PlantVisorPRO

Plant supervision

CAREL

ENG Software Development Kit ver. 2.0

LEGGI E CONSERVA QUESTE ISTRUZIONI

READ AND SAVE THESE INSTRUCTIONS

Integrated Control Solutions & Energy Savings

Contents

1. KEY

	**Note**	**Notes** contain important information, highlighted after the text they refer to.
- -	**Example**	Example code is always enclosed within borders
	Typical case	Customised application example

2. CONTEXT

SDK for PlantVisorPRO has been written in JAVA programming language.
The pages displaying graphs are not simple HTML templates but rather JSP documents (Java Server Pages).
The main advantages deriving from the introduction of this new technology for the customisation of PlantVisorPRO are:

- As JAVA is very widely and commonly used among developers, it's much easier to source development resources;
- The introduction of JSP provides greater flexibility and development power;
- JAVA language offers a vast range of methods and functions that are already available and tested.

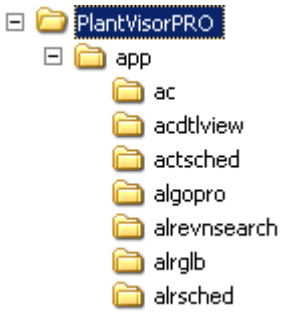
Statement syntax is similar to CAREL proprietary script.
All statements start and end with the following scripting syntax:
<% %>

The following simple example shows how to declare an integer variable:
<% int i = 0; %>

For complete information on JAVA language and the creation of JSP pages, the official SUN documents are available at the following address: http://java.sun.com/j2ee/1.4/docs/tutorial/doc/

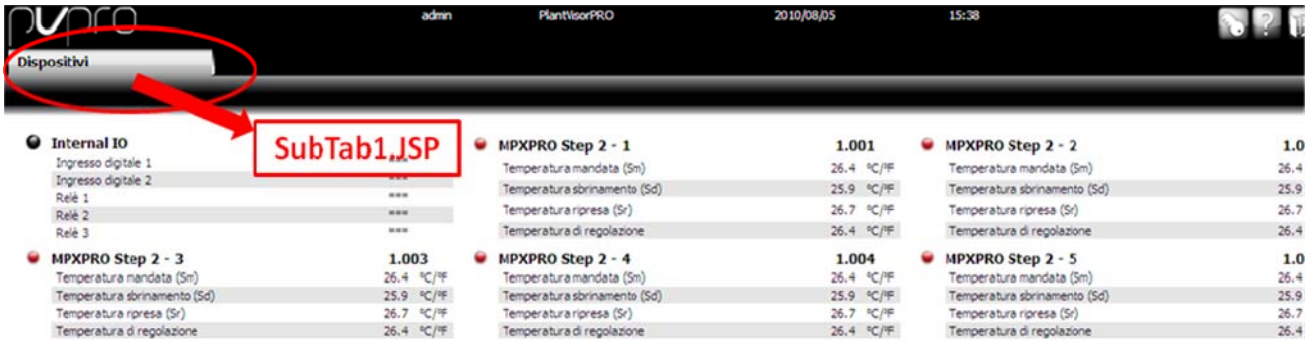
3. FILE STRUCTURE

The supervisor has a clearly defined directory structure in the file system. Each function offered by PlantVisorPRO is grouped in a directory. The set of all functions is available under the following directory: **C:\Carel\PlantVisorPRO\engine\webapps\PlantVisorPRO\app\,** where APP stands for applications.



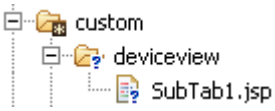
Each subdirectory represents a supervisor function and contains JSP files relating to the various tabs.

The figure above shows the “deviceview” directory, this is the directory for device display transactions called from the following menu: INSTALLATION -> DEVICES.
The “SubTab1.jsp” file represents the device tab.

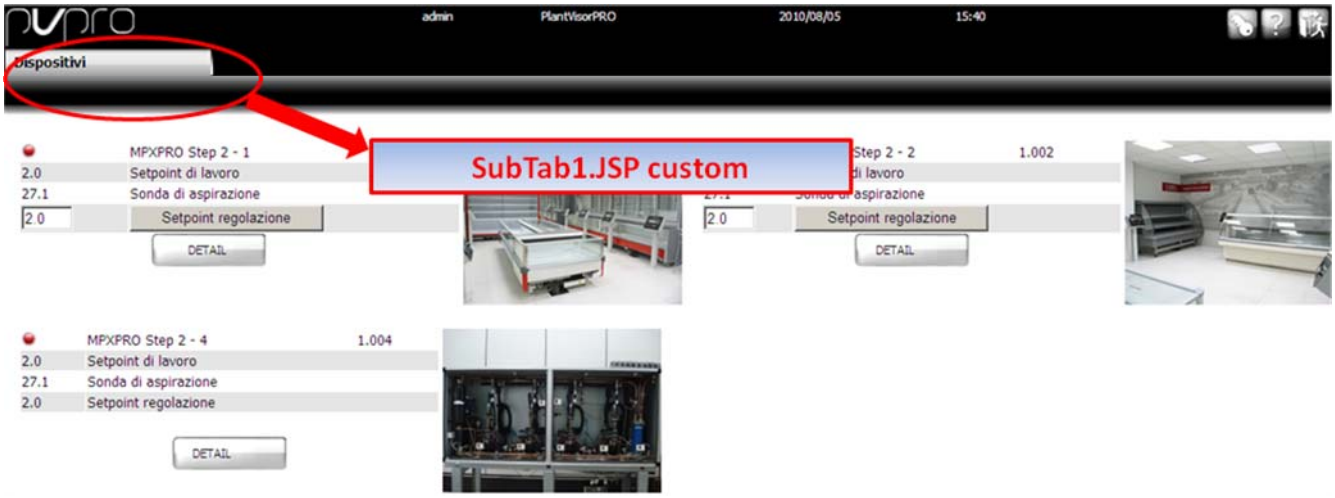


When needing to overwrite standard supervisor pages, the following starting path is used: **C:\Carel\PlantVisorPRO\engine\webapps\PlantVisorPRO\custom\.**

Starting from the **custom** directory, the user needs to create a new directory with the same name as the one being overwritten.
The example shown below shows how to overwrite the device tab presented above.
The user has to create a **deviceview** directory (the same name as in the standard directory) under the **custom** directory.
Inside this, the user can create a JSP file (empty if necessary) with the same name as the standard: SubTab1.jsp



This directory structure tells the supervisor to use the new custom section as soon as this is created.
In this way, the standard pages are not physically overwritten, meaning the user can retrieve them at a later stage by simply deleting the corresponding custom directory.

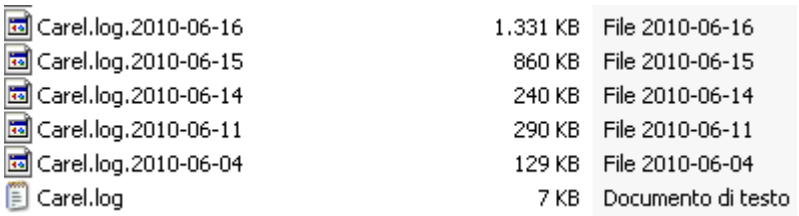


4. LOG STRUCTURE AND ERROR MANAGEMENT

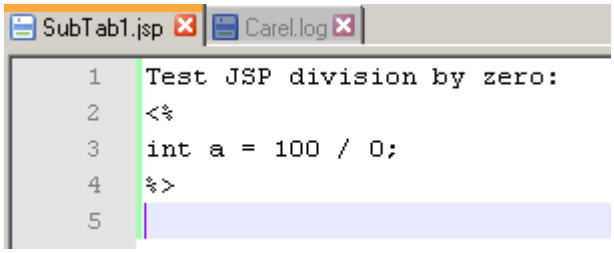
As regards the data logs, the supervisor again has a clearly defined structure for saving information relating to:

- System events;
- System errors;

The standard path is: **C:\Carel\PlantVisorPRO\log.**
The supervisor writes the **Carel.log** file to this path.
This file contains the information described above. The file is updated daily, meaning that it's renamed every day with data from the previous day and is completely rewritten so as to prevent the file from becoming too big.
The following is an example of the directory structure containing the various log files, where the file currently being used is Carel.log:



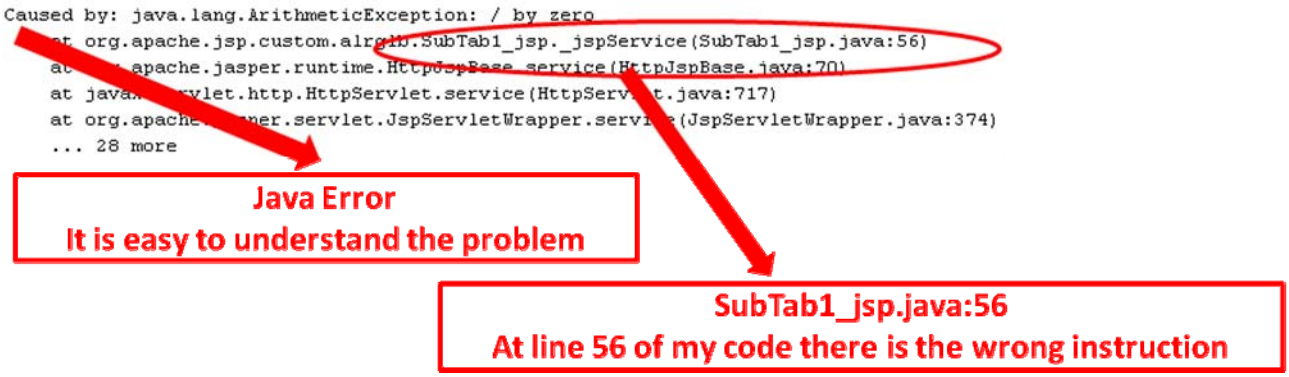
This file describes any programming errors on the custom JSP pages.
As an example, a forced error is created on a JSP by trying to divide by zero and assign the result to a numeric variable:



This type of statement is not allowed, being incorrect from a mathematical point of view. When accessing the page it will be displayed as follows:

To understand the problem, the procedure is to check the Carel.log file for information that may be useful in resolving the error.

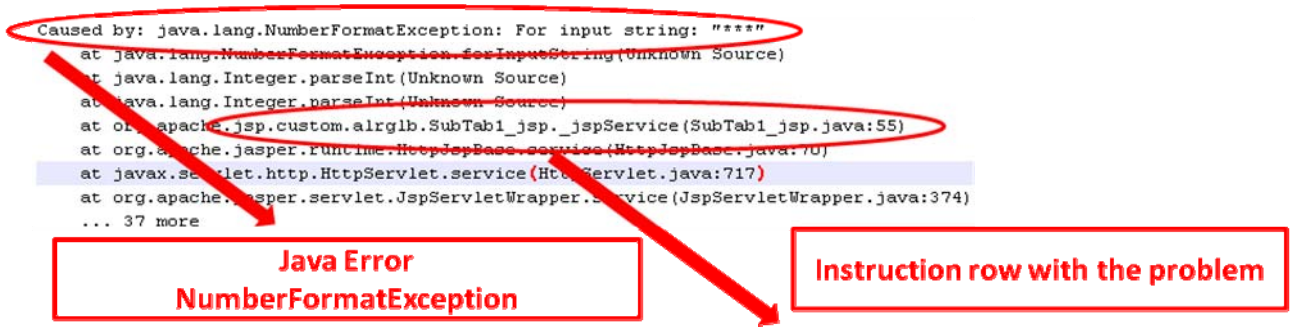
Each line in the file describes the date and time when the information was recorded. Consequently, going to the time the error occurred, the following line is shown:



The JSP file being executed effectively becomes a JAVA file and is thus compiled. The line of code with the error (56) refers to the compiled file and not the initial file.
To see the code of the compiled JSP file, go to the following path:
C:\Carel\PlantVisorPRO\engine\work\Catalina\localhost\PlantVisorPRO\org\apache\jsp\custom\alrglb\SubTab1_jsp.java
This required path relates to the global alarm functions (active and closed) and consequently the directory will be **alrglb**.

4.1 Typical case: DEVICE OFFLINE

Another typical error concerning the conversion of a value retrieved from the field into numerical format occurs when the peripheral is offline and the value returned is equal to "***".
The error shown on the screen will always be presented using the same page as shown above. In the log, the error will be highlighted in the same way, only the reason will be different:



These problems can only be resolved correctly using the structure offered by Java language called **try catch**.
This is used to capture the exceptions generated by the code and manage these without affecting the use of the page as a whole.
In the case of device OFFLINE, the following code could be implemented.

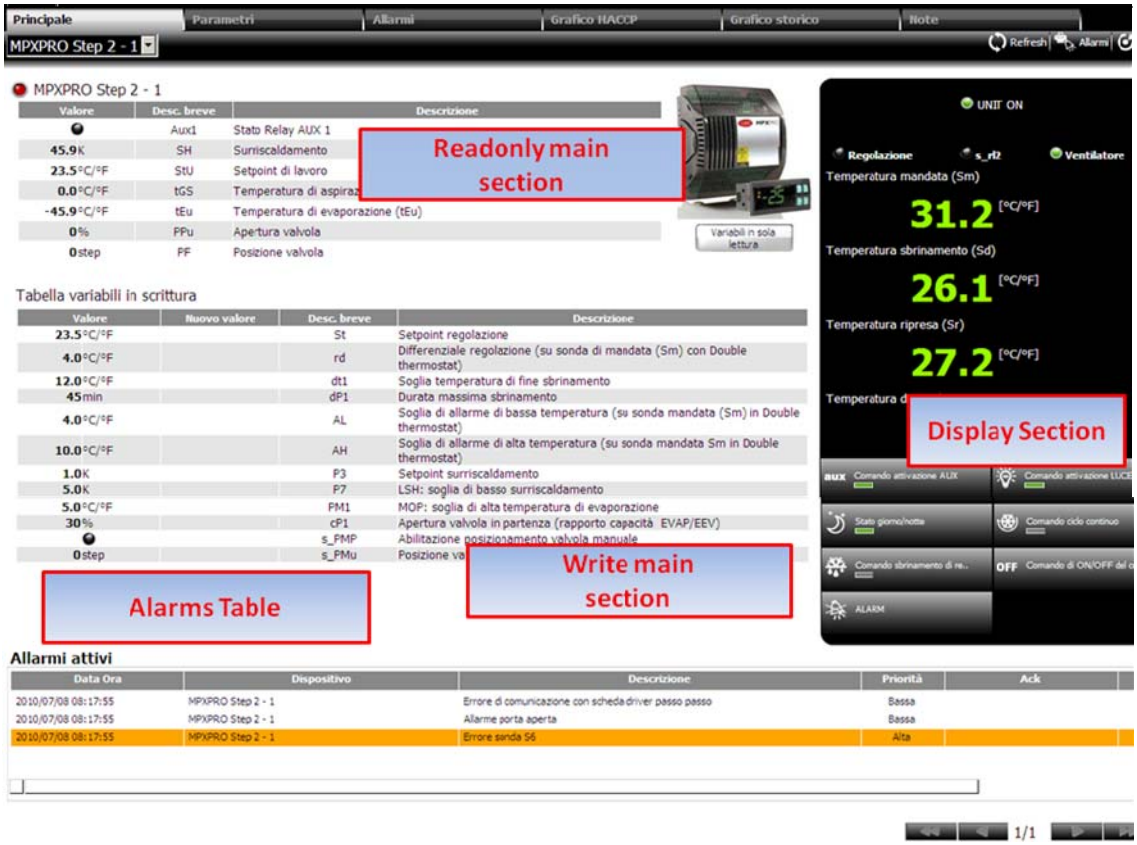
SubTab1.jsp

```
1  <%@ page language="java"
2      import="com.carel.supervisor.presentation.sdk.util.Sfera"
3      import="com.carel.supervisor.presentation.helper.ServletHelper"
4      import="com.carel.supervisor.presentation.session.UserSession"
5  %>
6
7  <jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" >
8      <jsp:setProperty name="CurrUnit" property="req" value="<%=request%>" />
9  </jsp:useBean>
10
11 <%
12     String valueFromFieldVariable_DI1 = CurrUnit.getVariable("DI1").getValue();
13     int intValueFromField = 0;
14     try
15     {
16         intValueFromField = Integer.parseInt(CurrUnit.getVariable(valueFromFieldVariable_DI1).getValue());
17         // I need to divide by ten
18         intValueFromField = intValueFromField / 10;
19         // Get back to string
20         valueFromFieldVariable_DI1 = String.valueOf(intValueFromField);
21     }
22     catch (Exception e) {
23         valueFromFieldVariable_DI1 = "****";
24     }
25 %>
26 <html>
27     <tr>
28         <td>Value from field: </td>
29         <td><%= valueFromFieldVariable_DI1 %> </td>
30     </tr>
31 </html>
```

TRY CATCH BLOCK

5. CUSTOM DEVICE DETAILS

Each standard device on the supervisor has a detail page with a clearly defined structure:



To customise a detail page, users have the following two options available:

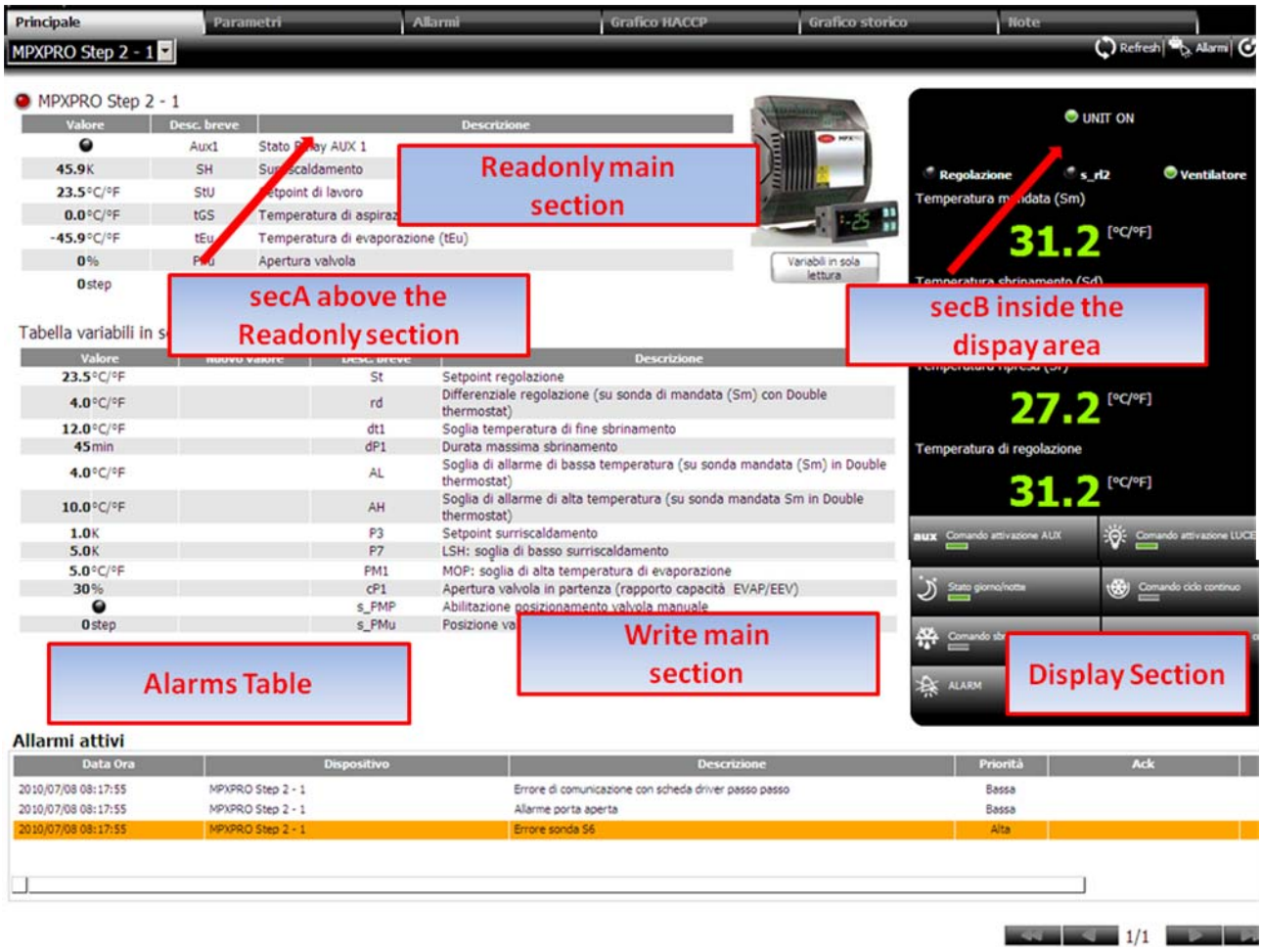
- Partial customisation;
- Complete customisation.

5.1 Partial customisation

Partial customisation allows users to start from the standard device detail page and add two custom sections. These sections are defined as:

- secA;
- secB;

The name of these two files will be: secA.jsp and secB.jsp.
If implemented, these are automatically added by the architecture to the standard page in the following positions:



In this way, customisations that require parts of the standard page (table of read-only variables, table of read/write variables and list of alarms) while adding one or two special tables can use this development method.
Below are some examples of partial customisation.

5.1.1 Typical case: ADDING A TABLE WITH INSTRUMENT VARIABLES

Below is a table created in HTML displaying two variables from MPXPRO, s_PMP and s_PMu. The structure and style sheet classes applied are the same as those used on the standard page for read-only variables.

```
<%@ page language="java"
	import="com.carel.supervisor.presentation.helper.ServletHelper"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />

<TABLE id="readonlyvartablecustom" class="table" cellSpacing=1 cellPadding=0 width="98%">
<THEAD>
	<TR>
		<TH class=th height=18 width="18%" colSpan=2>Valore</TH>
		<TH class=th height=18 width="15%">Desc. breve</TH>
		<TH class=th height=18 width=*>Descrizione</TH>
	</TR>
</THEAD>
<TBODY>
	<TR class=Row1 height=21>
		<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_PMP").getRefreshableValue()%></TD>
		<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_PMP").getMUnit()%></NOBR></TD>
		<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMP").getDescr1()%></TD>
		<TD class=standardTxt><%=CurrUnit.getVariable("s_PMP").getDescription()%></TD>
	</TR>
	<TR class=Row2 height=21>
		<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_PMu").getRefreshableValue()%></TD>
		<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_PMu").getMUnit()%></NOBR></TD>
		<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMu").getDescr1()%></TD>
		<TD class=standardTxt><%=CurrUnit.getVariable("s_PMu").getDescription()%></TD>
	</TR>
</TBODY>
</TABLE>
```

Where **CurrUnit** is the memory object that represents the current device.

The result displayed will be as follows:

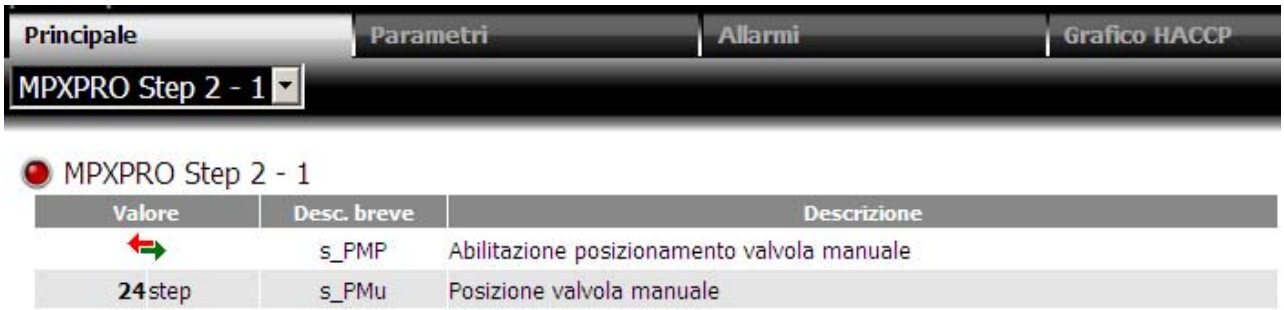
5.1.2 Typical case: ADDING CUSTOM IMAGES TO TABLES

Different images can be displayed depending on the value of the variable.

```
<%@ page language="java"
	import="com.carel.supervisor.presentation.helper.ServletHelper"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />

<TABLE id="readonlyvartablecustom" class="table" cellSpacing=1 cellPadding=0 width="98%">
<THEAD>
	<TR>
		<TH class=th height=18 width="18%" colSpan=2>Valore</TH>
		<TH class=th height=18 width="15%">Desc. breve</TH>
		<TH class=th height=18 width=*>Descrizione</TH>
	</TR>
</THEAD>
<TBODY>
	<TR class=Row1 height=21>
		<TD class=standardTxt width="9%" align=center colSpan=2>
		<%=CurrUnit.getVariable("s_PMP").getRefreshableAssint("<img
src='custom/dtlview_section/mpxprostep2/norevpx.png';<img
src='custom/dtlview_section/mpxprostep2/revpx.png'>")%>
		</TD>
		<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMP").getDescr1()%></TD>
		<TD class=standardTxt><%=CurrUnit.getVariable("s_PMP").getDescription()%></TD>
	</TR>
	<TR class=Row2 height=21>
		<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_PMu").getRefreshableValue()%></TD>
		<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_PMu").getMUnit()%></NOBR></TD>
		<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMu").getDescr1()%></TD>
		<TD class=standardTxt><%=CurrUnit.getVariable("s_PMu").getDescription()%></TD>
	</TR>
</TBODY>
</TABLE>
```

In this example, the variable s_PMP is represented with a custom image. The different image is selected according to its status. For further details, see the documents on **CurrVar**, **getRefreshableAssint** method.
The resulting page will be the same as shown in the previous example, with the custom image displayed in place of the normal status LED.



5.1.3 Typical case: ADDING FIELDS FOR SETTING A VARIABLE

A value can be set on the screen using an editable field.

```
<%@ page language="java"
	import="com.carel.supervisor.presentation.helper.ServletHelper"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />

<FORM name="frmSecA" id="frmSecA">
<TABLE id="readonlyvartablecustom" class="table" cellSpacing=1 cellPadding=0 width="98%">
<THEAD>
	<TR>
		<TH class=th height=18 width="18%" colSpan=2>Valore</TH>
		<TH class=th height=18 width="18%">Nuovo</TH>
		<TH class=th height=18 width="15%">Desc. breve</TH>
		<TH class=th height=18 width=*>Descrizione</TH>
	</TR>
</THEAD>
<TBODY>
	<TR class=Row1 height=21>
		<TD class=standardTxt width="9%" align=center colSpan=2>
		<%=CurrUnit.getVariable("s_PMP").getRefreshableAssint("<img
src='custom/dtlview_section/mpxprostep2/restart.png'>;<img
src='custom/dtlview_section/mpxprostep2/restart_red.png'>")%>
		</TD>
		<TD></TD>
		<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMP").getDescr1()%></TD>
		<TD class=standardTxt><%=CurrUnit.getVariable("s_PMP").getDescription()%></TD>
	</TR>
	<TR class=Row2 height=21>
		<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_PMu").getRefreshableValue()%></TD>
		<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_PMu").getMUnit()%></NOBR></TD>
		<TD class=standardTxt width="9%" align=center>
		<INPUT NAME="<%=CurrUnit.getVariable("s_PMu").getPostName()%>"
			id="<%=CurrUnit.getVariable("s_PMu").getPostName()%>" class="lswtype"
			onkeydown="checkOnlyAnalog(this,event);"
			onblur="sdk_checkMinMaxValue(this,9,16);checkOnlyAnalogOnBlur(this);" type="text"
size="5"/>
		</TD>
		<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMu").getDescr1()%></TD>
		<TD class=standardTxt><%=CurrUnit.getVariable("s_PMu").getDescription()%></TD>
	</TR>
</TBODY>
</TABLE>
</FORM>
```

In this example, an editable text field is used to set the value for the variable s_Pmu.

Examining the code for creating the text field in detail, the user can exploit a Javascript function called **sdk_checkMinMaxValue(this,min_value,max_value)**
With this function the user can declare the range of possible values for the variable.

- MIN_VALUE -> Minimum value that can be set for the variable;
- MAX_VALUE -> Maximum value that can be set for the variable;
- THIS -> Reference to the input object (Javascript standard)

This Javascript function is called whenever the text field loses focus (cursor.)
Together with this function a further two are used, one called when the text field loses focus and the other called when pressing any key so that the architecture makes sure the value entered is numeric and not a letter.
These functions can be used to control the fields on custom pages.

The text field is associated with a style, the same as the standard used by the other fields on the supervisor, **class="lswtype"**

 All the variables that need to be sent to the device must be placed on a FORM. These forms must have a standard name**: frmSecA** as shown in the example above.

5.1.4 Typical case: ADDING BUTTONS FOR SETTING A VARIABLE (RESET ALARMS)

Partial customisation includes the possibility to add buttons to directly set digital variables. The typical case is to RESET device alarms.

```
<%@ page language="java"
	import="com.carel.supervisor.presentation.helper.ServletHelper"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />

<FORM name="frmSecA" id="frmSecA">
<TABLE id="readonlyvartablecustom" class="table" cellSpacing=1 cellPadding=0 width="98%">
<THEAD>
	<TR>
		<TH class=th height=18 width="18%" colSpan=2>Valore</TH>
		<TH class=th height=18 width="18%">Nuovo</TH>
		<TH class=th height=18 width="15%">Desc. breve</TH>
		<TH class=th height=18 width=*>Descrizione</TH>
	</TR>
</THEAD>
<TBODY>
	<TR class=Row1 height=21>
		<TD class=standardTxt width="9%" align=center colSpan=2>
		<%=CurrUnit.getVariable("s_PMP").getRefreshableAssint("<img
src='custom/dtlview_section/mpxprostep2/restart.png';<img
src='custom/dtlview_section/mpxprostep2/restart_red.png'>")%>
		</TD>
		<TD></TD>
		<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMP").getDescr1()%></TD>
		<TD class=standardTxt><%=CurrUnit.getVariable("s_PMP").getDescription()%></TD>
	</TR>
	<TR class=Row2 height=21>
		<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_PMu").getRefreshableValue()%></TD>
		<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_PMu").getMUnit()%></NOBR></TD>
		<TD class=standardTxt width="9%" align=center>
		<INPUT NAME="<%=CurrUnit.getVariable("s_PMu").getPostName()%>"
			id="<%=CurrUnit.getVariable("s_PMu").getPostName()%>" class="lswtype"
			onkeydown="checkOnlyAnalog(this,event);"
			onblur="sdk_checkMinMaxValue(this,9,16);checkOnlyAnalogOnBlur(this);" type="text"
size="5"/>
		</TD>
		<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMu").getDescr1()%></TD>
		<TD class=standardTxt><%=CurrUnit.getVariable("s_PMu").getDescription()%></TD>
	</TR>
</TBODY>
</TABLE>
</FORM>
<TABLE id="btttablecustom"  cellSpacing=1 cellPadding=0 width="98%">
```

```
<TR>
	<TD width="50%">
		<TABLE class="table" cellspacing="1" cellpadding="0" width="98%">
		<THEAD>
			<TR>
				<TH class=th height=18 >Pulsante</TH>
				<TH class=th height=18 width=*>Descrizione</TH>
			</TR>
		</THEAD>
		<TBODY>
			<TR class="Row2" height="21">
				<TD class="standardTxt" align="center"><nobr>

	<%=CurrUnit.getVariable("s_PMP").getSimpleButton(1,"1","images/button/relay_on_black.png","Abilitazione")
%>
				</TD>
				<TD class="standardTxt"
align="center"><nobr><%=CurrUnit.getVariable("s_PMP").getDescription()%></nobr></TD>
			</TR>
			<TR class="Row2" height="21">
				<TD class="standardTxt" align="center"><nobr>

	<%=CurrUnit.getVariable("RESET").getSimpleButton(1,"1","images/button/alarmreset_on_black.png","Abilitazi
one")%>
				</TD>
				<TD class="standardTxt" align="center"><nobr>Reset Allarmi</nobr></TD>
			</TR>
			<TR class="Row2" height="21">
				<TD class="standardTxt" align="center"><nobr>

	<%=CurrUnit.getVariable("LIGHT").getSimpleButton(1,"1","images/button/light_on_black.png","Abilitazione")%
>
				</TD>
				<TD class="standardTxt" align="center"><nobr>Luce</nobr></TD>
			</TR>
		<TBODY>
		</TABLE>
	</TD>
	<TD width="50%"></TD>
</TR>
</TABLE>
```

All the variables that need to be sent to the device must be placed on a FORM. These forms must have a standard name**: frmSecA** as shown in the example above.
As regards the buttons, these don't require a FORM as the architecture automatically creates the HTML code for sending the data.

5.1.5 Typical case: CHECKING USER RIGHTS TO SET A VARIABLE

All the cases involving setting variables described above can also feature a check on user rights. Supervisor user accounts have two types of rights for writing data:

- SERVICE;
- MANUFACTURER;

```
<%@ page language="java"
	import="com.carel.supervisor.presentation.helper.ServletHelper"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<jsp:useBean id="CurrUser" class="com.carel.supervisor.presentation.sdk.obj.CurrUser" scope="session"/>

<%
 CurrUser.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
%>

<FORM name="frmSecA" id="frmSecA">
<TABLE id="readonlyvartablecustom" class="table" cellSpacing=1 cellPadding=0 width="98%">
<THEAD>
	<TR>
		<TH class=th height=18 width="18%" colSpan=2>Valore</TH>
		<TH class=th height=18 width="18%">Nuovo</TH>
		<TH class=th height=18 width="15%">Desc. breve</TH>
		<TH class=th height=18 width=*>Descrizione</TH>
	</TR>
</THEAD>
<TBODY>
	<TR class=Row1 height=21>
		<TD class=standardTxt width="9%" align=center colSpan=2>
		<%=CurrUnit.getVariable("s_PMP").getRefreshableAssint("<img
src='custom/dtlview_section/mpxprostep2/restart.png'>;<img
src='custom/dtlview_section/mpxprostep2/restart_red.png'>")%>
		</TD>
		<TD></TD>
		<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMP").getDescr1()%></TD>
		<TD class=standardTxt><%=CurrUnit.getVariable("s_PMP").getDescription()%></TD>
	</TR>
	<TR class=Row2 height=21>
		<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_PMu").getRefreshableValue()%></TD>
		<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_PMu").getMUnit()%></NOBR></TD>
		<TD class=standardTxt width="9%" align=center>
		<%if(CurrUser.haveServicesRight()) {%>
		<INPUT NAME="<%=CurrUnit.getVariable("s_PMu").getPostName()%>"
			id="<%=CurrUnit.getVariable("s_PMu").getPostName()%>" class="lswtype"
			onkeydown="checkOnlyAnalog(this,event);"
			onblur="sdk_checkMinMaxValue(this,9,16);checkOnlyAnalogOnBlur(this);" type="text"
size="5"/>
		<%}%>
		</TD>
		<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMu").getDescr1()%></TD>
		<TD class=standardTxt><%=CurrUnit.getVariable("s_PMu").getDescription()%></TD>
	</TR>
	<TR class=Row1 height=21>
		<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_PMu").getRefreshableValue()%></TD>
		<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_PMu").getMUnit()%></NOBR></TD>
		<TD class=standardTxt width="9%" align=center>
		<%if(CurrUser.haveManufacturerRight()) {%>
		<INPUT NAME="<%=CurrUnit.getVariable("s_PMu").getPostName()%>"
			id="<%=CurrUnit.getVariable("s_PMu").getPostName()%>" class="lswtype"
			onkeydown="checkOnlyAnalog(this,event);"
```

```
                                onblur="sdk_checkMinMaxValue(this,9,16);checkOnlyAnalogOnBlur(this);" type="text"
size="5"/>
                        <%}%>
                        </TD>
                        <TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMu").getDescr1()%></TD>
                        <TD class=standardTxt><%=CurrUnit.getVariable("s_PMu").getDescription()%></TD>
                </TR>
</TBODY>
</TABLE>
</FORM>
<TABLE id="btttablecustom"  cellSpacing=1 cellPadding=0 width="98%">
<TR>
        <TD width="50%">
                <TABLE class="table" cellspacing="1" cellpadding="0" width="98%">
                <THEAD>
                        <TR>
                                <TH class=th height=18 >Pulsante</TH>
                                <TH class=th height=18 width=*>Descrizione</TH>
                        </TR>
                </THEAD>
                <TBODY>
                        <TR class="Row2" height="21">
                                <TD class="standardTxt" align="center"><nobr>

        <%=CurrUnit.getVariable("s_PMP").getSimpleButton(1,"1","images/button/relay_on_black.png","Abilitazione")
%>
                                </TD>
                                <TD class="standardTxt"
align="center"><nobr><%=CurrUnit.getVariable("s_PMP").getDescription()%></nobr></TD>
                        </TR>
                        <TR class="Row2" height="21">
                                <TD class="standardTxt" align="center"><nobr>

        <%=CurrUnit.getVariable("RESET").getSimpleButton(1,"1","images/button/alarmreset_on_black.png","Abilitazi
one")%>
                                </TD>
                                <TD class="standardTxt" align="center"><nobr>Reset Allarmi</nobr></TD>
                        </TR>
                        <TR class="Row2" height="21">
                                <TD class="standardTxt" align="center"><nobr>

        <%=CurrUnit.getVariable("LIGHT").getSimpleButton(1,"1","images/button/light_on_black.png","Abilitazione")%
>
                                </TD>
                                <TD class="standardTxt" align="center"><nobr>Luce</nobr></TD>
                        </TR>
                <TBODY>
                </TABLE>
        </TD>
        <TD width="50%"></TD>
</TR>
</TABLE>
```

For further details see the documents on **CurrUser**, **haveServicesRight, haveManufacturerRight** methods.

5.1.6 File system for partial customisation

The custom sections created by partial customisation must be copied to the following path:
C:\Carel\PlantVisorPRO\engine\webapps\PlantVisorPRO\custom\dtlview_section\DEVICE_FOLDER
where DEVICE_FOLDER is replaced with the device ID code.

For certain devices, partial customisations may already be available on the supervisor by default.

If adding a new custom device created using DEVICE CREATOR, when loading it onto the supervisor, the supervisor itself automatically creates the directory to contain the partial customisation data for the device. This directory will have the same name as the code that uniquely identifies the device.

5.2 Complete customisation

Complete customisation of the device detail allows the user to completely rewrite the page. The partial customisation examples shown above can also be applied to total customisation.
The following is a practical example of this type of customisation:

The JSP code (SubTab1) used to create the page is shown below:

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<jsp:useBean id="CurrUser" class="com.carel.supervisor.presentation.sdk.obj.CurrUser" scope="session"/>
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUser.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<script>
PVPK_ActiveRefresh(30);
PVPK_addButtons();
</script>
<input type="hidden" id="s_minval" value="Minimum value: " />
<input type="hidden" id="s_maxval" value="Maximum value: " />

<table border="0" width="100%" cellpadding="1" cellspacing="1">
<tr>
        <td width="30%">
                <table border="0" width="100%" cellpadding="1" cellspacing="1">
                        <tr>
                                <td width="50%" valign="top"><img src="images/devices/mpxpro.jpg"
border="1"></td>
                                <td width="50%" valign="top">
```

```
                                        <table border="0" width="100%" cellpadding="1" cellspacing="1">
                                                <tr>
                                                        <td><%=Sfera.assint(CurrUnit.getStatus(),"<img
src='images/led/L0.gif'>;<img  src='images/led/L1.gif'>;<img  src='images/led/L2.gif'>;<img  src='images/led/L3.gif'>",
"<img src='images/led/L0.gif'>")%></td>
                                                        <td><%=CurrUnit.getLine()%></td>

        <td><%=CurrUnit.getVariable("s_cONOFF").getRefreshableAssint("UNIT ON;UNIT OFF","***")%></td>
                                                </tr>
                                                <tr><td><br/></td></tr>
                                                <tr>
                                                        <td height="30" width="350" align="left"
class="groupCategory_small" colspan="3" onclick="PVP_goToVariables('readonly')">
                                                        READ ONLY
                                                        </td>
                                                </tr>
                                        </table>
                                </td>
                        </tr>
                </table>
        </td>
        <td width="70%" valign="top">
                <TABLE id="readonlyvartablecustom" class="table" cellSpacing=1 cellPadding=0 width="98%">
                <THEAD>
                        <TR>
                                <TH class=th height=18 width="18%" colSpan=2>Valore</TH>
                                <TH class=th height=18 width="15%">Desc. breve</TH>
                                <TH class=th height=18 width=*>Descrizione</TH>
                        </TR>
                </THEAD>
                <TBODY>
                        <TR class=Row1 height=21>
                                <TD class=standardTxt width="9%" align=center colSpan=2>
                                <%=CurrUnit.getVariable("s_PMP").getRefreshableAssint("<img
src='custom/dtlview_section/mpxprostep2/restart.png'>;<img
src='custom/dtlview_section/mpxprostep2/restart_red.png'>")%>
                                </TD>
                                <TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMP").getDescr1()%></TD>
                                <TD
class=standardTxt><%=CurrUnit.getVariable("s_PMP").getDescription()%></TD>
                        </TR>
                        <TR class=Row2 height=21>
                                <TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_SuctionProbe").getRefreshableValue()%></TD>
                                <TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_SuctionProbe").getMUnit()%></NOBR></TD>
                                <TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_SuctionProbe").getDescr1()%></TD>
                                <TD
class=standardTxt><%=CurrUnit.getVariable("s_SuctionProbe").getDescription()%></TD>
                        </TR>
                        <TR class=Row1 height=21>
                                <TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_SetpointWork").getRefreshableValue()%></TD>
                                <TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_SetpointWork").getMUnit()%></NOBR></TD>
                                <TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_SetpointWork").getDescr1()%></TD>
                                <TD
class=standardTxt><%=CurrUnit.getVariable("s_SetpointWork").getDescription()%></TD>
                        </TR>
                </TBODY>
                </TABLE>
        </td>
</tr>
<tr>
        <td width="30%" valign="top">
```

```
		<TABLE id="btttablecustom"  cellSpacing=1 cellPadding=0 width="98%">
			<TR>
				<TD width="100%">
					<TABLE class="table" cellspacing="1" cellpadding="0" width="98%">
					<THEAD>
						<TR>
							<TH class=th height=18 >Pulsante</TH>
							<TH class=th height=18 width=*>Descrizione</TH>
						</TR>
					</THEAD>
					<TBODY>
						<TR class="Row2" height="21">
							<TD class="standardTxt" align="center"><nobr>

	<%=CurrUnit.getVariable("s_PMP").getSimpleButton(1,"1","images/button/relay_on_black.png","Abilitazione")
%>
							</TD>
							<TD class="standardTxt"
align="center"><nobr><%=CurrUnit.getVariable("s_PMP").getDescription()%></nobr></TD>
						</TR>
						<TR class="Row2" height="21">
							<TD class="standardTxt" align="center"><nobr>

	<%=CurrUnit.getVariable("RESET").getSimpleButton(2,"1","images/button/alarmreset_on_black.png","Abilitazi
one")%>
							</TD>
							<TD class="standardTxt" align="center"><nobr>Reset
Allarmi</nobr></TD>
						</TR>
						<TR class="Row2" height="21">
							<TD class="standardTxt" align="center"><nobr>

	<%=CurrUnit.getVariable("s_cLUX").getSimpleButton(3,"1","images/button/light_on_black.png","Abilitazione")
%>
							</TD>
							<TD class="standardTxt"
align="center"><nobr>Luce</nobr></TD>
						</TR>
					<TBODY>
					</TABLE>
				</TD>
				<TD width="50%"></TD>
			</TR>
		</TABLE>
	</td>
	<td width="70%" valign="top">
		<form name="formSettableVars" id="formSettableVars">
		<TABLE id="readonlyvartablecustom" class="table" cellSpacing=1 cellPadding=0 width="98%">
		<THEAD>
			<TR>
				<TH class=th height=18 width="18%" colSpan=2>Valore</TH>
				<TH class=th height=18 width="18%">Nuovo</TH>
				<TH class=th height=18 width="15%">Desc. breve</TH>
				<TH class=th height=18 width=*>Descrizione</TH>
			</TR>
		</THEAD>
		<TBODY>
			<TR class=Row1 height=21>
				<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_PMu").getRefreshableValue()%></TD>
				<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_PMu").getMUnit()%></NOBR></TD>
				<TD class=standardTxt width="9%" align=center>
				<%if(CurrUser.haveServicesRight()) {%>
				<INPUT NAME="<%=CurrUnit.getVariable("s_PMu").getPostName()%>"
					id="<%=CurrUnit.getVariable("s_PMu").getPostName()%>"
class="lswtype"
					onkeydown="checkOnlyAnalog(this,event);"
```

```
onblur="sdk_checkMinMaxValue(this,9,16);checkOnlyAnalogOnBlur(this);" type="text" size="5"/>
                                        <%}%>
                                        </TD>
                                        <TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMu").getDescr1()%></TD>
                                        <TD
class=standardTxt><%=CurrUnit.getVariable("s_PMu").getDescription()%></TD>
                                </TR>
                                <TR class=Row2 height=21>
                                        <TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("St").getRefreshableValue()%></TD>
                                        <TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("St").getMUnit()%></NOBR></TD>
                                        <TD class=standardTxt width="9%" align=center>
                                        <INPUT NAME="<%=CurrUnit.getVariable("St").getPostName()%>"
                                                id="<%=CurrUnit.getVariable("St").getPostName()%>" class="lswtype"
                                                onkeydown="checkOnlyAnalog(this,event);"
                                                onblur="sdk_checkMinMaxValue(this,-
10,5);checkOnlyAnalogOnBlur(this);" type="text" size="5"/>
                                        </TD>
                                        <TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("St").getDescr1()%></TD>
                                        <TD class=standardTxt><%=CurrUnit.getVariable("St").getDescription()%></TD>
                                </TR>
                        </TBODY>
                        </TABLE>
                        </form>
                </td>
</tr>
</table>
<br/>
<DIV id="alarms"></DIV>
```

Below are explanations of the individual sections of the code used to generate this graphic layout.

5.2.1 Import architectural objects

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
```

Function implemented: Architectural objects are imported.

5.2.2 Initialise architectural objects

```
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<jsp:useBean id="CurrUser" class="com.carel.supervisor.presentation.sdk.obj.CurrUser" scope="session"/>
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUser.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
```

Function implemented: Objects corresponding to the current device and user session are created. At the same time, device alarms are initialised and loaded in the last statement.

5.2.3 Initialise refresh and SET button

```
<script>
PVPK_ActiveRefresh(30);
PVPK_addButtons();
</script>
```

Function implemented: Initialise Javascript. The page is refreshed every 30 seconds.
SET buttons are displayed for making settings in the field.

5.2.4 Initialise user messages for settings out-of-range

```
<input type="hidden" id="s_minval" value="Minimum value: " />
<input type="hidden" id="s_maxval" value="Maximum value: " />
```

Function implemented: The two HIDDEN INPUTS shown above are required for managing messages to the user when the settings are out of range.
One typical example is when the SET POINT is entered below a certain threshold, the user is shown message such as “Minimum value allowed: XY”.

5.2.5 Line, status and device image information. Read-only variable link

```
<table border="0" width="100%" cellpadding="1" cellspacing="1">
<tr>
<td width="50%" valign="top"><img src="images/devices/mpxpro.jpg" border="1"></td>
<td width="50%" valign="top">
<table border="0" width="100%" cellpadding="1" cellspacing="1">
<tr>
<td><%=Sfera.assint(CurrUnit.getStatus(),"<img src='images/led/L0.gif'>;<img  src='images/led/L1.gif'>;<img
src='images/led/L2.gif'>;<img  src='images/led/L3.gif'>", "<img src='images/led/L0.gif'>")%>
</td>
<td><%=CurrUnit.getLine()%></td>
<td><%=CurrUnit.getVariable("s_cONOFF").getRefreshableAssint("UNIT ON;UNIT OFF","***")%></td>
</tr>
<tr><td><br/></td></tr>
<tr>
<td height="30" width="350" align="left" class="groupCategory_small" colspan="3"
onclick="PVP_goToVariables('readonly')">READ ONLY</td>
</tr>
</table>
</td>
</tr>
</table>
```

Function implemented: the image is shown using normal html tags.
The status LED is imported using the getStatus method in the CurrUnit object.
Information on the line is retrieved using the getLine method, again in the CurrUnit object.
The “READ-ONLY” button allows the user to open the standard page of device read-only variables. This link can be made using the Javascript function **PVP_goToVariables()**, called when clicking the button.

5.2.6 Table of read-only variables

```
<TABLE id="readonlyvartablecustom" class="table" cellSpacing=1 cellPadding=0 width="98%">
		<THEAD>
			<TR>
				<TH class=th height=18 width="18%" colSpan=2>Valore</TH>
				<TH class=th height=18 width="15%">Desc. breve</TH>
				<TH class=th height=18 width=*>Descrizione</TH>
			</TR>
		</THEAD>
		<TBODY>
			<TR class=Row1 height=21>
				<TD class=standardTxt width="9%" align=center colSpan=2>
				<%=CurrUnit.getVariable("s_PMP").getRefreshableAssint("<img
src='custom/dtlview_section/mpxprostep2/restart.png';<img
src='custom/dtlview_section/mpxprostep2/restart_red.png'>")%>
				</TD>
				<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMP").getDescr1()%></TD>
				<TD
class=standardTxt><%=CurrUnit.getVariable("s_PMP").getDescription()%></TD>
			</TR>
			<TR class=Row2 height=21>
				<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_SuctionProbe").getRefreshableValue()%></TD>
				<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_SuctionProbe").getMUnit()%></NOBR></TD>
				<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_SuctionProbe").getDescr1()%></TD>
				<TD
class=standardTxt><%=CurrUnit.getVariable("s_SuctionProbe").getDescription()%></TD>
			</TR>
			<TR class=Row1 height=21>
				<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_SetpointWork").getRefreshableValue()%></TD>
				<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_SetpointWork").getMUnit()%></NOBR></TD>
				<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_SetpointWork").getDescr1()%></TD>
				<TD
class=standardTxt><%=CurrUnit.getVariable("s_SetpointWork").getDescription()%></TD>
			</TR>
		</TBODY>
</TABLE>
```

Valore	Desc. breve	Descrizione
	s_PMP	Abilitazione posizionamento valvola manuale
27.3	s_SuctionProbe	Sonda di aspirazione
5.0°C/°F	StU	Setpoint di lavoro

Function implemented: All the variables shown above are refreshed automatically at a frequency equal to the initialisation value sent to the Javascript function (**PVPK_ActiveRefresh**).

5.2.7 Table containing buttons

```
<TABLE id="btttablecustom"  cellSpacing=1 cellPadding=0 width="98%">
                        <TR>
                                <TD width="100%">
                                        <TABLE class="table" cellspacing="1" cellpadding="0" width="98%">
                                        <THEAD>
                                                <TR>
                                                        <TH class=th height=18 >Pulsante</TH>
                                                        <TH class=th height=18 width=*>Descrizione</TH>
                                                </TR>
                                        </THEAD>
                                        <TBODY>
                                                <TR class="Row2" height="21">
                                                        <TD class="standardTxt" align="center"><nobr>

        <%=CurrUnit.getVariable("s_PMP").getSimpleButton(1,"1","images/button/relay_on_black.png","Abilitazione")
%>
                                                        </TD>
                                                        <TD class="standardTxt"
align="center"><nobr><%=CurrUnit.getVariable("s_PMP").getDescription()%></nobr></TD>
                                                </TR>
                                                <TR class="Row2" height="21">
                                                        <TD class="standardTxt" align="center"><nobr>

        <%=CurrUnit.getVariable("RESET").getSimpleButton(2,"1","images/button/alarmreset_on_black.png","Abilitazi
one")%>
                                                        </TD>
                                                        <TD class="standardTxt" align="center"><nobr>Reset
Allarmi</nobr></TD>
                                                </TR>
                                                <TR class="Row2" height="21">
                                                        <TD class="standardTxt" align="center"><nobr>

        <%=CurrUnit.getVariable("s_cLUX").getSimpleButton(3,"1","images/button/light_on_black.png","Abilitazione")
%>
                                                        </TD>
                                                        <TD class="standardTxt"
align="center"><nobr>Luce</nobr></TD>
                                                </TR>
                                        <TBODY>
                                        </TABLE>
                                </TD>
                                <TD width="50%"></TD>
                        </TR>
</TABLE>
```

Pulsante	Descrizione
	Abilitazione posizionamento valvola manuale
	Reset Allarmi
	Luce

Function implemented: three buttons are created for setting a variable directly. The following code is used in the example:
CurrUnit.getVariable("VAR_CODE").getSimpleButton(ID_UNIQUE,"VALUE","IMG_PATH","TOOLTIP"), where:

- VAR_CODE -> Unique code that identifies the variable;
- ID_UNIQUE -> If there are multiple buttons on the page (as in the example described above), this must be a progressive number;
- VALUE -> Value to send to the field when the button is pressed;
- IMG_PATH -> Image associated with the button;
- TOOLTIP -> Description displayed when the mouse is moved over the button.

5.2.8 Table of read/write variables

```
<form name="formSettableVars" id="formSettableVars">
                <TABLE id="readonlyvartablecustom" class="table" cellSpacing=1 cellPadding=0 width="98%">
                <THEAD>
                        <TR>
                                <TH class=th height=18 width="18%" colSpan=2>Valore</TH>
                                <TH class=th height=18 width="18%">Nuovo</TH>
                                <TH class=th height=18 width="15%">Desc. breve</TH>
                                <TH class=th height=18 width=*>Descrizione</TH>
                        </TR>
                </THEAD>
                <TBODY>
                        <TR class=Row1 height=21>
                                <TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_PMu").getRefreshableValue()%></TD>
                                <TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_PMu").getMUnit()%></NOBR></TD>
                                <TD class=standardTxt width="9%" align=center>
                                <%if(CurrUser.haveServicesRight()) {%>
                                <INPUT NAME="<%=CurrUnit.getVariable("s_PMu").getPostName()%>"
                                        id="<%=CurrUnit.getVariable("s_PMu").getPostName()%>"
class="lswtype"
                                        onkeydown="checkOnlyAnalog(this,event);"

onblur="sdk_checkMinMaxValue(this,9,16);checkOnlyAnalogOnBlur(this);" type="text" size="5"/>
                                <%}%>
                                </TD>
                                <TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMu").getDescr1()%></TD>
                                <TD
class=standardTxt><%=CurrUnit.getVariable("s_PMu").getDescription()%></TD>
                        </TR>
                        <TR class=Row2 height=21>
                                <TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("St").getRefreshableValue()%></TD>
                                <TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("St").getMUnit()%></NOBR></TD>
                                <TD class=standardTxt width="9%" align=center>
                                <INPUT NAME="<%=CurrUnit.getVariable("St").getPostName()%>"
                                        id="<%=CurrUnit.getVariable("St").getPostName()%>" class="lswtype"
                                        onkeydown="checkOnlyAnalog(this,event);"
                                        onblur="sdk_checkMinMaxValue(this,-
10,5);checkOnlyAnalogOnBlur(this);" type="text" size="5"/>
                                </TD>
                                <TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("St").getDescr1()%></TD>
                                <TD class=standardTxt><%=CurrUnit.getVariable("St").getDescription()%></TD>
                        </TR>
                </TBODY>
                </TABLE>
</form>
```

Valore	Nuovo	Desc. breve	Descrizione
14 step		s_PMu	Posizione valvola manuale
5.0 °C/°F		St	Setpoint regolazione

Function implemented: All text fields are validated by Javascript functions as follows:

- sdk_checkMinMaxValue(this,MIN,MAX) -> Validate that the value set for the parameter is within the range.
- checkOnlyAnalog() -> Check that only numeric values are entered.
- All the variables sent to the device must be entered on a FORM. The form must have a standard name: formSettableVars
- The value is set in the field by clicking the set button (

).

5.2.9 Alarms table

```
<DIV id="alarms"></DIV>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<table width="80%" align="left" border="0" class="table">
<thead>
	<tr>
		<th width="12%" class="th">Data ora</th>
		<th width="*" class="th">Descrizione</th>
		<th width="5%" class="th">Priorità</th>
		<th width="10%" class="th">Ack user</th>
		<th width="12%" class="th">Ack time</th>
	</tr>
</thead>

<tbody>
	<xsl:for-each select="response/alarms/alr">
		<xsl:variable name="classAlarm">statoAllarme<xsl:value-of select="prio"/>_w</xsl:variable>
		<tr class="{$classAlarm}">
			<td class="td"><xsl:value-of select="datetime"/></td>
			<td class="td"><xsl:value-of select="description"/></td>
			<td class="td"><xsl:value-of select="priority"/></td>
			<td class="td"><xsl:value-of select="ackuser"/></td>
			<td class="td"><xsl:value-of select="acktime"/></td>
		</tr>
	</xsl:for-each>
</tbody>
</table>
</xsl:template>
</xsl:stylesheet>
```

Data ora	Descrizione	Priorità	Ack user	Ack time
2010/08/03 10:42:26	*Allarme errore RTC*	*Media*		
2010/08/03 10:42:26	*Errore sonda S6*	*Alta*		

Function implemented: The alarms table is generated and automatically updated by the supervisor architecture. To import and use this, the user needs to create a DIV on the detail page as shown above**: <DIV id="alarms"></DIV>**.
In the same directory where the device detail customisation file resides, a special file needs to be created and renamed as: **stylesheet.xsl**
Users with knowledge of XSL can use this file to freely customise the layout of the alarms table.
The file is processed by the supervisor rendering engine, matching it to another XML file generated automatically mode.

This XML file is defined as follows:

```
<alarms>
	<alr>
		<datetime></datetime>
		<description></description>
		<priority></priority>
		<ackuser></ackuser>
		<acktime></acktime>
		<resetuser></resetuser>
		<resettime></resettime>
		<prio></prio>
	</alr>
</alarms>
```

5.2.10 File system for total customisation

The custom pages created by total customisation must be copied to the following path: **C:\Carel\PlantVisorPRO\engine\webapps\PlantVisorPRO\custom\dtlview\DEVICE_FOLDER** where DEVICE_FOLDER is replaced with the device ID code.

To display the alarms table, the **stylesheet.xsl** file must be placed in the same directory.

If adding a new custom device created using DEVICE CREATOR, when loading it onto the supervisor, the supervisor itself automatically creates the directory to contain the customisation data for the device. This directory will have the same name as the code that uniquely identifies the device.

6. STYLE SHEETS

Below are the documents relating to the style sheet used by the supervisor as a template for the graphic layout. Users can however create their own style sheets and import them into the custom settings, either integrating or even overwriting the standard CLASSES on the supervisor. The tables created previously to customise the device detail will be used as an example.

Valore	Nuovo	Desc. breve	Descrizione
14 step		s_PMu	Posizione valvola manuale
5.0 °C/°F		St	Setpoint regolazione

The CLASS of all **TABLE** blocks is "**table**". Column headers are described by a **THEAD** block. Nested in this block is the block describing the rows, **TR**, and then a **TH** block for each column. The **CLASS** of the latter is "**th**".
Once having defined the header, the table body containing the data (**TBODY**) needs to be defined. Each row containing variables will be described by a **TR** block. The **CLASS** of this **TR** may be "**Row1**" or "**Row2**". This allows alternate row colours to make the tables easier to read. Standard row height is 21 pixels, defined using the **height="21"** attribute in the **TR** block**.**
Each column is described by a **TD** block, where **CLASS** = "**standardTxt**".

```
<TABLE id="readonlyvartablecustom" class="table" cellSpacing=1 cellPadding=0 width="98%">
	<THEAD>
		<TR>
			<TH class=th height=18 width="18%" colSpan=2>Valore</TH>
			<TH class=th height=18 width="15%">Desc. breve</TH>
			<TH class=th height=18 width=*>Descrizione</TH>
		</TR>
	</THEAD>
	<TBODY>
		<TR class=Row1 height=21>
			<TD class=standardTxt width="9%" align=center colSpan=2>
				<%=CurrUnit.getVariable("s_PMP").getRefreshableAssint("<img
src='custom/dtlview_section/mpxprostep2/restart.png';<img
src='custom/dtlview_section/mpxprostep2/restart_red.png'>")%>
			</TD>
			<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_PMP").getDescr1()%></TD>
			<TD class=standardTxt><%=CurrUnit.getVariable("s_PMP").getDescription()%></TD>
		</TR>
		<TR class=Row2 height=21>
			<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_SuctionProbe").getRefreshableValue()%></TD>
			<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_SuctionProbe").getMUnit()%></NOBR></TD>
			<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_SuctionProbe").getDescr1()%></TD>
			<TD
class=standardTxt><%=CurrUnit.getVariable("s_SuctionProbe").getDescription()%></TD>
		</TR>
		<TR class=Row1 height=21>
			<TD class=standardTxt width="9%"
align=right><B><%=CurrUnit.getVariable("s_SetpointWork").getRefreshableValue()%></TD>
			<TD class=standardTxt width="9%"
align=left><NOBR><%=CurrUnit.getVariable("s_SetpointWork").getMUnit()%></NOBR></TD>
			<TD class=standardTxt width="15%"
align=middle><%=CurrUnit.getVariable("s_SetpointWork").getDescr1()%></TD>
			<TD
class=standardTxt><%=CurrUnit.getVariable("s_SetpointWork").getDescription()%></TD>
		</TR>
	</TBODY>
</TABLE>
```

To create/edit and import a style sheet, users need to add the following statement to the custom JSP page:
<LINK href="PATH_CSS_FILE" rel="stylesheet" type="text/css">
where PATH_CSS_FILE is the path of the style sheet being imported.

Standard CSS summary table:

TAG	CLASS
TABLE	table
THEAD	
TR	Row1,Row2
TH	th
TBODY	
TD	standardTxt

7. OBJECTS

SDK provides objects to interact with the following supervisor components:

- Plant;
- User;
- Devices;
- Variables;
- Alarms;

These objects are described below, together with the various methods available to retrieve information.

7.1 CurrNode

The CurrNode object represents the site and all connected devices.
This object can be used to retrieve references to the devices and consequently system variables.

7.1.1 Programming

The following statement is used to import this object into custom pages:

```
<jsp:useBean id="CurrNode" class="com.carel.supervisor.presentation.sdk.obj.CurrNode" scope="session" >
        <jsp:setProperty name="CurrNode" property="req" value="<%=request%>" />
</jsp:useBean>
```

CurrNode.getDevice(String deviceCode)
The above method requires the device code as the input, that is, the code made up of:

- Line number;
- Device serial address.

In the example shown below, the device in question is on line 1 with serial address 2, and its description is shown on the screen.

```
<%@ page language="java" %>
<jsp:useBean id="CurrNode" class="com.carel.supervisor.presentation.sdk.obj.CurrNode" scope="session" >
        <jsp:setProperty name="CurrNode" property="req" value="<%=request%>" />
</jsp:useBean>
<table border="1" width="20%" cellpadding="1" cellspacing="1">
<tr><td><%=CurrNode.getDevice("1.002").getDescription() %></td></tr>
</table>
```

CurrNode.getLenght()
The above method returns the number of devices on the site.

```
<%@ page language="java" %>
<jsp:useBean id="CurrNode" class="com.carel.supervisor.presentation.sdk.obj.CurrNode" scope="session" >
        <jsp:setProperty name="CurrNode" property="req" value="<%=request%>" />
</jsp:useBean>
<table border="1" width="20%" cellpadding="1" cellspacing="1">
<tr>
<td>Total Device Number:</td>
<td><%=CurrNode.getLenght()%></td>
</tr>
</table>
```

CurrNode.getCurrUnitAt(int index)
The above method requires an index as the input; this index represents the position of the device inside the array of devices in the supervisor memory.
This method can be used together with the method mentioned above, for example, to display all the devices present on the site on one global page.

```
<%@ page language="java" %>
<jsp:useBean id="CurrNode" class="com.carel.supervisor.presentation.sdk.obj.CurrNode" scope="session" >
        <jsp:setProperty name="CurrNode" property="req" value="<%=request%>" />
</jsp:useBean>
<table border="1" width="20%" cellpadding="1" cellspacing="1">
<%for (int i=0; i<CurrNode.getLenght(); i++) {%>
<tr>
<td><%=CurrNode.getCurrUnitAt(i)%></td>
</tr>
<% } %>
</table>
```

7.2 CurrUser

The CurrUser object represents the current user authenticated in the system. The object contains information relating to:

- User name: Name of the user account currently accessing the system (PVP 2.0);
- User profile: Name of the user profile currently accessing the system (PVP 2.0);
- User browser user: Name of the browser used by the current user (PVP 2.0);
- Screen resolution: Returns the current user's screen width and height connected (PVP 2.0);
- User parameter read/write authorisation.

Parameter read/write authorisation may be:

- Read-only: The user can only read the values of certain parameters, excluding manufacturer (OEM) parameters;
- Service write: The user can read and write service parameter, excluding manufacturer (OEM) parameters;
- Manufacturer write: The user can read and write both service and manufacturer (OEM) parameters.

7.2.1 Programming

The following statement is used to import this object into custom pages:

```
<jsp:useBean id="CurrUser" class="com.carel.supervisor.presentation.sdk.obj.CurrUser" scope="session"/>
<%
CurrUser.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
%>
```

The following methods are available:
CurrUser.getUserName()
CurrUser.getUserProfile()
CurrUser.getUserBrowser()
CurrUser.getScreenWidth()
CurrUser.getScreenHeight()

These methods are used respectively to retrieve the name, profile, browser, screen resolution width and screen resolution height for the current user.

```
<%@ page language="java" %>
<jsp:useBean id="CurrUser" class="com.carel.supervisor.presentation.sdk.obj.CurrUser" scope="session" >
        <jsp:setProperty name="CurrUser" property="req" value="<%=request%>" />
</jsp:useBean>
<table border="1" width="20%" cellpadding="1" cellspacing="1">
<tr><td><%=CurrUser.getUserName() %></td></tr>
<tr><td><%=CurrUser.getUserProfile() %></td></tr>
<tr><td><%=CurrUser.getUserBrowser() %></td></tr>
<tr><td><%=CurrUser.getScreenWidth() %></td></tr>
<tr><td><%=CurrUser.getScreenHeight() %></td></tr>
</table>
```

CurrUser.haveServicesRight()

This method returns TRUE if the current user has “service” write authorisation, otherwise FALSE.

```
<%@ page language="java" %>
<jsp:useBean id="CurrUser" class="com.carel.supervisor.presentation.sdk.obj.CurrUser" scope="session" >
        <jsp:setProperty name="CurrUser" property="req" value="<%=request%>" />
</jsp:useBean>
<table border="1" width="20%" cellpadding="1" cellspacing="1">
<tr><td><%=CurrUser.getUserName() %></td></tr>
<%if(CurrUser.haveServicesRight()) {%>
<tr><td>User with service right</td></tr>
<%} else {%>
<tr><td>User wihout service right</td></tr>
<%}%>
```

CurrUser.haveManufacturerRight()

This method returns TRUE if the current user has “manufacturer” write authorisation, otherwise FALSE.

```
<%@ page language="java" %>
<jsp:useBean id="CurrUser" class="com.carel.supervisor.presentation.sdk.obj.CurrUser" scope="session" >
        <jsp:setProperty name="CurrUser" property="req" value="<%=request%>" />
</jsp:useBean>
<table border="1" width="20%" cellpadding="1" cellspacing="1">
<tr><td><%=CurrUser.getUserName() %></td></tr>
<%if(CurrUser.haveManufacturerRight()) {%>
<tr><td>User with manufacturer right</td></tr>
<%} else {%>
<tr><td>User wihout manufacturer right</td></tr>
<%}%>
```

7.3 CurrUnit

The CurrUnit object represents the device that is currently selected in the memory. This object provides information relating to:

- Device status;
- The line the device is connected to;
- Device image (based on the model);
- Device description in current language;
- Total number of active alarms on the device.

7.3.1 Programming

The following statement is used to import this object into custom pages and load the related alarms:

```
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" /> <%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
```

The following methods are available.

CurrUnit.getStatus()
CurrUnit.getLine()
CurrUnit.getCode()
CurrUnit.getDescription()
CurrUnit.getAlarmNumber()
CurrUnit.getId()

EXAMPLE

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<table border="0" width="100%" cellpadding="1" cellspacing="1">
        <tr><td>Status: <%=CurrUnit.getStatus()%></td></tr>
        <tr><td>Line: <%=CurrUnit.getLine()%></td></tr>
        <tr><td>Code: <%=CurrUnit.getCode()%></td></tr>
        <tr><td>Description: <%=CurrUnit.getDescription()%></td></tr>
        <tr><td>Alarms count: <%=CurrUnit.getAlarmNumber()%></td></tr>
        <tr><td>Alarms count: <%=CurrUnit.getId()%></td></tr>
</table>
```

Device status may be:

- 0 -> OFFLINE (grey)
- 1 -> ONLINE (green)
- 2 -> ALARM (red)
- 3 -> DISABLED (blue)

CurrUnit.getVariable("CODE_VARIABLE")

This method is used to retrieve references to the device variable uniquely identified by CODE_VARIABLE.

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<table border="0" width="100%" cellpadding="1" cellspacing="1">
        <tr><td>Status: <%=CurrUnit.getStatus()%></td></tr>
        <tr><td>Line: <%=CurrUnit.getLine()%></td></tr>
        <tr><td>Description: <%=CurrUnit.getDescription()%></td></tr>
        <tr><td>Alarms count: <%=CurrUnit.getAlarmNumber()%></td></tr>
        <tr><td>Alarms count: <%=CurrUnit.getId()%></td></tr>
        <tr><td>Variable reference: <%=CurrUnit.getVariable("s_PMu").getDescription()%></td></tr>
</table>
```

7.4 CurrVar

The CurrVar object represents a variable on a specific device.

7.4.1 Programming

The object that refers to the variable can only be retrieved from the device.
CurrUnit. getVariable ("CODE_VAR").getValue();
CurrUnit. getVariable ("CODE_VAR").getDescription();
CurrUnit. getVariable ("CODE_VAR").getMUnit();
CurrUnit. getVariable ("CODE_VAR").getDescr1();
CurrUnit. getVariable ("CODE_VAR").getDescr2();
CurrUnit. getVariable ("CODE_VAR").getType();

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<table border="0" width="100%" cellpadding="1" cellspacing="1">
        <tr><td>Variable value: <%=CurrUnit.getVariable("s_PMu").getValue()%></td></tr>
        <tr><td>Variable description: <%=CurrUnit.getVariable("s_PMu").getDescription()%></td></tr>
        <tr><td>Variable unit: <%=CurrUnit.getVariable("s_PMu").getMUnit()%></td></tr>
        <tr><td>Variable short description: <%=CurrUnit.getVariable("s_PMu").getDescr1()%></td></tr>
        <tr><td>Variable long description: <%=CurrUnit.getVariable("s_PMu").getDescr2()%></td></tr>
        <tr><td>Variable type: <%=CurrUnit.getVariable("s_PMu").getType()%></td></tr>
</table>
```

CurrUnit. getVariable ("CODE_VAR"). getRefreshableValue()
This method displays the value of the variable, while the architecture automatically refreshes the value at the interval specified by the user.

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<table border="0" width="100%" cellpadding="1" cellspacing="1">
        <tr><td>Variable value: <%=CurrUnit.getVariable("s_PMu").getRefreshableValue()%></td></tr>
        <tr><td>Variable description: <%=CurrUnit.getVariable("s_PMu").getDescription()%></td></tr>
        <tr><td>Variable unit: <%=CurrUnit.getVariable("s_PMu").getMUnit()%></td></tr>
</table>
```

CurrUnit. getVariable ("CODE_VAR"). getFormattedValue(PATTERN)
This method displays the value of the variable formatted to a specific pattern set as the method input. The following patterns can be used:

INPUT	PATTERN	OUTPUT
123456.789	###,###.###	123,456.789
123456.789	###.##	123456.79
123.78	000000.000	000123.780
12345.67	$###,###.###	$12,345.67

```
<%@ page language="java"
	import="com.carel.supervisor.presentation.helper.ServletHelper"
	import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<script type="text/javascript" src="scripts/arch/arkustom.js"></script>
<table border="0" width="100%" cellpadding="1" cellspacing="1">
	<tr><td>Variable description: <%=CurrUnit.getVariable("s_PMu").getDescription()%></td></tr>
	<tr><td>Variable unit: <%=CurrUnit.getVariable("s_PMu").getMUnit()%></td></tr>
	<tr><td>Variable value: <%=CurrUnit.getVariable("s_PMu").getFormattedValue("000.00")%></td></tr>
</table>
```

CurrUnit. getVariable (“CODE_VAR”). getRefreshableFormattedValue(PATTERN)

As above, however with automatic refresh.

```
<%@ page language="java"
	import="com.carel.supervisor.presentation.helper.ServletHelper"
	import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<script type="text/javascript" src="scripts/arch/arkustom.js"></script>
<table border="0" width="100%" cellpadding="1" cellspacing="1">
	<tr><td>Variable description: <%=CurrUnit.getVariable("s_PMu").getDescription()%></td></tr>
	<tr><td>Variable unit: <%=CurrUnit.getVariable("s_PMu").getMUnit()%></td></tr>
	<tr><td>Variable value:
<%=CurrUnit.getVariable("s_PMu").getRefreshableFormattedValue("##.#")%></td></tr>
</table>
```

CurrUnit. getVariable (“CODE_VAR”). getRefreshableAssint(VALUE, DEFAULT)

This method is used when wanting to associate images to specific values and refresh these automatically. The method has an input VALUE that represents the set of images to be displayed based on the value of the variable, plus a DEFAULT used when the value of the variable is not included in the domain of images set as the input.

```
<%@ page language="java"
	import="com.carel.supervisor.presentation.helper.ServletHelper"
	import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<table border="0" width="100%" cellpadding="1" cellspacing="1">
	<tr><td>Variable value: <%=CurrUnit.getVariable("s_PMP").getRefreshableAssint("<img
src='custom/dtlview_section/mpxprostep2/restart.png';<img
src='custom/dtlview_section/mpxprostep2/restart_red.png'>")%></td></tr>
	<tr><td>Variable description: <%=CurrUnit.getVariable("s_PMu").getDescription()%></td></tr>
	<tr><td>Variable unit: <%=CurrUnit.getVariable("s_PMu").getMUnit()%></td></tr>
	<tr><td>Variable short description: <%=CurrUnit.getVariable("s_PMu").getDescr1()%></td></tr>
	<tr><td>Variable long description: <%=CurrUnit.getVariable("s_PMu").getDescr2()%></td></tr>
	<tr><td>Variable type: <%=CurrUnit.getVariable("s_PMu").getType()%></td></tr>
</table>
```

CurrUnit. getVariable ("CODE_VAR"). getPostName ()

This method can be used to freely create an HTML INPUT, such as a text or combo box, and give this element a NAME attribute based on the variable name understood by the architecture.

EXAMPLE (TEXT)

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<form name="formSettableVars" id="formSettableVars">
<table border="0" width="100%" cellpadding="1" cellspacing="1">
        <tr><td>Variable description: <%=CurrUnit.getVariable("s_PMu").getDescription()%></td></tr>
        <tr><td>Variable unit: <%=CurrUnit.getVariable("s_PMu").getMUnit()%></td></tr>
        <tr>
        <td width="9%">New value:
        <INPUT type="text" NAME="<%=CurrUnit.getVariable("s_PMu").getPostName()%>"
                id="<%=CurrUnit.getVariable("s_PMu").getPostName()%>" class="lswtype"
                onkeydown="checkOnlyAnalog(this,event);"
                onblur="sdk_checkMinMaxValue(this,9,16);checkOnlyAnalogOnBlur(this);" type="text" size="5"/>
        </td>
        </tr>
</table>
</form>
```

EXAMPLE (SELECT)

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<form name="formSettableVars" id="formSettableVars">
<table border="0" width="100%" cellpadding="1" cellspacing="1">
        <tr><td>Variable description: <%=CurrUnit.getVariable("s_PMu").getDescription()%></td></tr>
        <tr><td>Variable unit: <%=CurrUnit.getVariable("s_PMu").getMUnit()%></td></tr>
        <tr>
        <td width="9%">New value:
        <select NAME="<%=CurrUnit.getVariable("s_PMu").getPostName()%>"
                id="<%=CurrUnit.getVariable("s_PMu").getPostName()%>" class="lswtype">
                <option value="9">9</option>
                <option value="10">10</option>
                <option value="11">11</option>
                <option value="12">12</option>
        </select>
        </td>
        </tr>
</table>
</form>
```

CurrUnit. getVariable (“CODE_VAR”). getSimpleButton()
This method can be used to create an image and then use it as a button, for example to reset alarms or switch the lights on or off.

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<script type="text/javascript" src="scripts/arch/arkustom.js"></script>
<table border="0" width="100%" cellpadding="1" cellspacing="1">
        <tr><td>Variable description: <%=CurrUnit.getVariable("s_PMu").getDescription()%></td></tr>
        <tr><td>Variable unit: <%=CurrUnit.getVariable("s_PMu").getMUnit()%></td></tr>
        <tr>
        <td width="9%">New value:
        <%=CurrUnit.getVariable("s_PMP").getSimpleButton(1,"1","images/button/relay_on_black.png","Abilitazione")
%>
        </td>
        </tr>
</table>
```

7.5 CurrAlarm

The CurrAlarm object represents the alarm associated with a certain device.

7.5.1 Programming

The object that refers to the alarm can only be retrieved from the device.

CurrUnit. getAlarm (IDX).getDate()
CurrUnit. getAlarm (IDX).getDesc()
CurrUnit. getAlarm (IDX).getPriority()
CurrUnit. getAlarm (IDX).getAckTime()
CurrUnit. getAlarm (IDX).getAckUser()

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
CurrUnit.loadAlarms();
%>
<script type="text/javascript" src="scripts/arch/arkustom.js"></script>
<table border="0" width="100%" cellpadding="1" cellspacing="1">
        <tr><td>Device: <%=CurrUnit.getDescription()%></td></tr>
        <tr><td>Alarm desc: <%=CurrUnit.getAlarmAt(0).getDesc()%></td></tr>
        <tr><td>Priority: <%=CurrUnit.getAlarmAt(0).getPriority()%></td></tr>
</table>
```

7.6 Utility methods

The following method allows the user to get the PVPRO engine status.

Sfera.getEngineStatus()
The above method returns the PVPRO engine status.

```
<%@page language="java" %>
<%@page import=" com.carel.supervisor.presentation.sdk.util.Sfera " %>
<jsp:useBean id="CurrNode" class="com.carel.supervisor.presentation.sdk.obj.CurrNode" scope="session" >
      <jsp:setProperty name="CurrNode" property="req" value="<%=request%>" />
</jsp:useBean>
<table border="0" width="100%" height="100%" cellpadding="1" cellspacing="1">
<tr><td> Engine status: <%= Sfera.getEngineStatus()%></td></tr>
</table>
```

Engine status may be:

- 0 -> RUNNING
- 1 -> DEBUG
- 2 -> STOPPED
- 3 -> MUST RESTART

8. JAVASCRIPT

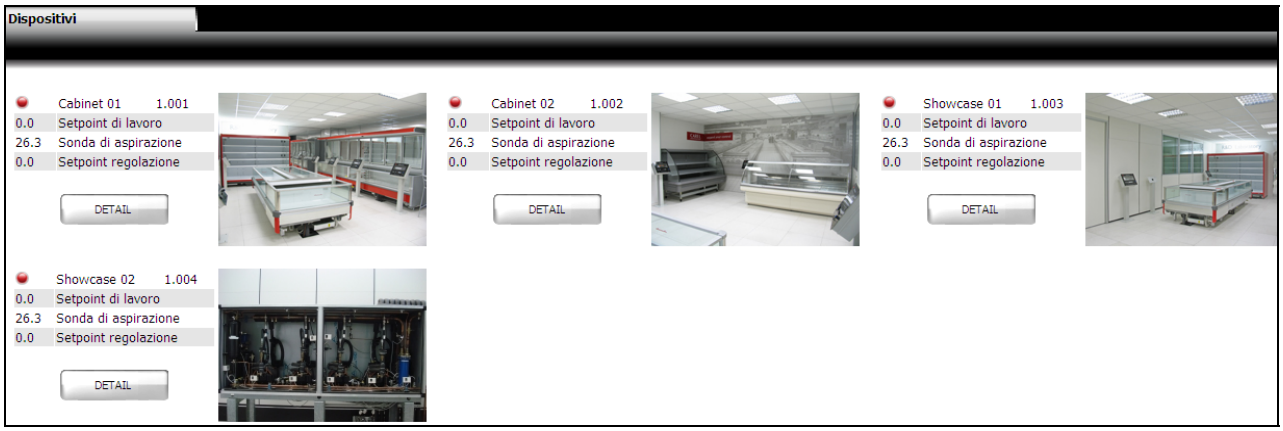
Javascript is fundamental for managing user events on the pages. The supervisor architecture provides functions used to standardise certain behaviour. For completely customised pages, the architecture imports the following default Javascript file:
C:\Carel\PlantVisorPRO\engine\webapps\PlantVisorPRO\scripts\custom\custom.js

This file contains the functions that can be called by the user.
To import one or more Javascript files into the JSP, use the following code:

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
%>
<script type="text/javascript" src="custom/dtlview/mpxprostep2/MYJS.js"></script>
```

Below is a description of the functions and examples on how to use them.

Example: overwriting the page under INSTALLATION -> DEVICES.
The new page will have the following layout:

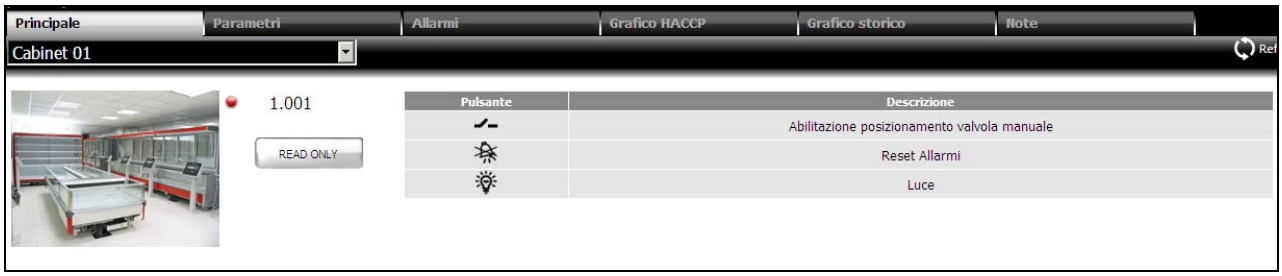


8.1 Link to device detail

The DETAIL button is used to access the details for the selected device. This is handled by the following function:
PVPK_goToDetail(ID_DEVICE);

```
<div onclick="PVPK_goToDetail(<%=CurrNode.getDevice("1.001").getId()%>);">DETAIL</div>
```

In the example described above, the ONCLICK event calls the Javascript function that opens the device detail. The method input is the device's unique identifier.

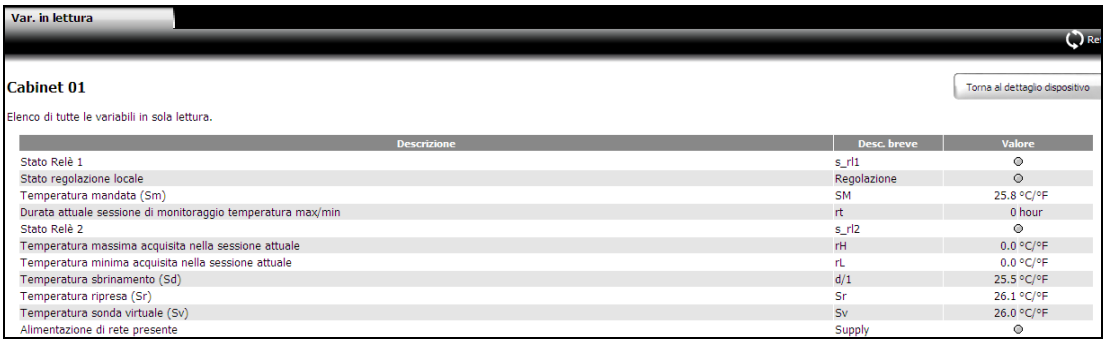


8.2 Link to list of read-only variables

The READ-ONLY button, as shown in the previous image, allows the user to open the page of device read-only variables. The following Javascript function is used:
PVP_goToVariables(DESCRIPTION);

```
<div onclick="PVP_goToVariables()">READ ONLY</div>
```

In the example described above, the ONCLICK event calls the Javascript function that opens the page of read-only variables.



8.3 Refresh device detail page

This function is provided to allow the device detail page to be refreshed automatically. The input is the refresh frequency in seconds.
PVPK_ActiveRefresh(SECONDS);

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
%>
<script type="text/javascript" src="scripts/arch/arkustom.js"></script>
<script>
PVPK_ActiveRefresh(30);
</script>
```

8.4 Display REFRESH and SET buttons

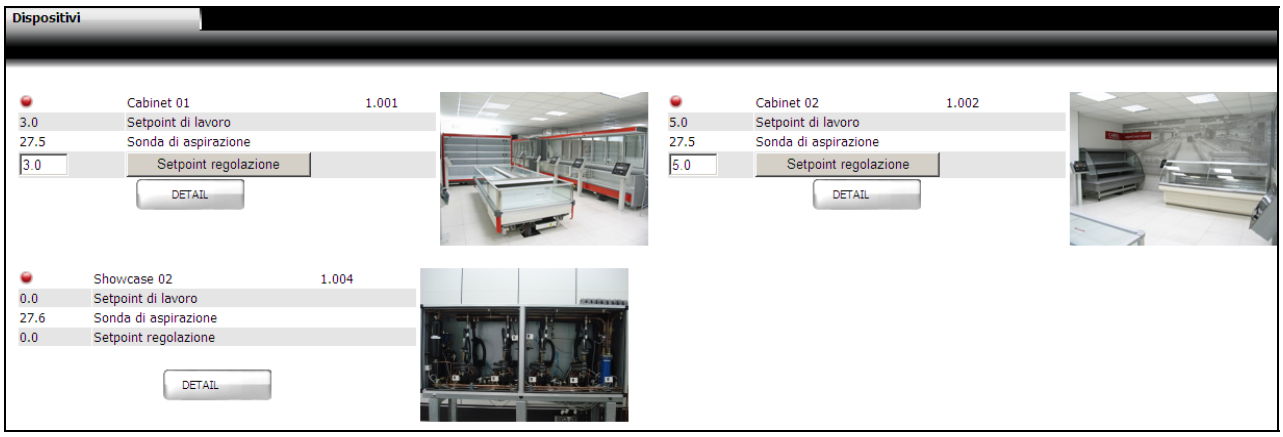
This function is provided to activate and display the refresh and set buttons respectively on the parameter detail page.
PVPK_addButtons();

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
%>
<jsp:useBean id="CurrUnit" class="com.carel.supervisor.presentation.sdk.obj.CurrUnit" scope="session" />
<%
CurrUnit.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
%>
<script type="text/javascript" src="scripts/arch/arkustom.js"></script>
<script>
PVPK_ActiveRefresh(30);
PVPK_AddButtons();
</script>
```

For the complete code, see the paragraph on PARTIAL and COMPLETE customisation of the device detail.

8.5 Set point

The user can set the field device control set point from a page other than the device detail page. Once again using the example of the customised device page:



HTML code can be added to create a text field and send the new control set point value to the field.

```
<form>
<table border="0" class="table" cellpadding="5" cellspacing="5" width="100%">
	<tr>
		<td width="50%">
			<table border="0" cellpadding="1" cellspacing="1" width="100%">
				<tr>

					<td valign="top">

						<table border="0" cellpadding="1" cellspacing="1" width="100%">
							<tr>
<td class="standardTxt"><%=Sfera.assint(CurrNode.getDevice("1.001").getStatus(),"<img src='images/led/L0.gif';<img src='images/led/L1.gif';<img  src='images/led/L2.gif';<img  src='images/led/L3.gif'>", "<img src='images/led/L0.gif'>")%></td>
<td class="standardTxt"><%=CurrNode.getDevice("1.001").getDescription() %>	</td>
<td class="standardTxt"><%=CurrNode.getDevice("1.001").getLine()%>	</td>
</tr>
<tr class="Row2">
<td class="standardTxt"><%=CurrNode.getDevice("1.001").getVariable("s_SetpointWork").getValue()%>	</td>
<td class="standardTxt" colspan="2"><%=CurrNode.getDevice("1.001").getVariable("s_SetpointWork").getDescription()%>	</td>
</tr>
<tr class="Row1">
<td class="standardTxt"><%=CurrNode.getDevice("1.001").getVariable("s_SuctionProbe").getValue()%>	</td>
<td class="standardTxt" colspan="2"><%=CurrNode.getDevice("1.001").getVariable("s_SuctionProbe").getDescription()%>	</td>
</tr>
<tr class="Row2">
<td class="standardTxt">
<input type="text" size="2" name="<%=CurrNode.getDevice("1.001").getVariable("St").getPostName()%>" id="<%=CurrNode.getDevice("1.001").getVariable("St").getPostName()%>" value="<%=CurrNode.getDevice("1.001").getVariable("St").getValue()%>"/>
</td>
<td class="standardTxt" colspan="2"><button onclick="PVP_setData(this);"><%=CurrNode.getDevice("1.001").getVariable("St").getDescription()%></button></td>
</tr>
```

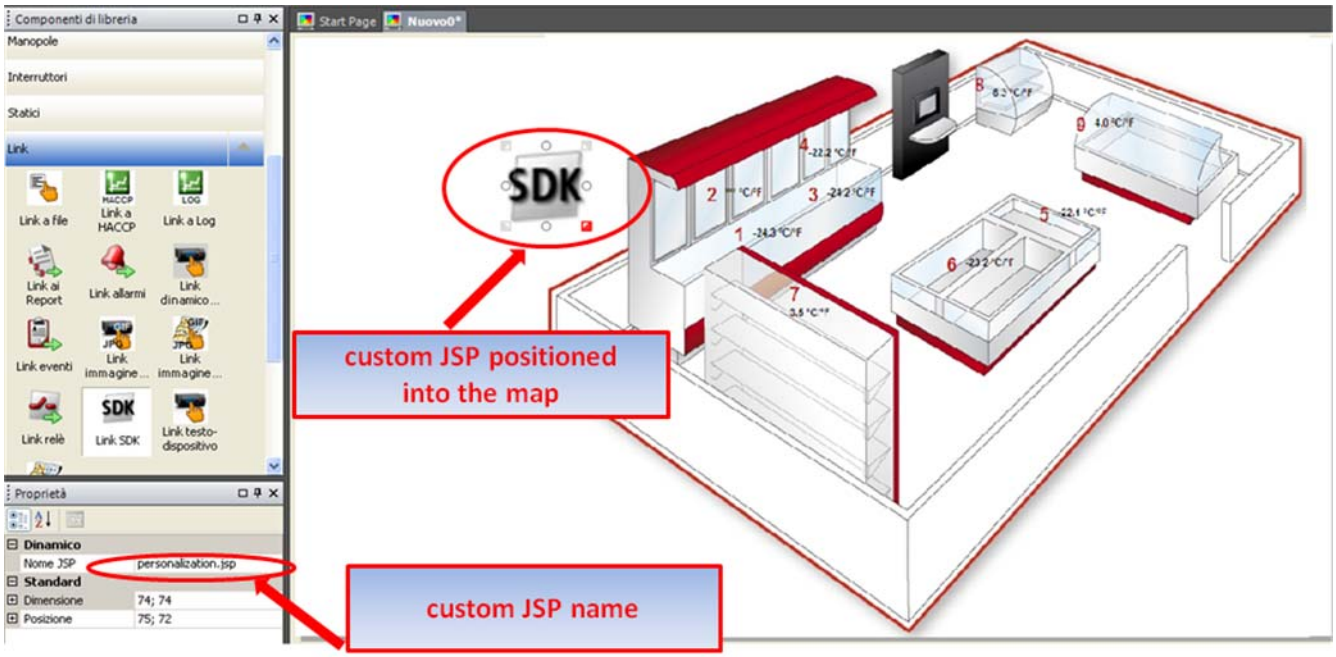
```
                                                    <tr>
                                                        <td height="30" width="350" align="right"
class="groupCategory_small" colspan="2" >
                                                        <div
onclick="PVPK_goToDetail(<%=CurrNode.getDevice("1.001").getId()%>);">DETAIL</div>
                                                        </td>
                                                    </tr>
                                                </table>
                                            </td>
                                            <td>
                                                <img src="custom/deviceview/1.Jpeg" />
                                            </td>
                                        </tr>

                                    </table>
```

Clicking the SET button calls the Javascript function **PVP_setData(this);** this function sends (POST method) the data on the FORM containing the button pressed.

9. INTEGRATION WITH LAYOUT EDITOR

Layout Editor version 3.1.0 and higher can be used to introduce SDK customisations onto the layouts.



As a result, sections can be added that the user can then customise as desired using SDK.
In the example described above, the custom JSP must have the following name: "**personalization.jsp**" as defined in the field called: "**JSP name**".
As per the standard procedure for creating layouts, the layout must be saved in exported in Layout Editor, then the zip file imported into the supervisor. The layout will be complete when the JSP file is created for implementing the SDK.
The file called "**personalization.jsp**" must be located in the following directory:
C:\Carel\PlantVisorPRO\engine\webapps\PlantVisorPRO\app\mstrmaps

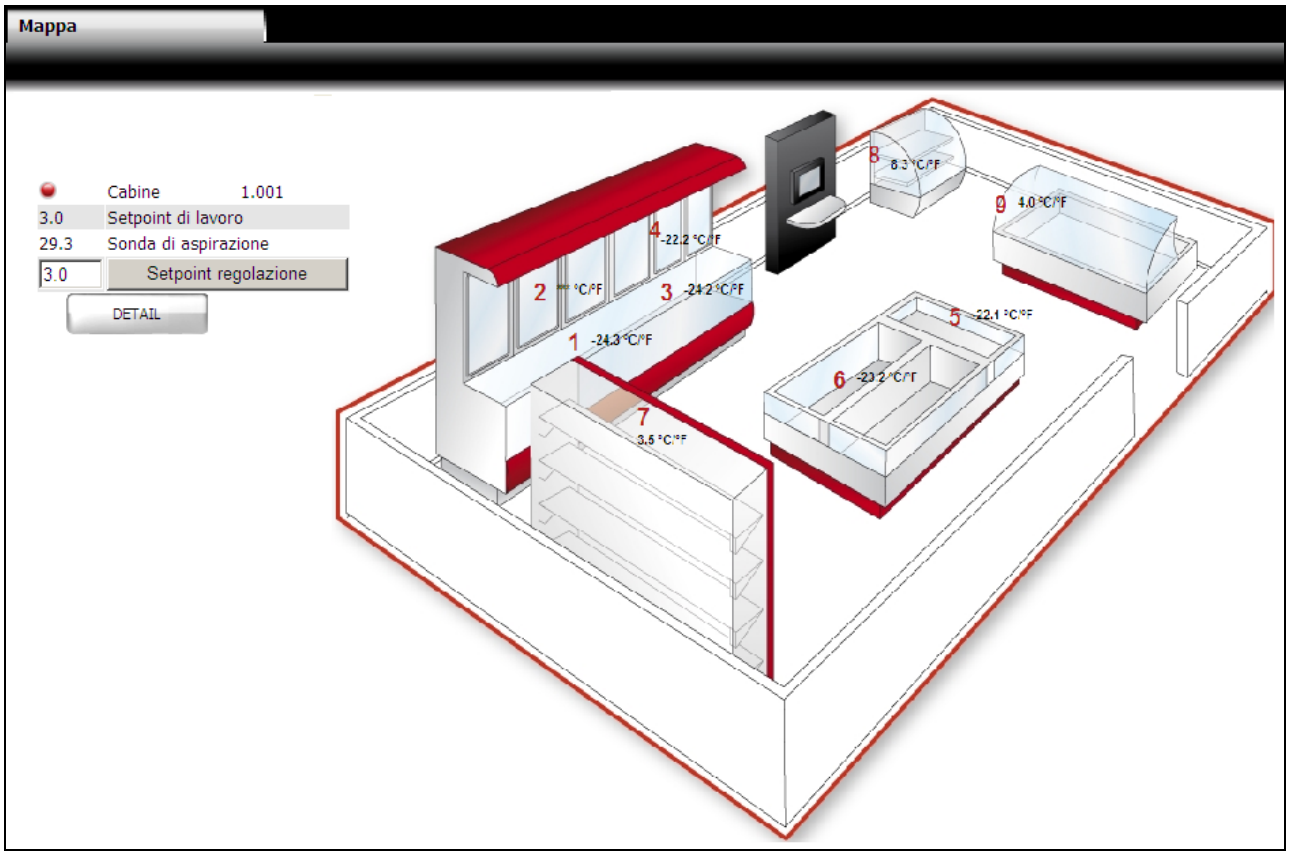
The "**personalization.jsp**" file can be defined as follows:

```
<%@ page language="java"
        import="com.carel.supervisor.presentation.helper.ServletHelper"
        import="com.carel.supervisor.presentation.sdk.util.Sfera"
%>
<jsp:useBean id="CurrNode" class="com.carel.supervisor.presentation.sdk.obj.CurrNode" scope="session" />

<script type="text/javascript" src="scripts/arch/arkustom.js"></script>
<script type="text/javascript" src="scripts/custom/custom.js"></script>
<%
 CurrNode.setCurrentSession(ServletHelper.retrieveSession(request.getRequestedSessionId(),request));
%>
<form>
<table border="0" cellpadding="1" cellspacing="1" width="100%">
        <tr>
                <td class="standardTxt"><%=Sfera.assint(CurrNode.getDevice("1.001").getStatus(),"<img
src='images/led/L0.gif';<img  src='images/led/L1.gif';<img  src='images/led/L2.gif';<img  src='images/led/L3.gif'>",
"<img src='images/led/L0.gif'>")%></td>
                <td class="standardTxt"><%=CurrNode.getDevice("1.001").getDescription() %> </td>
                <td class="standardTxt"><%=CurrNode.getDevice("1.001").getLine()%>          </td>
        </tr>
        <tr class="Row2">
                <td
class="standardTxt"><%=CurrNode.getDevice("1.001").getVariable("s_SetpointWork").getValue()%>        </td>
```

```
                    <td class="standardTxt"
colspan="2"><%=CurrNode.getDevice("1.001").getVariable("s_SetpointWork").getDescription()%>            </td>
          </tr>
          <tr class="Row1">
                    <td
class="standardTxt"><%=CurrNode.getDevice("1.001").getVariable("s_SuctionProbe").getValue()%>          </td>
                    <td class="standardTxt"
colspan="2"><%=CurrNode.getDevice("1.001").getVariable("s_SuctionProbe").getDescription()%>            </td>
          </tr>
          <tr class="Row2">
                    <td class="standardTxt">
                              <input type="text" size="2"
name="<%=CurrNode.getDevice("1.001").getVariable("St").getPostName()%>"
id="<%=CurrNode.getDevice("1.001").getVariable("St").getPostName()%>"
value="<%=CurrNode.getDevice("1.001").getVariable("St").getValue()%>"/>
                    </td>
                    <td class="standardTxt" colspan="2"><button
onclick="PVP_setData(this);"><%=CurrNode.getDevice("1.001").getVariable("St").getDescription()%></button></td>
          </tr>
          <tr>
                    <td height="30" width="350" align="right" class="groupCategory_small" colspan="2" >
                    <div onclick="PVPK_goToDetail(<%=CurrNode.getDevice("1.001").getId()%>);">DETAIL</div>
                    </td>
          </tr>
</table>
</form>
```

The resulting custom layout will be:



10. CUSTOMISABLE FUNCTIONS

The following table summarises the functions and the corresponding directories / JSP pages that can be customised using SDK.

MENU	FUNCTION	DIRECTORY	JSP
 **INSTALLATION**	**DEVICES**	**deviceview**	**SubTab1.jsp**
 **INSTALLATION**	**LAYOUT**	**mstrmaps**	**SubTab1.JSP**
 **INSTALLATION**	**MAIN DEVICE DETAIL**	**dtlview\DEVICE_CODE**	**SubTab1.jsp**
 **INSTALLATION**	**DEVICE PARAMETERS DETAIL**	**dtlview\DEVICE_CODE**	**SubTab2.jsp**
 **INSTALLATION**	**PARTIAL – MAIN DEVICE DETAIL**	**dtlview_section \DEVICE_CODE**	**secA.jsp secB.jsp**

Where **DEVICE_CODE** is the unique code that identifies the device.

11. LIST OF DEVICES

The following table lists the devices and corresponding PVP code.
For custom applications, with models developed in device creator, the code used for the custom device is the unique code entered in the CODE field of the device general information window.

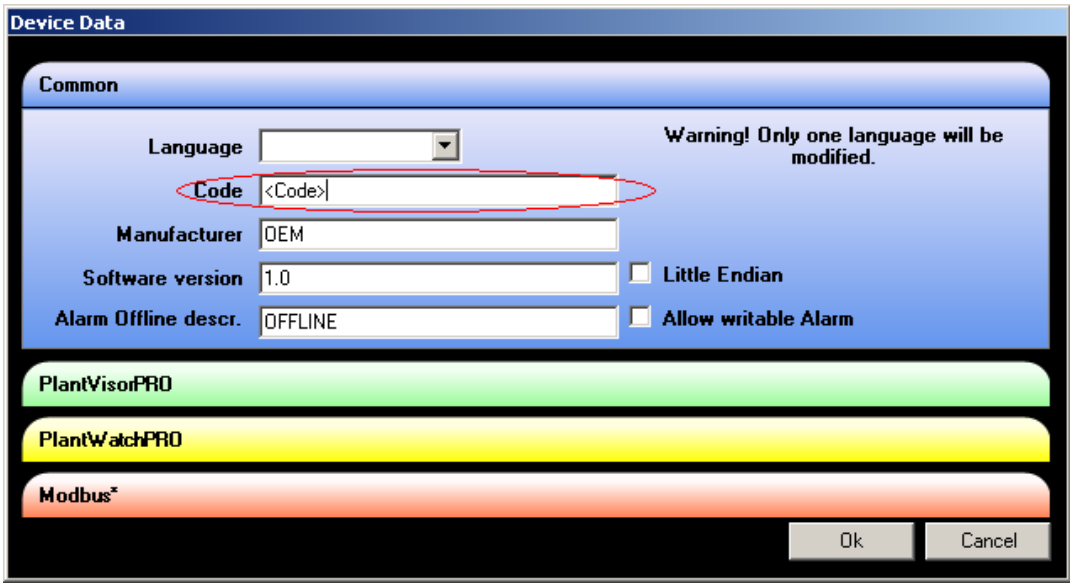


Table of devices/directories:

DEVICE DESCRIPTION	CODE /DIRECTORY
Access Point ZigBee (Carel)	APZigBee_Carel
Access Point rTM System (Modbus)	AP-ZED_modbus
Access Point rTM System SE (Modbus)	AP-ZED_modbus_SE
Air Handling Unit	FLSTDmAHUE
Anti-sweat controller	ACC
Blast Chiller	Blast Chiller FLSTDmFZCE
Blocksystem	BLOCKSYSTEM
Seasoning room (EPSTDICS00)	pCOb Cella stagion. EPSTDICS00
Ducati energy smart plus	DUCATI
E-drofan 1.8 version	edrofan
E-drofan 2.0 version	edrofan_20
EVD evolution	evdevo
EVD evolution twin	evdevotwin
EVD300 Expansion valve driver	EVD300
EVD400 - Retail Universal	EVD400
Easy AP	EasyAP
Easy Read Temperature	EasyReadT
Easy Read Temperature & Humidity	EasyReadTH
Easy Set Temperature	EasySetT
Easy Set Temperature and Humidity	EasySetTH
Energy Meter - Gavazzi CPT DIN - WM14	pwr_an
Energy Meter - IME Nemo96HD	Nemo96HD
Energy Meter - IME Nemod4	Nemod4

Energy2	Energy2
Fan speed control	C590
Generic pCO2	GENERIC
HP	FLSTDmHPGE
I/O Module	I/O Module
IR 33 - C	IR 33 - C
IR 33 - F	IR 33 - F
IR 33 - M	IR 33 - M
IR 33 - S	IR 33 - S
IR 33 - Y	IR 33 - Y
IR 33 DIN - C	IR 33 DIN - C
IR 33 DIN - F	IR 33 DIN - F
IR 33 DIN - H	IR 33 - H
IR 33 DIN - M	IR 33 DIN - M
IR 33 DIN - S	IR 33 DIN - S
IR 33 DIN - Y	IR 33 DIN - Y
IR 33 Universal 1 output	IR33U1
IR 33 Universal 2 outputs	IR33U2
IR 33 Universal 4 outputs	IR33U4
IR 33 Universal Temperatures 1 output	IR33UT1
IR 33 Universal Temperatures 2 outputs	IR33UT2
IR 33 Universal Temperatures 4 outputs	IR33UT4
IR Mpx	Mpx
IR32 M S Y X C for refrigeration	IR32 - M S Y X C
IR32 V W Z Universal 1 humidity probe	IR32 VWZ Univ. humidity probe
IR32 V W Z Universal 1 pressure probe	IR32 VWZ Univ. pressure probe
IR32 V W Z Universal 2 NTC probes	IR32 VWZ Univ. 2 NTC probes
IR33C Modbus	IR33_MODBUS
IRS2C0LN00	IRS2C0LN00
Internal IO	Internal IO
MC MULTI ZONE	MC_MULTI_ZONE
MPXPRO Step 2	mpxprostep2
MasterCase	MasterCase
MasterCase2	MasterCase2
MasterCase3	mastercase3
MasterCella (MD33)	mcella_v1
MasterCella (MTC)	mcella
MasterCella Split	MasterCella Split
MicroChiller3	MicroChiller3
Modular Chiller-HP screw compr. (MSCA) - Unit type 0	pCOb M.Ch-HP screw (MSCA) ty 0
Modular Chiller-HP screw compr. (MSCA) - Unit type 1	pCOb M.Ch-HP screw (MSCA) ty 1
Modular Chiller-HP screw compr. (MSCA) - Unit type 2	pCOb M.Ch-HP screw (MSCA) ty 2
Modular Chiller-HP screw compr. (MSCA) - Unit type 3	pCOb M.Ch-HP screw (MSCA) ty 3
Modular Chiller-HP screw compr. (MSCA) - Unit type 4	pCOb M.Ch-HP screw (MSCA) ty 4
Modular Chiller-HP screw compr. (MSCA) - Unit type 5	pCOb M.Ch-HP screw (MSCA) ty 5
Modular Chiller-HP screw compr. on pCO1/2/c (*MSDE) - type 0	pCO M.Ch-HP screw (*MSDE) ty 0

Modular Chiller-HP screw compr. on pCO1/2/c (*MSDE) - type 1	pCO M.Ch-HP screw (*MSDE) ty 1
Modular Chiller-HP screw compr. on pCO1/2/c (*MSDE) - type 2	pCO M.Ch-HP screw (*MSDE) ty 2
Modular Chiller-HP screw compr. on pCO1/2/c (*MSDE) - type 3	pCO M.Ch-HP screw (*MSDE) ty 3
Modular Chiller-HP screw compr. on pCO1/2/c (*MSDE) - type 4	pCO M.Ch-HP screw (*MSDE) ty 4
Modular Chiller-HP screw compr. on pCO1/2/c (*MSDE) - type 5	pCO M.Ch-HP screw (*MSDE) ty 5
Modular chiller-HP 1-4 comp. (MC0E) - type 0 - master	mod_ch_14_0m
Modular chiller-HP 1-4 comp. (MC0E) - type 0 - slave	mod_ch_14_0s
Modular chiller-HP 1-4 comp. (MC0E) - type 1 - master	mod_ch_14_1m
Modular chiller-HP 1-4 comp. (MC0E) - type 1 - slave	mod_ch_14_1s
Modular chiller-HP 1-4 comp. (MC0E) - type 2	mod_ch_14_2
Modular chiller-HP 1-4 comp. (MC0E) - type 3	mod_ch_14_3
Modular chiller-HP 1-4 comp. (MC0E) - type 4	mod_ch_14_4
Modular chiller-HP 1-4 comp. (MC0E) - type 5	mod_ch_14_5
Modular chiller-HP 1-8 comp. (EPSTDMMCHA)	pCOb M. Ch-HP 8Comp EPSTDMMCHA
Modular chiller-HP with driver (FLSTDMMCDE)	pCO M. Ch-HP driver FLSTDMMCDE
NXL Inverter	NXL
OEM kit KUE	CP
PC-Gate	PC-Gate
PJ-Easy	pjeasy
PJ-Easy Split	pjeasysplit
Plug-In	plugin
Power compact - C v.2	Power compact - C v.2
Power compact - F v.2	Power compact - F v.2
Power compact - H v.2	Power compact - H v.2
Power compact - M v.2	Power compact - M v.2
Power compact - S v.2	Power compact - S v.2
Power compact - Y v.2	Power compact - Y v.2
PowerSplit	PowerSplit
Rack controller v1.8 (FLSTDmFC0A)	pCO Rack v1.8 FLSTDmFC0A
Rack controller v2.1 or higher (FLSTDmFC0A)	pCO Rack v2.1 FLSTDmFC0A
Roof-top 1/2 compressors on pCO1 (FLSTDMRT0E)	pCO1 Roof-top FLSTDMRT0E
Roof-top 1/2 compressors on pCO2 (FLSTDMRT0E)	pCO2 Roof-top FLSTDMRT0E
Roof-top 1/2 compressors on pCOc (FLSTDMRT0E)	pCOc Roof-top FLSTDMRT0E
Roof-top 1/2 compressors on pCOxs (FLSTDMRT0E)	pCOxs Roof-top FLSTDMRT0E
Serial Probe DPW (Carel)	SP-DPW_Carel
Service E3 HS10	Service E3 HS10
Service E3 HS5	Service E3 HS5
Shelter on pCO1-2 medium (*MSHE)	pCO1-2 medium Shelter (*MSHE)
Shelter on pCO1-2 small (*MSHE)	pCO1-2 small Shelter (*MSHE)
Shelter on pCOxs (*MSHE)	pCOxs Shelter (*MSHE)
Standard Air-conditioners on pCO1/2 (FLSTDMCZ0E)	pCO1-2 Std Aircond. FLSTDMCZ0E
Standard Air-conditioners on pCOxs (FLSTDMCZ0E)	pCOxs Std Aircond. FLSTDMCZ0E
Standard Air handling unit (FLSTDMAHUA)	FLSTDMAHUA
Standard rooftop 4 comp. 2 circ. with EVD400 on pCO* Large. Version: 2.0	FLSTDmRT0E

Standard rooftop 4 comp. 2 circ. with EVD400 on pCO* Medium. Version: 2.0	FLSTDmRT0E_pCO3M
Standard rooftop 4 comp. 2 circ. with EVD400 on pCO* XSmall. Version: 2.0	FLSTDmRT0E_pCO3XS
Wireless Probe ZigBee (Carel)	WPZigBee_Carel
Wireless probe	Wireless probe
Wireless router ver.230/Bridge - rTM system SE (Modbus)	RouterW_modbus_SE
gaSteam	Gasteam
heaterSteam	HeaterSteam
humiFog for AHU/duct	Humifog 1.1
humiFog for Ambient	Humifog 0.2
humiSteam	HumiSteam
pCO Rack controller 1/3 compr. (EPSTD*FC2A)	pCOb Rack 3 comp. EPSTD*FC2A
pCO Roof-top 1/2 compressors (EPSTDMRT0A)	pCOb Roof-top EPSTDMRT0A
pCO Universal stage controller (EPSTDIIU0A)	pCOb Univ.stage c (EPSTDIIU0A)
pCO air conditioning unit (EPSTDECZUB)	pCOb air cond. EPSTDECZUB
pCO air handling unit (FLSTDMAHUA) (>= version 1.8)	pCO AHU FLSTDMAHUA >= ver1.8
pRack v1.0 (FLSTDmRC0E)	pRack PR100 v1.0 FLSTDmRC0E
pRack v1.0 (FLSTDmRC0E) line 1	pRack PR100 v1.0 FLSTDmRC0E line 1
pRack v1.0 (FLSTDmRC0E) line 2	pRack PR100 v1.0 FLSTDmRC0E line 2
rTM Wireless sensor BP version SE (Modbus)	WP-SPM_modbus_SE
rTM Wireless sensor BP version (Modbus)	WP-SPM_modbus
rTM Wireless sensor EP version (Modbus)	WP-ZED_modbus
rTM Wireless sensor EP-EP1 version SE (Modbus)	WP-ZED_modbus_SE
uAC	uAC
uChiller	uChiller
uChiller compact	uChiller compact
uChiller2 - digital outputs 0	uChiller2 - digital outputs 0
uChiller2 - digital outputs 1	uChiller2 - digital outputs 1
uChiller2 - digital outputs 2	uChiller2 - digital outputs 2
uChiller2 - digital outputs 3	uChiller2 - digital outputs 3
uChiller2 - digital outputs 4	uChiller2 - digital outputs 4
uChiller2 - digital outputs 5	uChiller2 - digital outputs 5
uRack (MRK)	uRack (MRK)

CAREL

CAREL INDUSTRIES HeadQuarters
Via dell'Industria, 11 - 35020 Brugine - Padova (Italy)
Tel. (+39) 049.9716611 - Fax (+39) 049.9716600
e-mail: carel@carel.com - www.carel.com

Agenzia / *Agency*:

"SDK PlantVisorPRO" cod. +030220766 Rel. 2.1 – 21/11/11