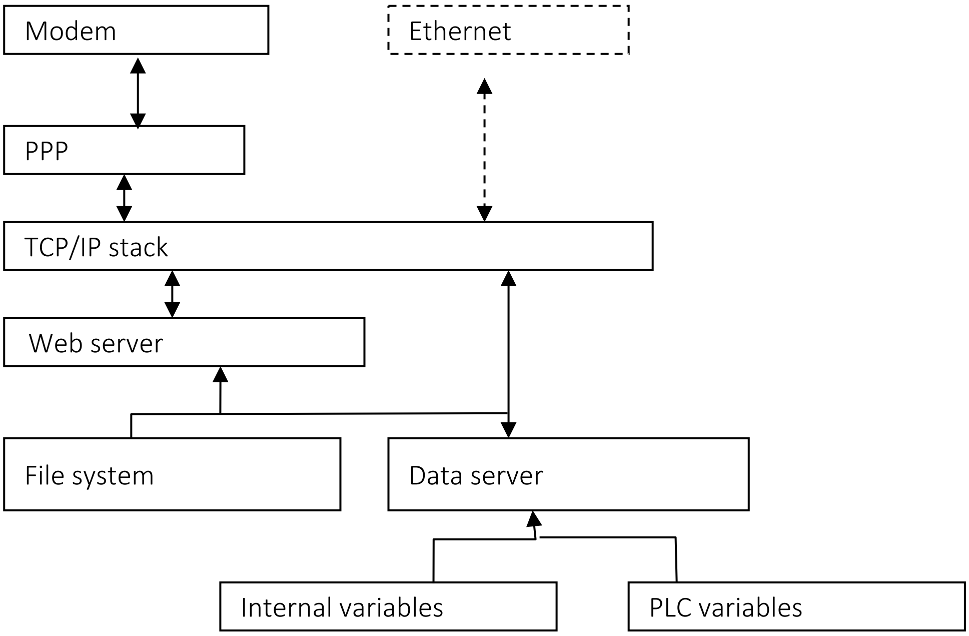
The FP IoT gateways are available with an integrated web server that enables PPP dial-up or an Ethernet connection to be used to display PLC data in a web browser.

Web server access

Default IP address:	192.168.0.1.
Home page:	index.html.

The FP IoT gateway can open up to eight http or TiXML connections simultaneously via TCP/IP.

The web server can also be addressed using a dial-out from special ISPs.



The web server displays HTML pages that are stored in the file system. The pages can contain references to internal or PLC variables that are decoded by the data server. This is achieved using special HTML codes that insert the variable’s actual value when opening the website. Direct access to the data server is possible via a CGI script in order to therefore create websites that are updated dynamically. The web server supports an HTTP file cache.

On http://www.inovolabs.com, you can obtain HTML examples and a library of website elements.

Additional information regarding file access via HTTP is provided in the “SCADA / GLT Connection for FP Gateways Manual”.

1.1 Displaying PLC and gateway variables

In order to display the values for variables using a website, a reference to their value must be inserted into the HTML code. The reference to be used in the HTML is identical to the TiXML reference: It starts with the reference code (® or in XML: &#xae;), followed by the variable’s path (e.g.: /Process/MB/IO/I/P0 to display the input for a Hutline modem).

Part of a website that displays inputs:

```
<TABLE ID="Table1" BORDER=1 CELLSPACING=3 CELLPADDING=1 WIDTH=200>
  <TR>
    <TD WIDTH=104><P>Port P0:</TD>
    <TD WIDTH=77>&#xae;/Process/MB/IO/I/P0;</TD>
  </TR>
  <TR>
    <TD WIDTH=104><P>Port P1:</TD>
    <TD WIDTH=77>&#174;/Process/MB/IO/I/P1;</TD>
  </TR>
</TABLE>
```

1.2 Special techniques for creating dynamic websites

1.2.1 CGIs

The data server can be addressed directly via a CGI script and supports log file access, as well as access to individual or groups of variables.

`cgi-bin/readLog.exe?`	to read the log file
`cgi-bin/readVal.exe?`	to read the variable
`cgi-bin/writeVal.exe?`	to write the variable
`cgi-bin/RefreshValues.exe?`	to update the variable
`cgi-bin/tixml.exe?`	to process TiXML commands via http POST

The CGIs can be used in HTML forms via “GET” (exception: tixml-exe).
The syntax for some of these commands is identical to the commands in the “Ininet SpiderControl" software. It is therefore possible to integrate “SpiderControl” TEQ files into the websites. These TEQ files contain a display that updates the values periodically via the data server. For more information, please see the SpiderControl manual. Use the aforementioned references as PPOs.

Examples

(further examples are available on http://www.inovolabs.com)

Example 1

This code generates a button to call the last 10 values in a log file. The log file is displayed in the same window.
The “Range” is specified after the +. Areas known by the TiXML command of the same name are supported by the “#“ characters must be replaced with “%23” and “-” replaced with “%2D”.

```
<html>
 <body>
    <input type=button
    onclick="location='cgi-bin/readLog.exe?Event+%2310%2D;" value="Event-Log
    lesen">
 </body>
</html>
```

Example 2

This code generates a button to set output Q2 (alias, see next section). The result is displayed in the same window:

```
<html>
 <body>
   <input type=button value="set Q/P2"
   onclick="location='cgi-bin/writeVal.exe?QP2+1';" >
</body>
</html>
```

Example 3

This code generates a button to switch QP2. The result is displayed in a second window using JavaScript:

```
<html>
<script language="JavaScript">
function window()
{
NewWindow1=window.open("","Result","width=20,height=20,resizable=no,scrollb
ars=no,top=50,left=50");
NewWindow1.focus();
}
</script>
<body>
 <FORM action="../cgi-bin/writeVal.exe" method="get" target="Result">
    Q/P2 Value:
    <INPUT type="text" name="QP2" value="">
    <INPUT type="submit" value="set" onclick="window()">
 </FORM>
</body>
</html>
```

1.2.2 HTML commands

The FP IoT gateway and its data server can be addressed using special HTML commands:

HTML commands		
Syntax	`&#xae;(`*`commandname=parameters`*`);`	
Description	HTML command to control the FP IoT gateway’s web server	
Parameter	**Disconnect**	The FP IoT gateway disconnects the connection.
	Format	Shows a variable and formats the value, e.g. &#xae;(Format=/Process/MB/IO/I/P0+?on’off);
	The format instruction after the “+“ is identical to the “External” or “Record” format instruction (see TiXML Reference Manual). Note that the unit is specified using an inverted comma “ ‘ ” instead of a semicolon “;“. In boolean formats, the separating comma “,“ is also replaced with an inverted comma “’”. The format is only possible for direct references, i.e. not for alias variables.	
	Replace	Shows a variable and replaces part of the content. e.g.: &#xae;(Replace=/Process/PV/Text:String_to_replace:new_string); The value for the referenced variable is scanned for the “String_to_replace“ string that is replaced with “new_string“.

Include Inserts the “Process” branch or a subarea into the HTML document.

ReadLog Displays the contents of a log file as an HTML table.

Examples

Example 1

When this page is loaded, the FP IoT gateway disconnects the connection:

```
<HTML>
  <BODY>
       Disconnecting... &#xae;(Disconnect);
  </BODY>
</HTML>
```

Example 2

The value “12345” for the Word02 variable and the value 0/1 for the Bit02 variable are displayed formatted:

```
<HTML>
  <BODY>
   &#xae;(Format=/Process/Bus1/Device_0/Word02+F,2'm);<br>
    &#xae;(Format=/Process/Bus1/Device_0/Bit02+?on’off);
  </BODY>
</HTML>
```

Value displayed: 123,45m

Example 3

The “Company Name” string for the “Text” variable is edited:

```
<HTML>
  <BODY>
     &xae;(Replace=/Process/PV/Text:Name:Tixi);
  </BODY>
</HTML>
```

String displayed: “Company Tixi”

Example 4

The Hutline I/Os from the “Process” branch are displayed on the website:

```
<HTML>
  <BODY>
        &#xae;(Include=/Process/MB/IO/);
  </BODY>
</HTML>
```

Example 5

This example displays the contents from the last 30 minutes of the log file directly as an HTML table:

```
<html>
 <body>
    <div align=”center”>
        <p><u>Logdata of last 30 minutes:</u></p>
        <table cellpadding=”5” cellspacing=”1” border=”1”
        align=”center”>
            &#xae;(ReadLog=Datalog+last 30 minutes);
        <table>
    </div>
 </body>
</html>
```

1.2.3 Alias name for variable references

The FP IoT gateway provides a database with alias names for I/O and PLC variable references in order to make programming CGI, Applet and HTML code easier. This enable static names to be used within the source code and these to be linked to the variables in the TiXML project.

In order to obtain the value for an input without the alias database, the following instruction is required:
HTML reference: `&#xae;/Process/MB/IO/I/P0;`
CGI call: `cgi-bin/readVal.exe?/Process/MB/IO/I/P0`
The reference contains the MB module address. In order to use the reference with a different module, e.g. address C42, the source code must be changed.

The alias database can now be used to guide a static name such as “`ExtIP0`” to a reference “`/Process/MB/IO/I/P0`”. The static name is used within the reference:
HTML: `&#xae;ExtIP0`
CGI call: `cgi-bin/readVal.exe?ExtIP0`
and the system replaces it with the reference value. This enables the same website to be used with different hardware simply by changing the alias database.

A further benefit of the alias database is the option to define access rights for variables. Write access to the variables can therefore be prevented.

In addition, the variables’ output can be formatted. For example, if different formatting is used in the external definition to that required in the web server, this can be defined separately using the alias database. The format instructions are identical to those for the log file records (see the TiXML Reference Manual) or the PLC variables.
Database path: /ISP/WebServer

```
<WebServer>
   <Aliases>
     <ExtIP0 _="/Process/MB/IO/I/P0" acc="R" />
     <ExtOP0 _="/Process/MB/IO/Q/P0" acc="RW" />
     <ExtOP1 _="/Process/MB/IO/Q/P1" acc="RW" format="?on,off"/>
     <PLCI1 _="/Process/Bus1/D0/I1" acc="R" format="F;"/>
     <PLCO1 _="/Process/Bus1/D0/O1" acc="RW" />
   </Aliases>
</WebServer>
```

1.3 LogAliases to format log data

The FP IoT gateway outputs log files in XML format.
The “LogAliases” group can be used to format the output, e.g. to generate CSV files.

The “LogAliases” group is part of the ISP/WebServer group:

Database path: /ISP/Webserver

```
<WebServer>
   <LogAliases>
     <Aliasname _="Logfile" FORMAT saveAs=”File name”/>
   </LogAliases>
…
</WebServer>
```

The FORMAT instructions are identical to those for the “IncludeLogTXT” command, described in the data logging section of the TiXML Reference Manual.

The file name entered in “saveAs” (as of FW 2.2) is specified when saving the file (as of Microsoft Internet Explorer 6.0.2900)

Example

Database path: /ISP/WebServer

```
<WebServer>
   <LogAliases>
      <DatalogCSV _="Datalog" type="CSV" saveAs=“Datalog.csv“/>
   </LogAliases>
…
</WebServer>
```

1.4 Uploading a website

The complete directory structure for the website must be compiled in a BASE64 webSrc.bin file. The webSrc.bin must be packed in a TiXML frame:

```
[<SetBinary _="HTTP" mode="Base64" name="WebSrc.bin">
<D _="NgAAAPYHAADYAycUBAElNYXN0ZXIuamFyACYcAABpbmRleC5odG1sAKcIAABJT19Q"/>
...
<D _="CiAgICAgICAgPUj4NCiAgICA8L1RBQkxFPg0KPC9CT0RZPg0KPC9IVE1MPg0KIA=="/>
</SetBinary>]
```

Tixi.Com provides tools for conversion to ASCII files.

1.5 TFTP file transfer

A TFTP program can be used to transfer files such as website to or from the FP IoT gateway. The FP IoT gateway enables this communication via an active TCP/IP connection (Ethernet or internet).
Free TFTP programs are available on the internet.

The ISP database is used to configure the binary data and access to the files.

Database path: /ISP/TFTP

TFTP file transfer	

Syntax	`<TFTP>`
	`  <Port _="`*`Number`*`"/>`
	`  <Files>`
	`    <`*`Description`*` _="`*`filename`*`" acc="`*`access`*`" size="`*`size`*`"/>`
	`  </Files>`
	`</TFTP>`

Description	File definition for TFTP file transfer.
Parameter	*Number:* TCP/IP port for TFTP communication. Default: 69 *Description:* Description of the files available. *filename:* Name of the registered file *access:* Access rights for this file: R – read-only W – write only RW – read and write access *Size:* Maximum possible size in bytes (during the upload)

Example

Three files are created: The Tixi FP IoT gateway
- Website as binary file
- A copy of the TiXML project as a Zip file
- Another binary file with graphics for the website:

```
<TFTP>
    <Port _="69"/>
    <Files>
        <Website _="websrc.bin" acc="RW" size="40960"/>
        <Project _="project.zip" acc="R" size="40960"/>
        <webpictures _="pictures.bin" acc="RW"
size="10240"/>
    </Files>
</TFTP>
```

1.6 Web server configuration

The ISP database contains further web server settings. This is the default configuration:

Database path: /ISP/Webserver

```
<WebServer>
    <ActiveSite _="Site1"/>
    <Site1>
        <Archive1 _="WebSrc.bin"/>
    </Site1>
    <Site2>
        <Archive1 _=""/>
        <Archive2 _=""/>
        <Archive3 _=""/>
    </Site2>
</WebServer>
```

1.6.1 Multiple websites, dividing websites

Several websites can be stored in the FP IoT gateway. The active site is defined using “ActiveSite”. The FP IoT gateway can divide the web contents over up to three archives. This enables data that is changed often to be separated from data that is changed rarely in order to save transmission time.

Web server - Sites	

Syntax	`<WebServer>`
	`    <ActiveSite _="`*`SiteName`*`"/>`
	`    <SiteName1>`
	`        <Archive1 _="FileName"/>`
	`    </SiteName1>`
	`    <SiteName2>`
	`        <Archive1 _="FileName"/>`
	`        <Archive2 _="FileName"/>`
	`        <Archive3 _="FileName"/>`
	`    </SiteName2>`
	`</WebServer>`

Description	Site Web server configuration.
Parameter	*ActiveSiteName:* Name of the active site. *SiteNameX*: Names of the sites available. *FileName:* Names of the binary files with web content.

Example	Two sites are configured and Site2 is active. The website is divided into three parts. One containing the HTML code, one with images and one with Java Applets:
	
	`<WebServer>`
	`    <ActiveSite _="Site2"/>`
	`    <Site1>`
	`        <Archive1 _="WebSrc.bin"/>`
	`    </Site1>`
	`    <Site2>`
	`        <Archive1 _="html.bin"/>`
	`        <Archive2 _="pictures.bin"/>`
	`        <Archive3 _="applets.bin"/>`
	`    </Site2>`
	`</WebServer>`

1.6.2 Web server directory access restrictions

Access to individual directories or files can be limited (realms) for each FP IoT gateway web server “site”. The required authentication is valid for all sites within the HTML path and is transferred from the AccessRights database. (see the TiXML Reference Manual).

Restricted directory access	

Syntax	`<Site1>`
	`  <Archive1 _="WebSrc.bin"/>`
	`  <Restricted>`
	`    <Restriction Path="`*`path`*`" Realm="`*`name`*`"`
	`AccLevel="`*`level`*`"/>`
	`  </Restricted>`
	`</Site1>`

Description	Site Web server configuration.
Parameter	*Restriction:* Name of the restriction. *path:* Relative URL for the restricted directory *name:* Freely-selectable name for the access area *level:* Access level for this restriction (see the TiXML Reference Manual)

Example	One site is created. Access to the SYSTEM/ path is only permitted to users with access
	level 2. All other directories (*) are open to all.
	
	Database path: /ISP/WebServer
	
	`<WebServer>`
	`    <ActiveSite _="Site1"/>`
	`        <Site1>`
	`            <Archive1 _="WebSrc.bin"/>`
	`            <Restricted>`
	`            <System Path="SYSTEM/*" Realm="TAM_SYSTEM"`
	`             AccLevel="2"/>`
	`           <DeviceData Path="*" Realm="TAM_ALL"`
	`           AccLevel="1"/>`
	`           </Restricted>`
	`        </Site1>`
	`</WebServer>`

1.6.3 Web server connection timeout

The FP IoT gateway disconnects the connection automatically after 60 s inactivity. If this time is to be extended or shortened, the timeout must be defined separately for incoming and outgoing connections.

The following entry can be added to the web server configuration for outgoing connections such as CBIS:

Web server - connection timeout	
Syntax	**`<KeepConnected _=”`*`periode`*`”/>`**
Description	Inactivity time after which the modem hangs up (only for outgoing connections such as CBIS).

Parameter	*periode* = Time (e.g. 60s, 5m, 1h, etc.)

Example	Disconnection after 5 minutes:
	
	Database path: /ISP/WebServer
	
	`<WebServer>`
	    `<KeepConnected _=”300s”/>`
	    `<ActiveSite _="Site1"/>`
	    `<Site1>`
	        …
	    `</Site1>`
	    `<Site2>`
	        …
	    `</Site2>`
	`</WebServer>`

An entry must be made in the timeout database for incoming connections:

```
[<SetConfig _="USER" ver="v">
    <Timeouts>
        <PPPTimeout _="600s"/>
    </Timeouts>
</SetConfig>]
```

1.6.4 Web server TCP/IP port

The FP IoT gateway web server receives HTTP requests on port 80. If the modem is connected behind a firewall, the HTTP port can also be changed:

Web server – TCP/IP port	
Syntax	**`<Port _=”`*`number`*`”/>`**
Description	TCP/IP port for the connection to the web server.
Parameter	*number* = Port number, e.g. 8080 (default: 80)

Example	Connection to port 8080:
	
	Database path: /ISP/WebServer
	
	`<WebServer>`
	    `<Port _=”8080”/>`
	    `<ActiveSite _="Site1"/>`
	    `<Site1>`
	        …
	    `</Site1>`
	    `<Site2>`
	        …
	    `</Site2>`
	`</WebServer>`

1.6.5 Restricting the web server IP address range

The address range for the FP IoT gateways’ web server can be restricted to a certain range. This can be used to ensure that only IP addresses from a specific IP address range receive access to the FP IoT gateway.
The access restriction can be used to prevent external access to the web server completely.

<table>
<tr><th colspan="2">Web server - restricting the address range</th></tr>
<tr><td><i>Syntax</i></td><td><b><code>&lt;RestrictionStartIP _="<i>Start IP</i>" /&gt;</code></b>
<b><code>&lt;RestrictionEndIP _="<i>End IP</i>" /&gt;</code></b></td></tr>
<tr><td><i>Description</i></td><td>Address range to restrict the web server access
Both options must always be used together.
In order to disable external web server access completely, the <b><i>Start IP</i></b> and the <b><i>End IP</i></b> must be set to the value <code>127.0.0.1</code>.</td></tr>
<tr><td><i>Parameter</i></td><td><i>Start IP</i> = Start IP address as of which access is to be permitted
<i>End IP</i> = End IP address up to which access is to be permitted</td></tr>
<tr><td><i>Example 1</i></td><td>Restricting the address range to <code>192.168.0.100</code> to <code>192.168.0.200</code>:

Database path: /ISP/WebServer
<pre>
<WebServer>
    <Port _="80"/>
    <RestrictionStartIP _="192.168.0.100"/>
    <RestrictionEndIP _="192.168.0.200"/>
    <ActiveSite _="Site1"/>
    <Site1>
        …
    </Site1>
    <Site2>
        …
    </Site2>
</WebServer>
</pre></td></tr>
<tr><td><i>Example 2</i></td><td>Blocking access to the web server completely:

Database path: /ISP/WebServer
<pre>
<WebServer>
    <Port _="80"/>
    <RestrictionStartIP _="127.0.0.1"/>
    <RestrictionEndIP _="127.0.0.1"/>
    <ActiveSite _="Site1"/>
    <Site1>
        …
    </Site1>
    <Site2>
        …
    </Site2>
</WebServer>
</pre></td></tr>
</table>

1.6.6 TiXML TCP/IP port

The Tixi FP IoT gateway can receive and process TiXML commands via TCP/IP. The TCP/IP port for communication is defined in the “TiXML” group on the ISP database.

Database path: /ISP/TiXML

TiXML – TCP/IP Port	
Syntax	**<Port _=”*number*”/>**
Description	CP/IP port for TiXML communication.
Parameter	*number* = Port number, e.g. 8300 (default)

Example	Connection via port 8300:
	
	`<TiXML>`
	`    <Port _="8300"/>`
	`</TiXML>`

In order to communicate with the FP IoT gateway via the network, both the TICO – TiXML console and any other terminal program that supports TCP/IP can be used (e.g. HyperTerminal, Telnet), as well as a virtual serial port via TCP/IP (e.g. Tibbo).

TiXML – restricting access via the address range	
Syntax	**<RestrictionStartIP _="*Start IP*" />** **<RestrictionEndIP _="*End IP*" />**
Description	Address range to restrict TiXML access via the TiXML port. Both options must always be used together. In order to disable external TiXML access completely, the ***Start IP*** and the ***End IP*** must be set to the value `127.0.0.1`.
Parameter	*Start IP* = Start IP address as of which access is to be permitted *End IP* = End IP address up to which access is to be permitted

Example	Blocking access to the web server completely:
	
	`<TiXML>`
	`    <Port _="8300"/>`
	`    <RestrictionStartIP _="127.0.0.1"/>`
	`    <RestrictionEndIP _="127.0.0.1"/>`
	`</TiXML>`

1.7 Web server connectivity settings

1.7.1 Dial-in access

A PPP server must be activated in order to enable a PPP client such as the Windows remote transmission network to establish a connection to the FP IoT gateway. The TCP/IP data for the point-to-point connection is defined in this.
Access via the ISDN syncPPP protocol is only possible if the caller’s phone number was already stored in the modem. After this, remote modem configuration from this connection is only possible via X.75.
This entry is irrelevant for analogue modems.
This is configured in the “PPP_Server” group on the ISP database:

Database path: /ISP/PPP_Server

Web server - PPP access	
Syntax	**<PPP_Server>** **<OwnIP _="ServerIP"/>** **<OwnSubnet _="SubnetMask"/>** **<RemoteIP _="ClientIP"/>** **<AuthentFlags _="Flag"/>** **<DOPx _="*CallerID* "/>** **</PPP_Server>**
Description	PPP server configuration.
Parameter	*ServerIP:* IP address for the PPP server (modem) *SubnetMask:* Subnet mask for the IP network used *ClientIP:* IP address that is assigned to the PPP client (PC) *Flag:* Authentication method when dialling into the PPP 1 = PAP 2 = CHAP 3 = AUTO *x*: Number for the DOP entry (count up) *CallerID:* Phone number for the incoming call
Example	The modem uses the private IP address 192.168.0.1 (default) and assigns the dialled-in device the IP 192.168.0.10. The device with the phone number 0301234567 can dial in via ISDN-syncPPP. <PPP_Server> <OwnIP _="192.168.0.1"/> <OwnSubnet _="255.255.255.0"/> <RemoteIP _="192.168.0.10"/> <AuthentFlags _="3"/> <DOP1 _="0301234567"/> <DOP2 _=""/> <DOP3 _=""/> </PPP_Server>

1.7.2 Call back initiation service CBIS (only for 5th generation devices)

A further option for accessing the FP IoT gateway web server is the “call back initiation service” (CBIS). This service uses a dial-in via telephone lines and is only available on the older device generation “G5”, i.e. for devices in the H400 series. The CBIS service is no longer available in the current generation “G6” and will no longer be available in the future, as that telephone dial-in technology is no longer up to date and is no longer supported by many service providers.
When the CBIS procedure is started, which can be done by phone call or via the EventHandler (see the TiXML Reference Manual), the FP IoT gateway dials in to the internet and sends an e-mail with the current IP address to the pre-set recipient.

The CBIS function requires internet access (see the TiXML Reference Manual) and is part of the ISP database.

Database path: /ISP/CBIS

CBIS – Call back initiation service	
Syntax	**<CBIS>** **<ServerName _="*Address*"/>** **<PhoneNumber _="*CallerID*"/>** **<Account _="*Recipient*"/>**

	<ResponseTime _="*IdleTime*"/> **</CBIS>**
Description	Call back initiation service configuration.
Parameter	*Address*: Mail server *CallerID:* CallerID for the caller *Recipient:* CBIS e-mail recipient *IdleTime:* Waiting time until the first access (connect)
Example	Internet access via “Freenet” is configured. The CBIS e-mail is sent via the Server “193.101.167.194” to the recipient cbis@devicecontrolweb.com. The CBIS procedure is triggered by a call from the number “0301234567”: <CBIS> <ServerName _="193.101.167.194"/> <PhoneNumber _="0301234567"/> <Account _="cbis@devicecontrolweb.com "/> <ResponseTime _="60s"/> </CBIS>

The CBIS CallerID trigger requires a special system EventHandler with the CBIS command:

Database path: /EVENTS/EventHandler/System

```
<System>
    <CBISRequest>
     <CBIS/>
    </CBISRequest>
</System>
```

The CBIS procedure can be started using the CBIS command in any EventHandler (see the TiXML Reference Manual).

As the result of the CBIS procedure, the recipient “Account” receives an e-mail from the sender CBIS@Tixi.Com with the following subject:

```
Subject: CBIS Connect IP-Address   (e.g. CBIS Connect 192.168.0.1)
```

The message body contains the SITE_TAG (see the “Tixi HTTP data interface” manual).

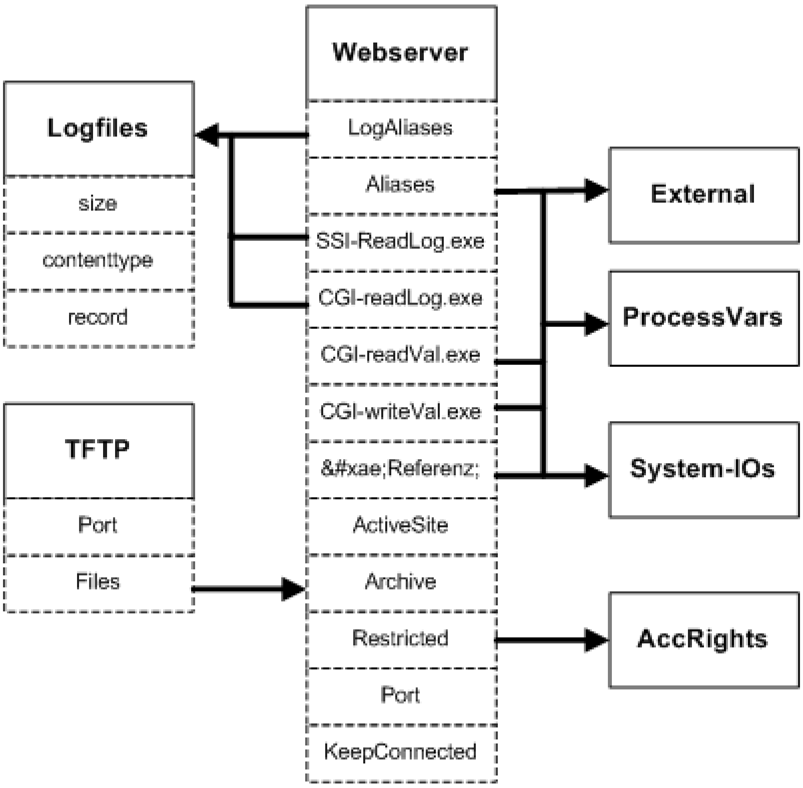
The recipient then has the time specified in “ResponseTime” to access the web server.

1.7.2.1 Changing the CBIS sender address

Some mail servers do not allow messages to be sent using the address CBIS@Tixi.Com.
In this case, a MessageJobTemplate (see the TiXML Reference Manual) can be generated, in which the sender and recipient can be selected from the address book:

```
<SendIP _="CBIS">
  <Sender _="/D/AddressBook/MySelf"/>
  <Recipient _="/D/AddressBook/Receiver_0"/>
..
..
</SendIP >
```

2 Appendix: Project structure and coherence



Notes