Supervisor Device Template

for Saia PCD® Supervisor

0 Contents

Page

1. Document version

Date	Version	From	Description
2020-01-07	ENG01	M. Egner	Translated from German version GER01
2020-05-14	ENG02	M. Egner	New hints in chapter 5.2
2020-04-28	ENG02	M. Egner	Anpassungen und Support für N4.8 & N4.9 Neue Hinweise in Kapitel 13 hinzugefügt

2. Description

The device template is an HTML template that represents a home page with a freely configurable menu containing all possible views as well as PX pages, HTML pages from the Supervisor/Niagara. In addition, alarms can also be displayed according to priority

The menu supports nesting on up to 4 levels. The approach means that a separate navigation menu can be created for each user within the Supervisor.

Alarm lists have already been defined as examples for this

In addition, several users have already been created as examples, each with their own roles, rights and languages, as well as their own individual views of the menu structure.

3. System requirements

List of the required software versions:

- Starting from Template Version 1.0.21 | Supervisor 2.1 & 3.0 | Niagara 4.8 & 4.9
- Browser:
  - Google Chrome Version 79 or newer
  - Firefox Version 71.0 or newer
  - EDGE 88 or newer
- Windows 10

4. Installation

There are two ways to use the template. These are explained below.

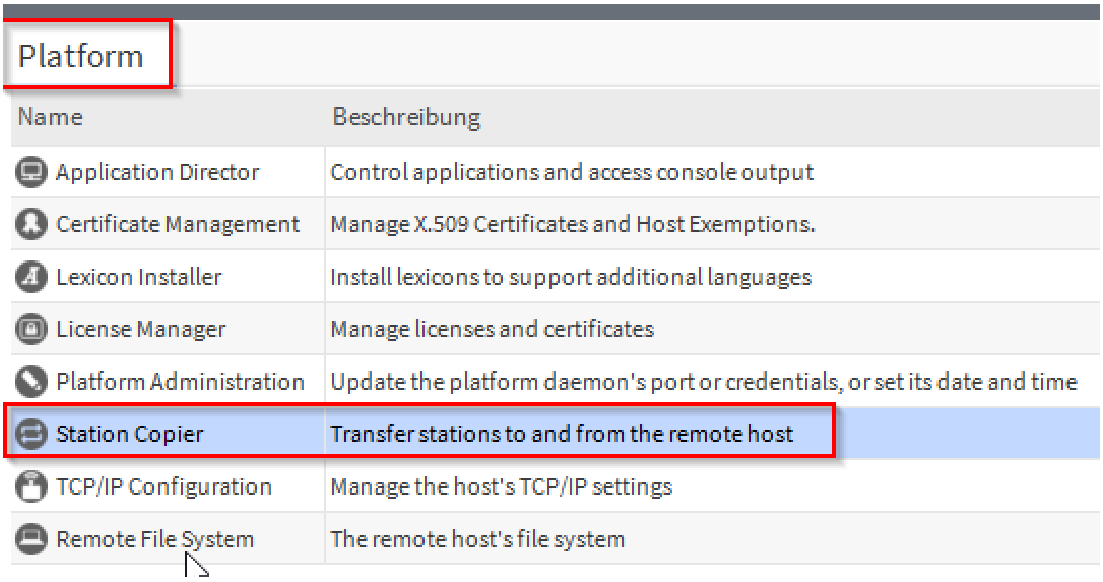
4.1 Creating a new project with the device template

The template is a ready-made project and can therefore easily be used directly.

To be able to use it, copy the “sbcDeviceTemplate21vX_X_X” folder from the zip file to the following directory (User Home): “C:\Users\MyUserName\Niagara4.8\SBC\stations” for Supervisor 2.1 or “C:\Users\MyUserName\Niagara4.9\SBC\stations” for Supervisor 3.0.

Start the Supervisor and go to the platform

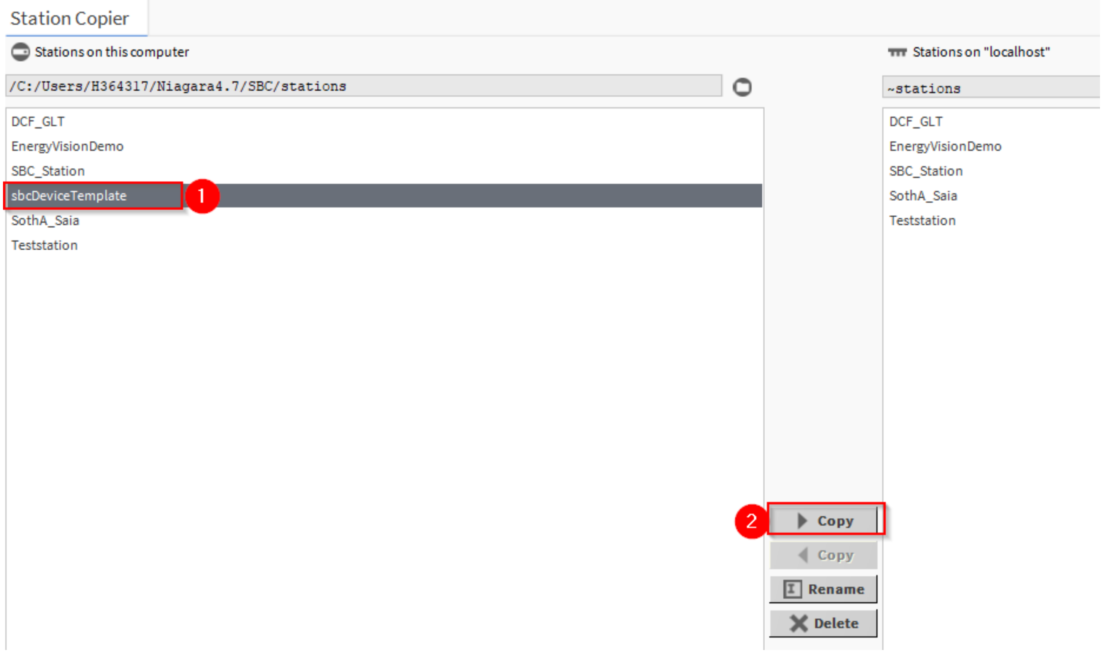
Go to the “Station Copier”:



Platform

Name	Beschreibung
Application Director	Control applications and access console output
Certificate Management	Manage X.509 Certificates and Host Exemptions.
Lexicon Installer	Install lexicons to support additional languages
License Manager	Manage licenses and certificates
Platform Administration	Update the platform daemon's port or credentials, or set its date and time
Station Copier	Transfer stations to and from the remote host
TCP/IP Configuration	Manage the host's TCP/IP settings
Remote File System	The remote host's file system

In the “Station Copier” select the project and copy it to the platform.



Station Copier

Stations on this computer

/C:/Users/H364317/Niagara4.7/SBC/stations

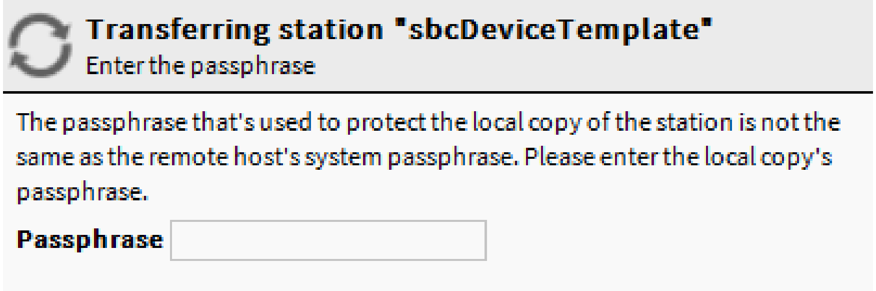
DCF_GLT
EnergyVisionDemo
SBC_Station
sbcDeviceTemplate
SothA_Saia
Teststation

Stations on "localhost"

~stations

DCF_GLT
EnergyVisionDemo
SBC_Station
SothA_Saia
Teststation

First enter the pass phrase “SaiaPCD@1978” and then click Next.



Transferring station "sbcDeviceTemplate"

Enter the passphrase

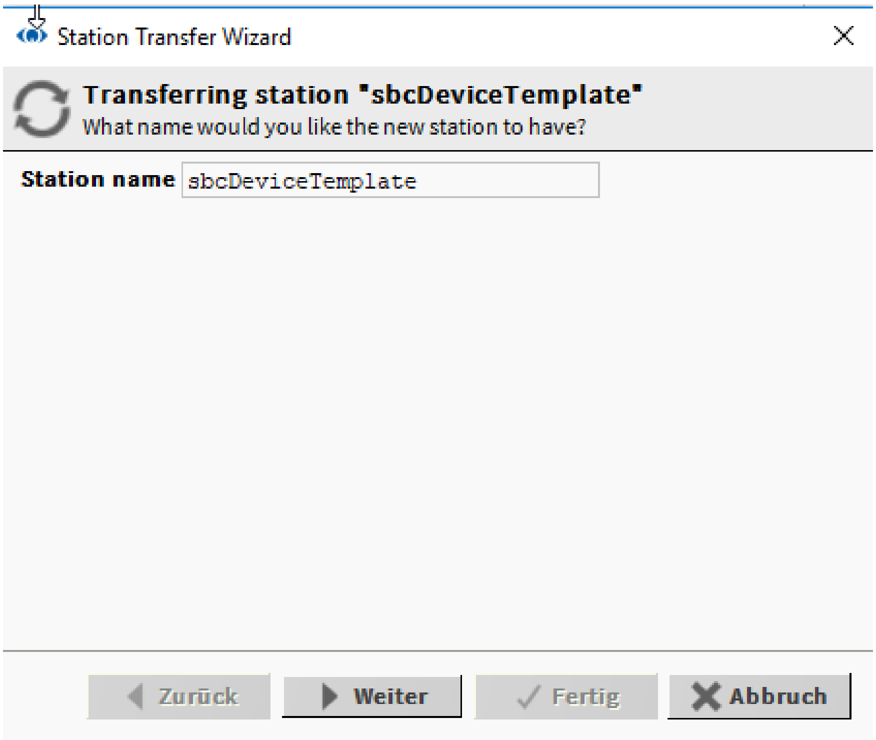
The passphrase that's used to protect the local copy of the station is not the same as the remote host's system passphrase. Please enter the local copy's passphrase.

Passphrase

You can also rename the station directly in the next window.

The name can be chosen freely.

Then click Next



Station Transfer Wizard

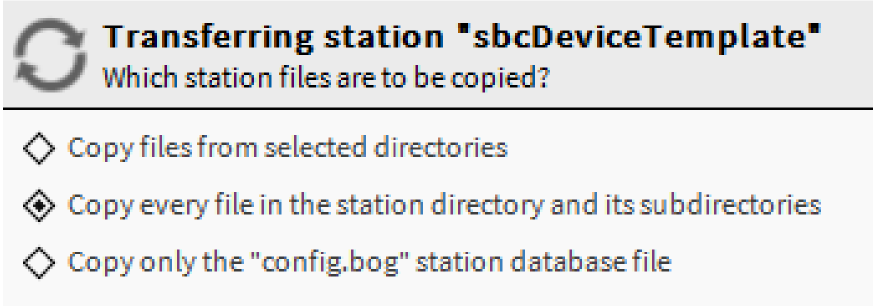
Transferring station "sbcDeviceTemplate"

What name would you like the new station to have?

Station name sbcDeviceTemplate

Then choose the option shown so that everything gets copied.

Then click Next



Transferring station "sbcDeviceTemplate"

Which station files are to be copied?

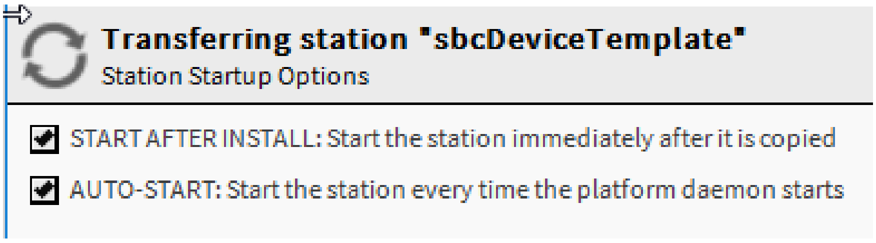
◇ Copy files from selected directories

◈ Copy every file in the station directory and its subdirectories

◇ Copy only the "config.bog" station database file

Set the startup options according to your needs.

Then click Next and finally Done



Transferring station "sbcDeviceTemplate"
Station Startup Options

☑ START AFTER INSTALL: Start the station immediately after it is copied

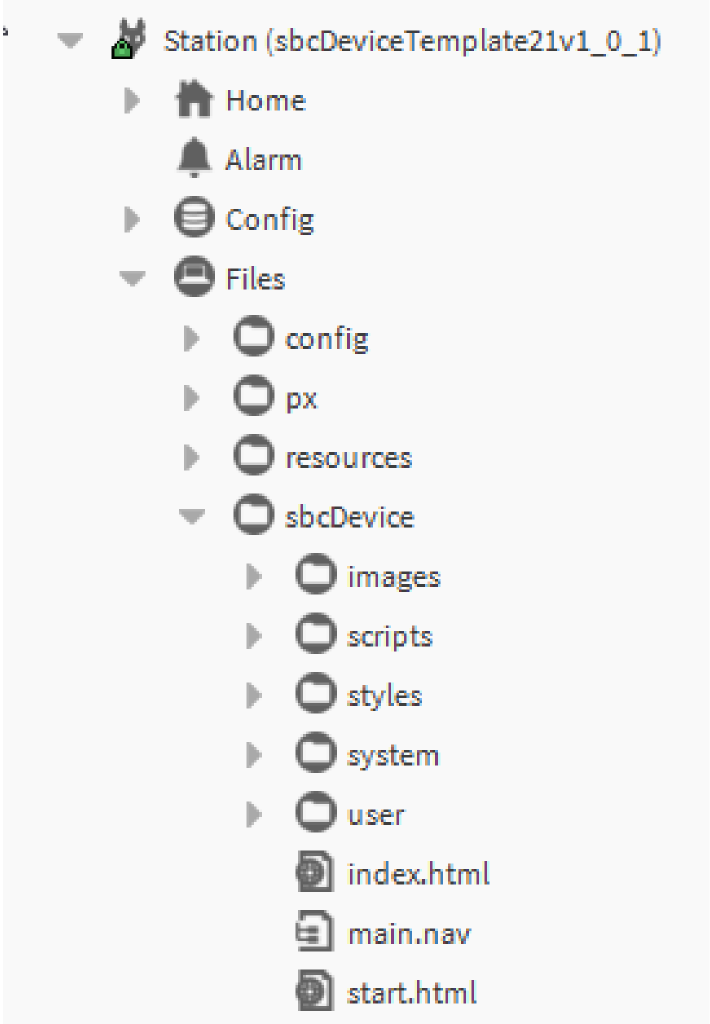
☑ AUTO-START: Start the station every time the platform daemon starts

4.2 Incorporating a device template in an existing project

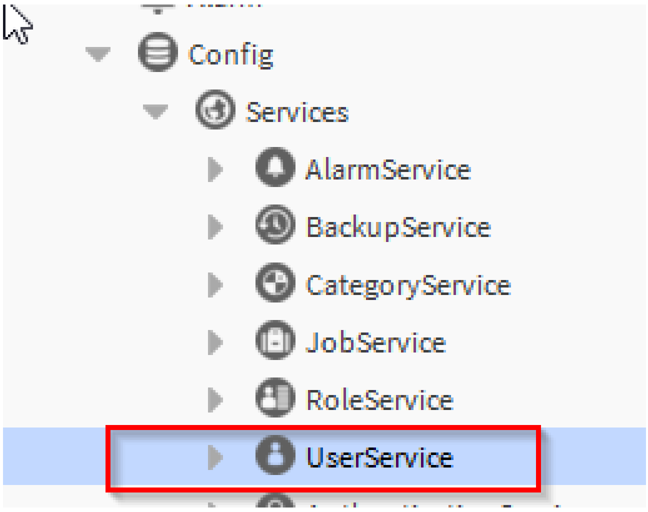
The whole template can also be transferred directly to an already running station.

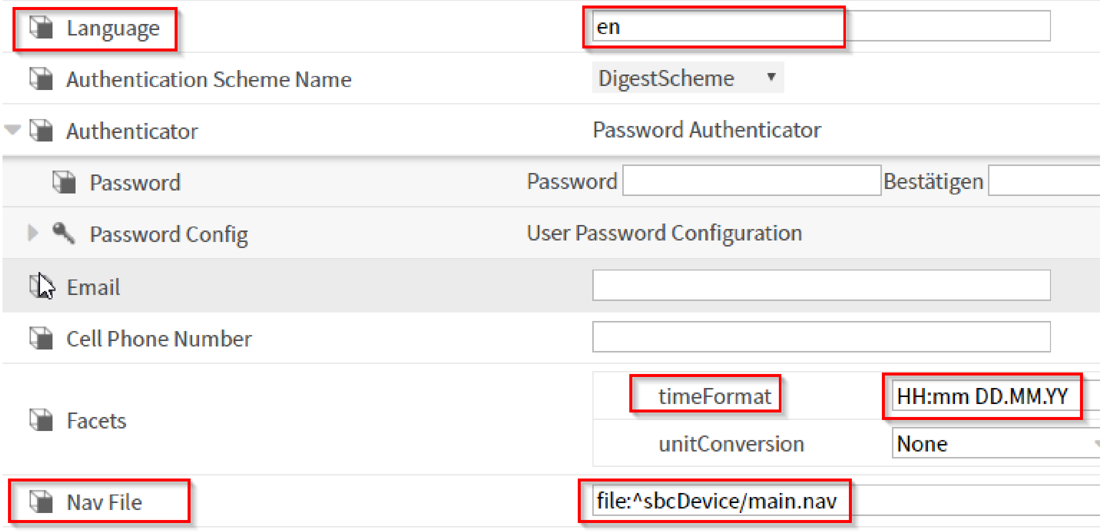
Copy the “sbcDevice” folder from the “..\sbcDeviceTemplate21vX_X_X\shared\” folder to the system path of your station: “C:\ProgramData\Niagara4.8\SBC\stations\IhreStation\shared\” for Supervisor 2.1 or “C:\ProgramData\Niagara4.9\SBC\stations\IhreStation\shared\” for Supervisor 3.0

You will then find the “sbcDevice” folder in the station under Files (update if necessary):



To ensure that the whole thing can be used, the main.nav file, the language and the time format must be specified for each user in the “User Service”:

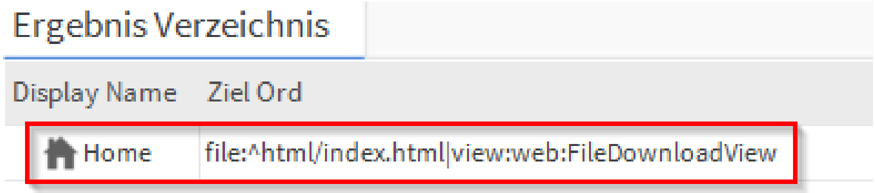




Language	en
Authentication Scheme Name	DigestScheme ▾
Authenticator	Password Authenticator
Password	Password Bestätigen
Password Config	User Password Configuration
Email	
Cell Phone Number	
Facets	timeFormat HH:mm DD.MM.YY unitConversion None
Nav File	file:^sbcDevice/main.nav

In addition, the date format can be specified here, which later defines the time and date display in the template.

The main.nav transforms the index.html with the corresponding view into the home page:



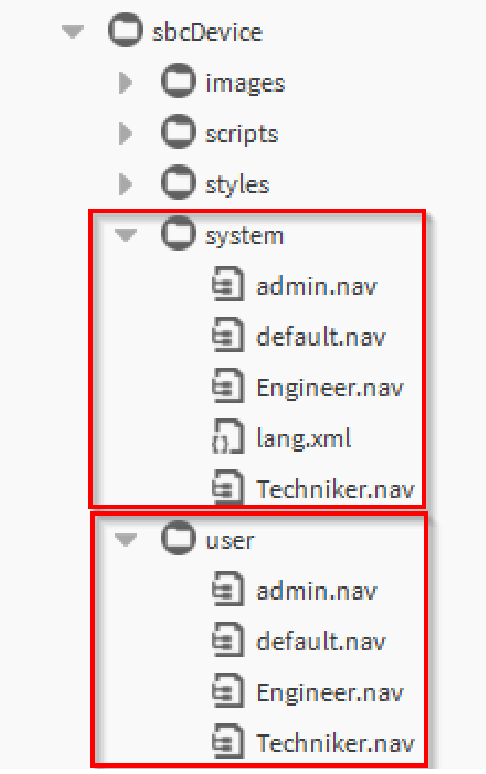
Ergebnis Verzeichnis

Display Name	Ziel Ord
Home	file:^html/index.html|view:web:FileDownloadView

5. Configuration

5.1 Configuration files

A nav file must be created for each user in the “users” and “system” folders.



- sbcDevice
  - images
  - scripts
  - styles
  - system
    - admin.nav
    - default.nav
    - Engineer.nav
    - lang.xml
    - Techniker.nav
  - user
    - admin.nav
    - default.nav
    - Engineer.nav
    - Techniker.nav

The name must be the same as the user name (case-sensitive)!

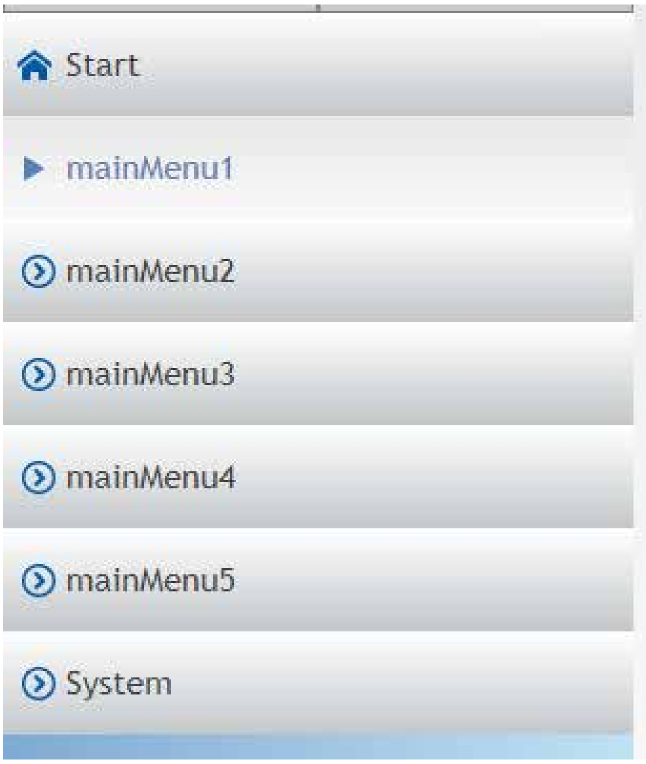
This enables the entire template to be adapted individually for each user and, if necessary, in different languages.

The default.nav can be used as a template.

If the user-specific NAV file is missing, the default.nav file is used!

5.2 User Nav File menu

The entire menu is defined In the nav file under “Users” as what the user is intended to see



The user nav file is fully customisable in terms of its structure and name.

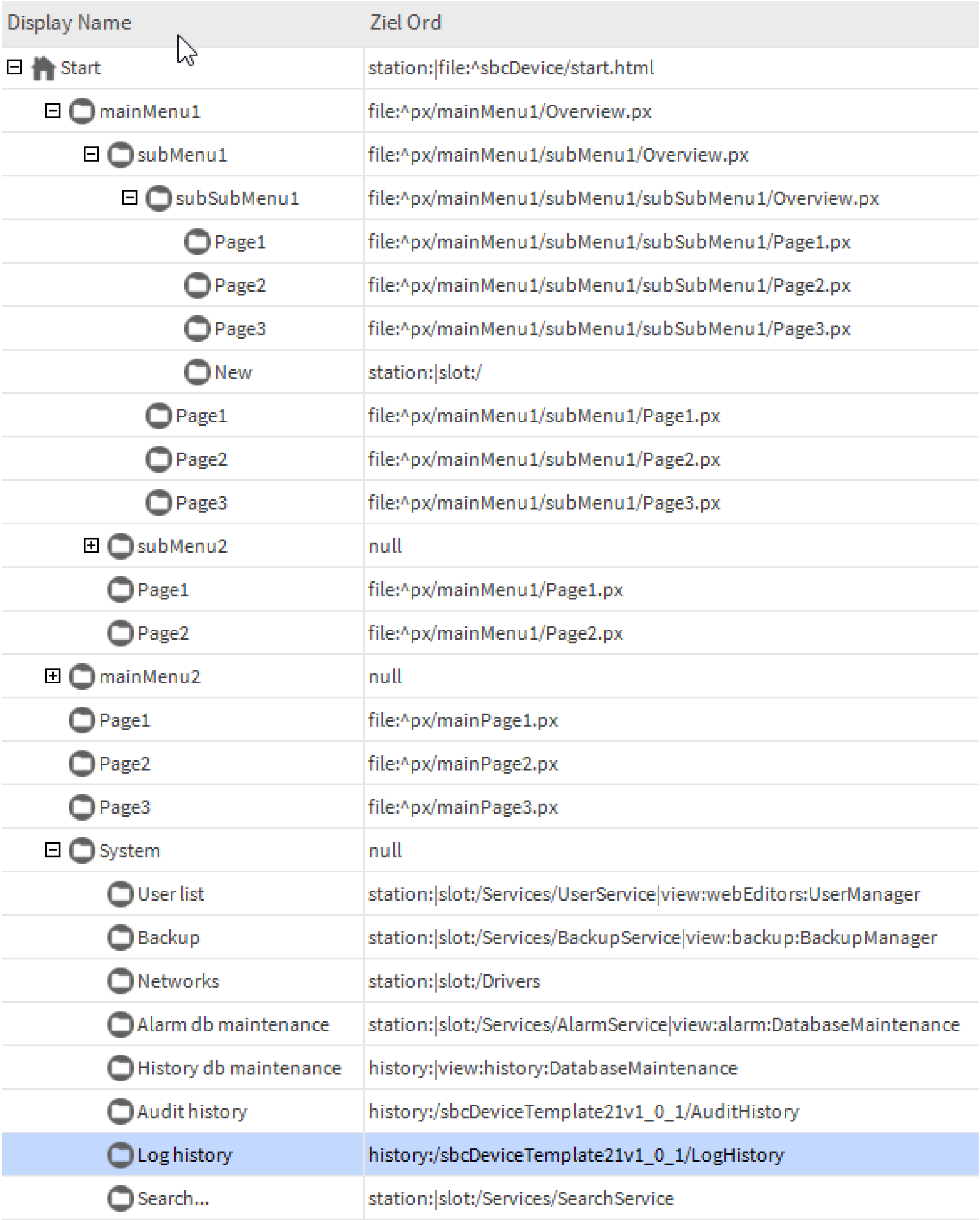
The first entry is permanently defined as the home page in the menu and has no submenus.

The entries under “Start” define the structure of the menu below in hierarchical terms.

Up to 4 levels are possible!

1. mainMenu
    2. subMenu
        3. subSubMenu
            4. Pages

As is usual with all NAV files in Niagara, everything can be linked here.

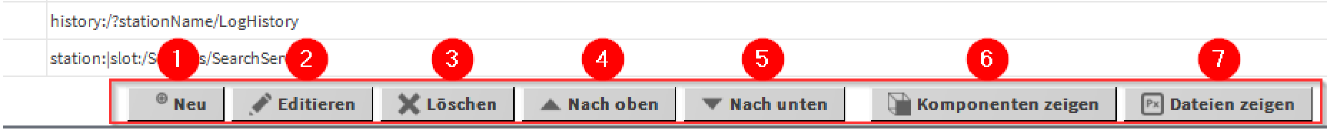


Display Name	Ziel Ord
⊟ Start	station:|file:^sbcDevice/start.html
⊟ mainMenu1	file:^px/mainMenu1/Overview.px
⊟ subMenu1	file:^px/mainMenu1/subMenu1/Overview.px
⊟ subSubMenu1	file:^px/mainMenu1/subMenu1/subSubMenu1/Overview.px
Page1	file:^px/mainMenu1/subMenu1/subSubMenu1/Page1.px
Page2	file:^px/mainMenu1/subMenu1/subSubMenu1/Page2.px
Page3	file:^px/mainMenu1/subMenu1/subSubMenu1/Page3.px
New	station:|slot:/
Page1	file:^px/mainMenu1/subMenu1/Page1.px
Page2	file:^px/mainMenu1/subMenu1/Page2.px
Page3	file:^px/mainMenu1/subMenu1/Page3.px
⊞ subMenu2	null
Page1	file:^px/mainMenu1/Page1.px
Page2	file:^px/mainMenu1/Page2.px
⊞ mainMenu2	null
Page1	file:^px/mainPage1.px
Page2	file:^px/mainPage2.px
Page3	file:^px/mainPage3.px
⊟ System	null
User list	station:|slot:/Services/UserService|view:webEditors:UserManager
Backup	station:|slot:/Services/BackupService|view:backup:BackupManager
Networks	station:|slot:/Drivers
Alarm db maintenance	station:|slot:/Services/AlarmService|view:alarm:DatabaseMaintenance
History db maintenance	history:|view:history:DatabaseMaintenance
Audit history	history:/sbcDeviceTemplate21v1_0_1/AuditHistory
Log history	history:/sbcDeviceTemplate21v1_0_1/LogHistory
Search...	station:|slot:/Services/SearchService

Links can also contain additional information about the "view" (e.g. | view:alarm:DatabaseMaintenance)

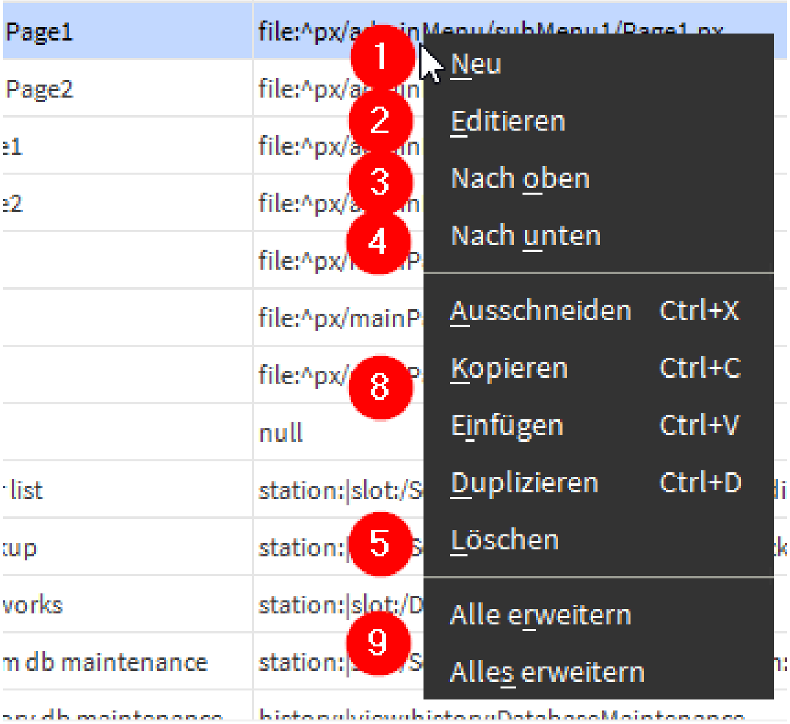
❗	From version 1.0.4 onwards	ORDs in the user nav file can now contain *"?stationName"*, which is replaces with the station name (sbcDeviceTemplateXXvX_X_X). e.g. history: *"/?stationName/AuditHistory"*
⚠	Attention, with versions older than 1.0.4	Links for AuditHistory or LogHistory must be modified when renaming the device, as they contain the device name.
❗	From version 0.8.4 onwards	It is also possible to specify *"|view:?fullScreen=false"* or *"?fullscreen=false"* in the nav file, so that the nav tree can be seen.

5.2.1. Editing the entries in the NAV file



If an entry is selected, it can be edited (2), moved (4 and 5) or deleted (3) using the buttons on the lower bar.

You can also double-click an entry to edit it.

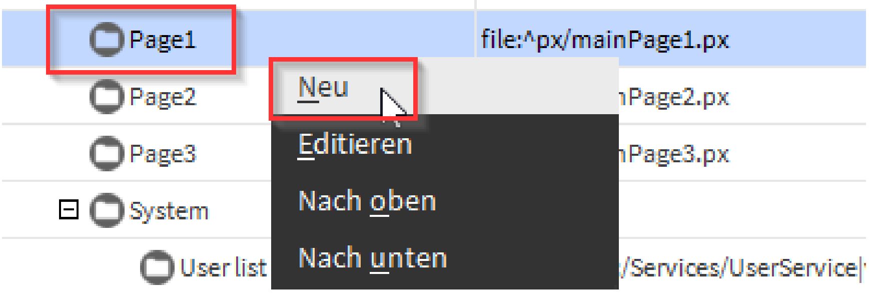


The same is also possible if you right-click to open the context menu at a particular point.

You also have the option of copying or reproducing points here (8).

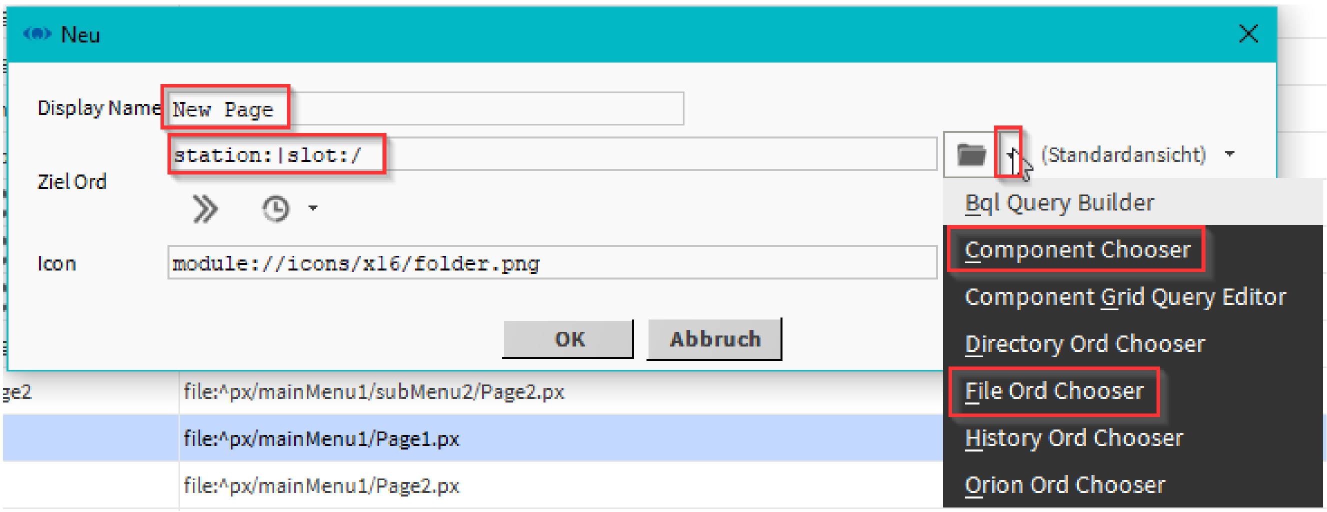
You can also expand and collapse the view of the levels (9).

You can use the New (1) button to create a new entry.



You will then see this popup window.

The element can be given a name in the first field which is subsequently displayed in the menu for this purpose. In the second field you can save or select a folder for any component or file.



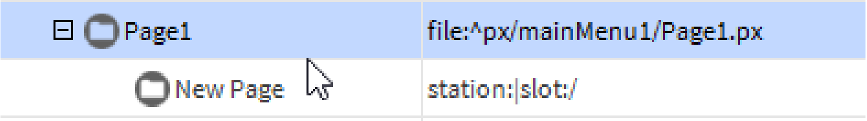
Neu

Display Name: New Page

Ziel Ord: station:|slot:/

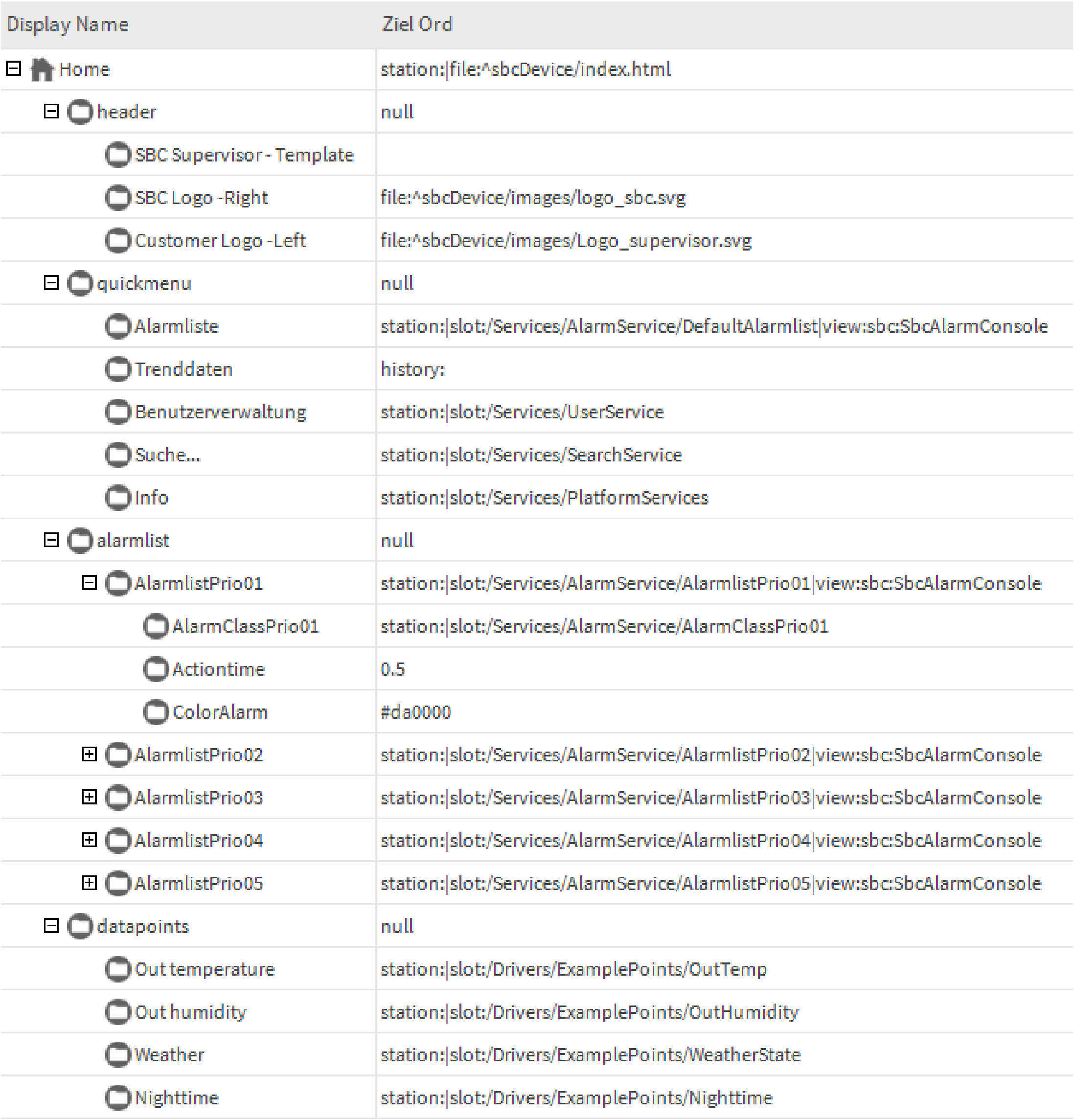
Icon: module://icons/x16/folder.png

Please note that a new entry is always created below the element that you have selected.



Page1	file:^px/mainMenu1/Page1.px
New Page	station:|slot:/

5.3 Nav file system



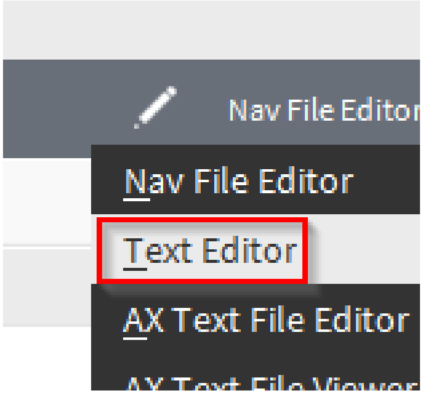
Display Name	Ziel Ord
⊟ Home	station:|file:^sbcDevice/index.html
⊟ header	null
SBC Supervisor - Template	
SBC Logo -Right	file:^sbcDevice/images/logo_sbc.svg
Customer Logo -Left	file:^sbcDevice/images/Logo_supervisor.svg
⊟ quickmenu	null
Alarmliste	station:|slot:/Services/AlarmService/DefaultAlarmlist|view:sbc:SbcAlarmConsole
Trenddaten	history:
Benutzerverwaltung	station:|slot:/Services/UserService
Suche...	station:|slot:/Services/SearchService
Info	station:|slot:/Services/PlatformServices
⊟ alarmlist	null
⊟ AlarmlistPrio01	station:|slot:/Services/AlarmService/AlarmlistPrio01|view:sbc:SbcAlarmConsole
AlarmClassPrio01	station:|slot:/Services/AlarmService/AlarmClassPrio01
Actiontime	0.5
ColorAlarm	#da0000
⊞ AlarmlistPrio02	station:|slot:/Services/AlarmService/AlarmlistPrio02|view:sbc:SbcAlarmConsole
⊞ AlarmlistPrio03	station:|slot:/Services/AlarmService/AlarmlistPrio03|view:sbc:SbcAlarmConsole
⊞ AlarmlistPrio04	station:|slot:/Services/AlarmService/AlarmlistPrio04|view:sbc:SbcAlarmConsole
⊞ AlarmlistPrio05	station:|slot:/Services/AlarmService/AlarmlistPrio05|view:sbc:SbcAlarmConsole
⊟ datapoints	null
Out temperature	station:|slot:/Drivers/ExamplePoints/OutTemp
Out humidity	station:|slot:/Drivers/ExamplePoints/OutHumidity
Weather	station:|slot:/Drivers/ExamplePoints/WeatherState
Nighttime	station:|slot:/Drivers/ExamplePoints/Nighttime

All other points of the template are defined in the system file.

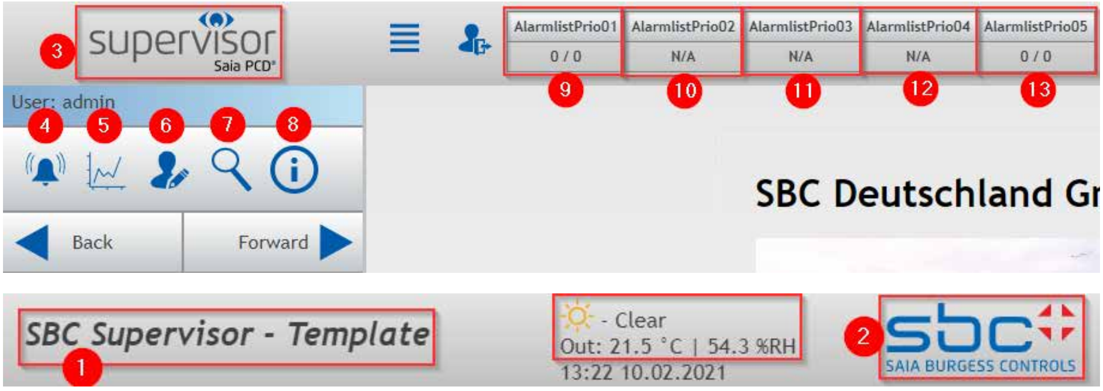
The structure or layout must not be changed, otherwise the template will no longer work.

!	*From version 0.9.6 onwards*	*The entry for “header/user” and for “navigation” is not included in the system nav file.* *No compatibility with versions ≤ 0.9.5; older versions need to be adapted (deleted) as necessary.*

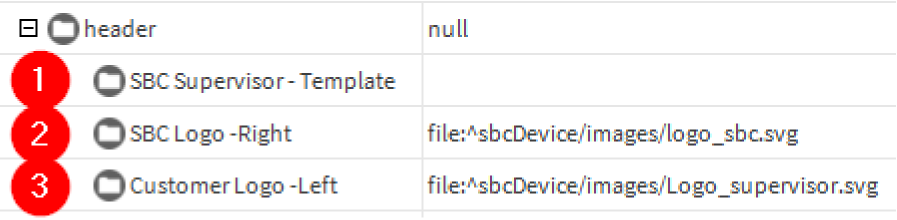
Some special changes require the text editor, as the selection window of the NavFile editor does not accept all entries.



Explanation of the structure:



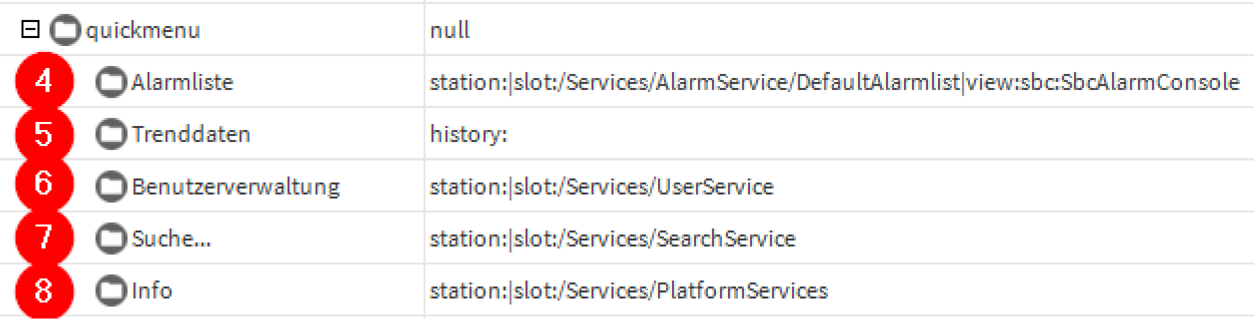
5.3.1 Header



	⊟ header	null
1	SBC Supervisor - Template	
2	SBC Logo -Right	file:^sbcDevice/images/logo_sbc.svg
3	Customer Logo -Left	file:^sbcDevice/images/Logo_supervisor.svg

1. Template header
2. Logo on the right hand side
3. Logo on the left hand side
   SVG, PNG, GIF or JPEG files can be used as images, for example.
   In the case of fixed-pixel images such as PNG, GIF or JPEG, the height should be 50 pixels.
   If no images are needed, then enter “null” or “none” in the “ord” field instead

5.3.2 Quick menu



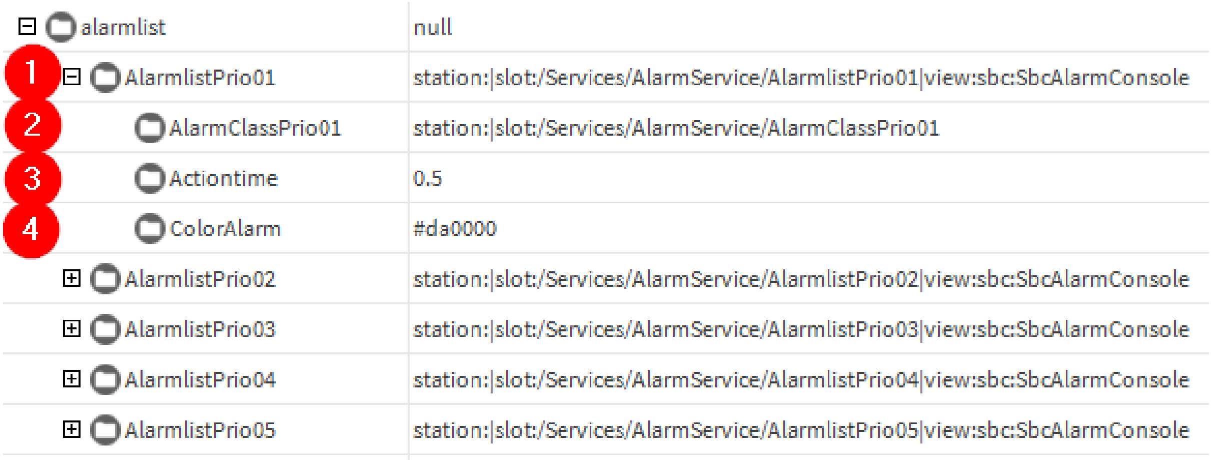
	⊟ quickmenu	null
4	Alarmliste	station:|slot:/Services/AlarmService/DefaultAlarmlist|view:sbc:SbcAlarmConsole
5	Trenddaten	history:
6	Benutzerverwaltung	station:|slot:/Services/UserService
7	Suche...	station:|slot:/Services/SearchService
8	Info	station:|slot:/Services/PlatformServices

4. Name and link to the list of alarms
5. Name and link to the history
6. Name and link to user management
7. Name and link to the search function[1]
8. Name and link to the info

If an item in the quick menu is to be hidden or left unused, enter “null” or “none” in the “ord” field instead.

[1] Note: The search function is used to search for items using the tagging function. If the function is not required, it can simply be deactivated.

5.3.3 Alarm list



	⊟ alarmlist	null
1	⊟ AlarmlistPrio01	station:|slot:/Services/AlarmService/AlarmlistPrio01|view:sbc:SbcAlarmConsole
2	AlarmClassPrio01	station:|slot:/Services/AlarmService/AlarmClassPrio01
3	Actiontime	0.5
4	ColorAlarm	#da0000
	⊞ AlarmlistPrio02	station:|slot:/Services/AlarmService/AlarmlistPrio02|view:sbc:SbcAlarmConsole
	⊞ AlarmlistPrio03	station:|slot:/Services/AlarmService/AlarmlistPrio03|view:sbc:SbcAlarmConsole
	⊞ AlarmlistPrio04	station:|slot:/Services/AlarmService/AlarmlistPrio04|view:sbc:SbcAlarmConsole
	⊞ AlarmlistPrio05	station:|slot:/Services/AlarmService/AlarmlistPrio05|view:sbc:SbcAlarmConsole

There are 5 ready-made alarm lists (shown above from 9 to 13).

1. The first topmost entry contains the name of the list and the link to the alarm console (ConsoleRecipient).
   If an alarm list is to be hidden or left unused, enter “null” or “none” in the “ord” field.
   More options can be set in the 3 fields below the alarm list. The name is of no relevance, just the “ord” field.
2. The path of the associated alarm class is defined in the first subfield.
3. A time can be set for the intervals at which the macro should flash for new alarms in the second subfield. If you do not want that, enter “null” or “none” in the “ord” field.
4. In the third field you can define the colour of the alarm field when an alarm is active.
   The colours can be defined according to the HTML/CSS standard.

   See:

   https://www.w3schools.com/cssref/css_colors_legal.asp
   https://www.w3schools.com/colors/colors_names.asp

The following definitions are thus possible:

- **Hexadecimal colour specification**

  A hexadecimal colour is specified with: #RRGGBB, where the hexadecimal integers RR (red), GG (green) and BB (blue) specify the components of the colour. All values must be between 00 and FF.

  #ff0000 Red

  #00ff00 Green

  #0000ff Blue

- **RGB colours**

  An RGB colour value is specified with the rgb () function, which has the following syntax:

  rgb(red, green, blue)

  Each parameter (red, green and blue) defines the intensity of the colour and can be an integer between 0 and 255 or a percentage (from 0% to 100%).

  For example, the value for rgb (0,0,255) is displayed as blue because the parameter blue is set to the highest value (255) and the others are set to 0.

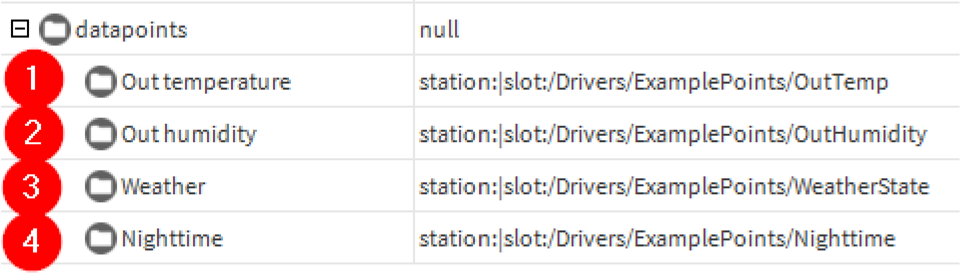
  The following values also define the same colour: rgb (0,0,255) and rgb (0%,0%,100%).

  - **Predefined colour names common to all browsers**

    https://www.w3schools.com/colors/colors_names.asp

  - **ETC. See links**

5.3.4 Datapoints



⊟ datapoints	null
Out temperature	station:|slot:/Drivers/ExamplePoints/OutTemp
Out humidity	station:|slot:/Drivers/ExamplePoints/OutHumidity
Weather	station:|slot:/Drivers/ExamplePoints/WeatherState
Nighttime	station:|slot:/Drivers/ExamplePoints/Nighttime

The following applies to all values: If you do not want a value displayed, enter “null” or “none” in the “ord” field.

1. A datapoint for the outdoor temperature can be entered in the “Ord” field for “Out Temperature”.
2. A datapoint for the outdoor humidity can be entered in the “Ord” field for “Out humidity”.
3. A datapoint for the weather conditions can be entered in the “Ord” field for “Weather”.
4. A data point can be entered for the status of day and night, the weather display in the “Ord” field of “Nighttime”.

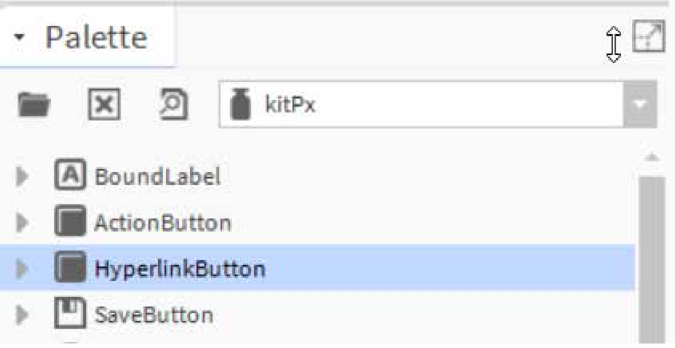
For more details see Chapter 9 “Connecting a weather station (BETA)”.

6. Special features on PX pages

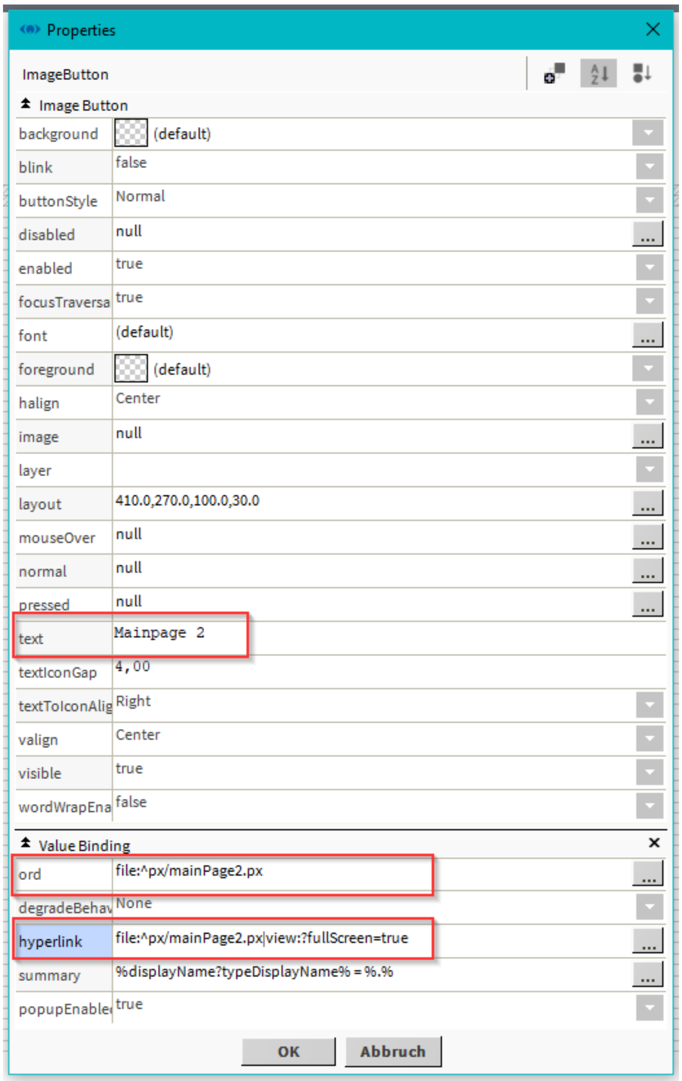
6.1. Use of buttons in connection with the device template

In order to navigate to a PX page in connection with the device template, navigation buttons must have an extra option: “|view:?fullScreen=true” or “?fullScreen=true”.

1. First drag a HyperlinkButton from the kitPx palette to the PX page.



Secondly, we have to define a hyperlink to the target page in the properties:



ImageButton

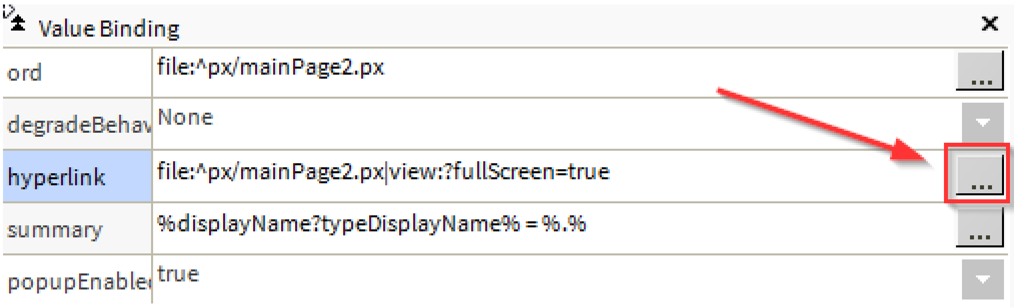
Image Button

background	(default)
blink	false
buttonStyle	Normal
disabled	null
enabled	true
focusTraversa	true
font	(default)
foreground	(default)
halign	Center
image	null
layer	
layout	410.0,270.0,100.0,30.0
mouseOver	null
normal	null
pressed	null
text	Mainpage 2
textIconGap	4,00
textToIconAlig	Right
valign	Center
visible	true
wordWrapEna	false

Value Binding

ord	file:^px/mainPage2.px
degradeBehav	None
hyperlink	file:^px/mainPage2.px|view:?fullScreen=true
summary	%displayName?typeDisplayName% = %.%
popupEnable	true

3. Open the “hyperlink” option:

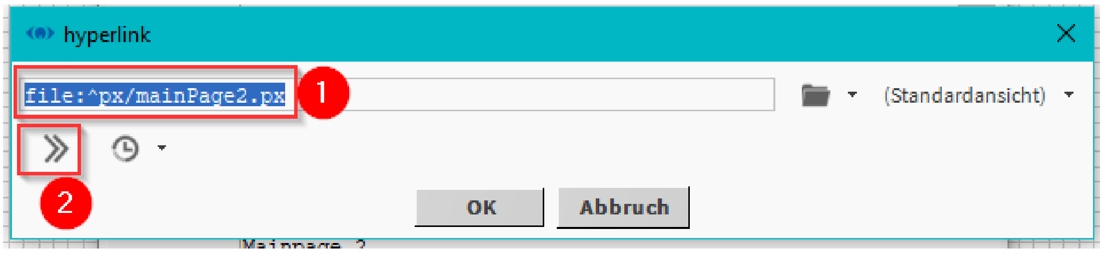


Value Binding

ord	file:^px/mainPage2.px
degradeBehav	None
hyperlink	file:^px/mainPage2.px|view:?fullScreen=true
summary	%displayName?typeDisplayName% = %.%
popupEnable	true

4. Then define the folder in the usual way (1).

Then click on the two arrows (2);

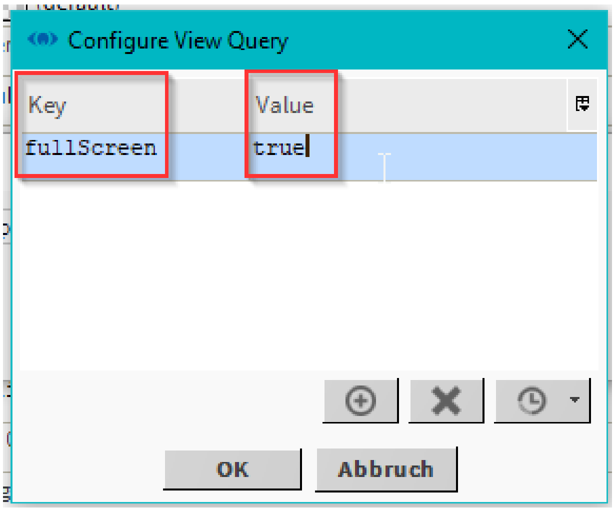


hyperlink

file:^px/mainPage2.px

(Standardansicht)

5. In the subsequent window enter “fullScreen” in the Key field and “true” in the Value field:

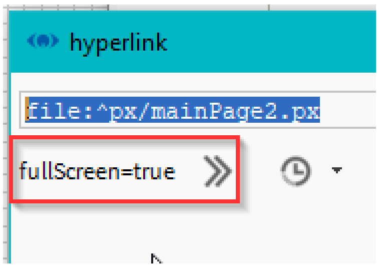


Configure View Query

Key	Value
fullScreen	true

6. The window can now be confirmed with “OK”

You can then see the change here. Confirm again with “OK”

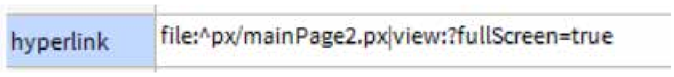


hyperlink

file:^px/mainPage2.px

fullScreen=true

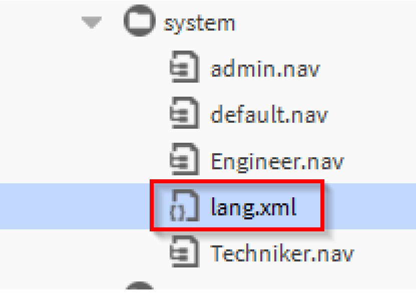
You will now also see the change in the “Hyperlink” field.

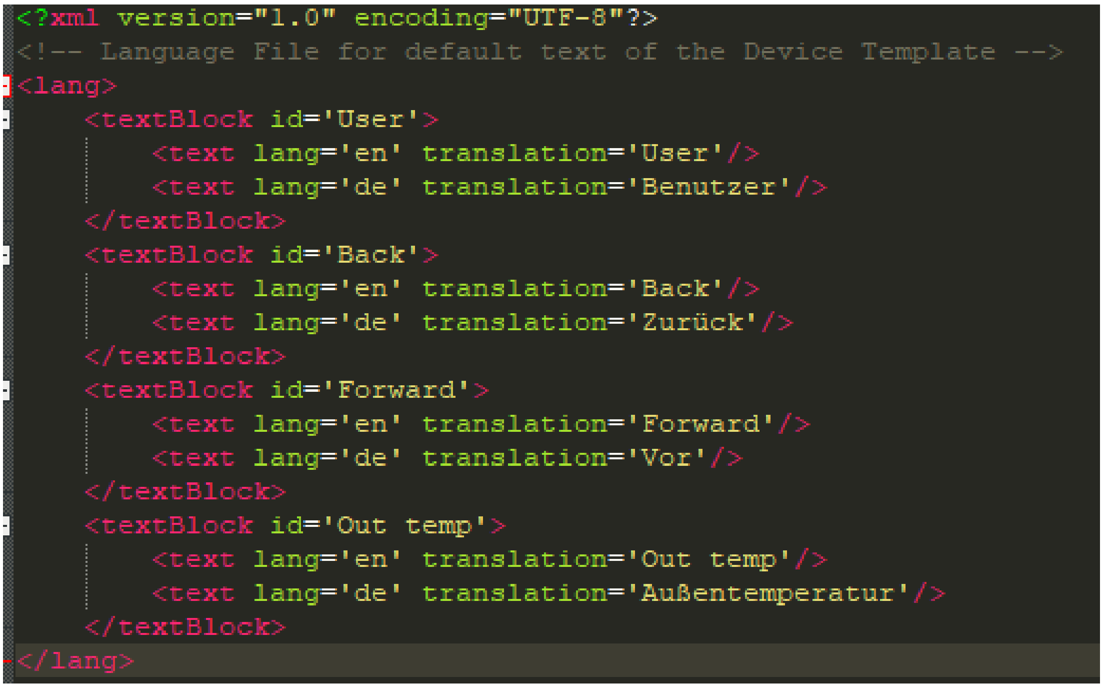


hyperlink file:^px/mainPage2.px|view:?fullScreen=true

7. Translation of the standard texts

The “system” folder contains a file named *lang.xml* where additional languages can be defined as necessary.





```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Language File for default text of the Device Template -->
<lang>
    <textBlock id='User'>
        <text lang='en' translation='User'/>
        <text lang='de' translation='Benutzer'/>
    </textBlock>
    <textBlock id='Back'>
        <text lang='en' translation='Back'/>
        <text lang='de' translation='Zurück'/>
    </textBlock>
    <textBlock id='Forward'>
        <text lang='en' translation='Forward'/>
        <text lang='de' translation='Vor'/>
    </textBlock>
    <textBlock id='Out temp'>
        <text lang='en' translation='Out temp'/>
        <text lang='de' translation='Außentemperatur'/>
    </textBlock>
</lang>
```

Simply copy the text tag (<text…>) and define the language under “lang” and define the text in the respective language under “translation”.

German, English, French and Italian versions are already available in the example shown.

If a language is not defined, the English original is used automatically.

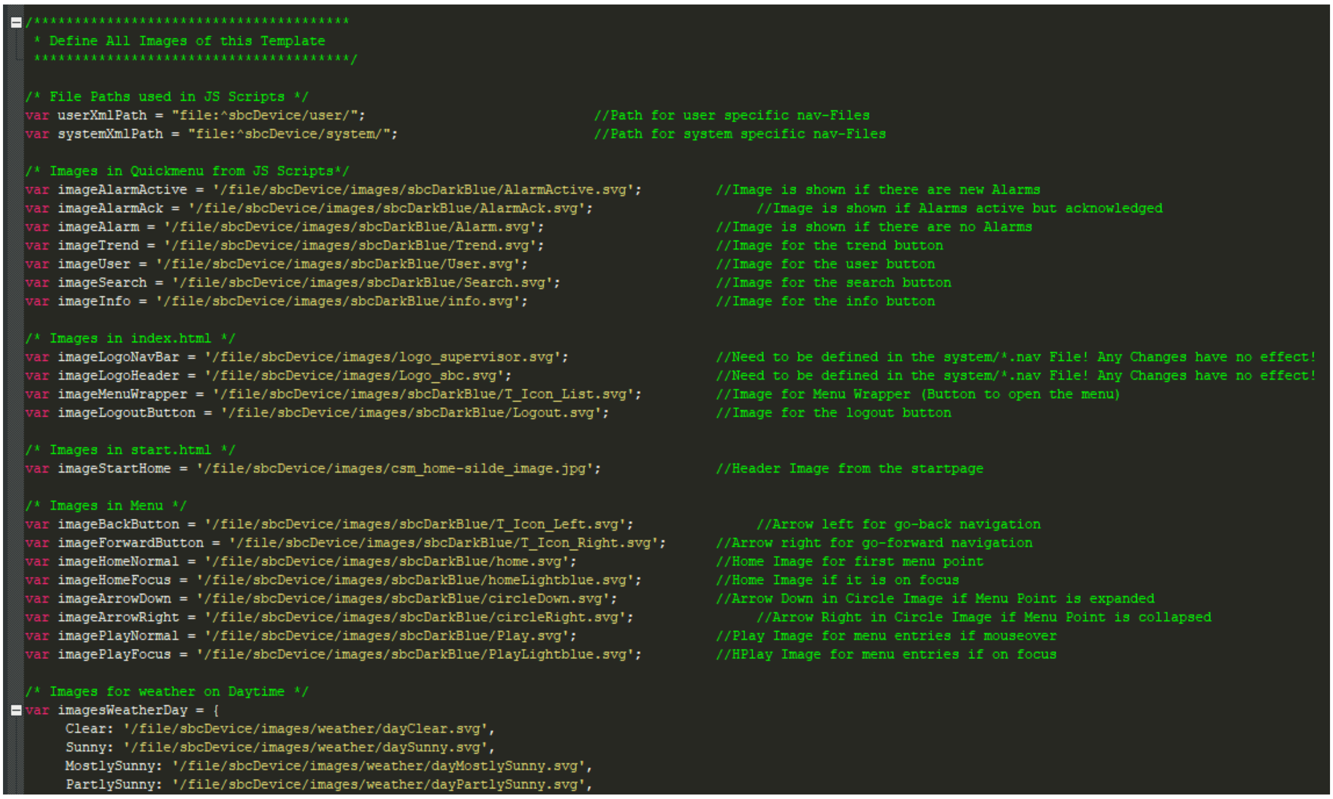
8. Changing the images

Only possible from version 1.0.11 onwards.

There is a file for changing the images under “Scripts \ theme.js”

All paths to the images used can be found there.

These can be customized.



```
/***************************************
 * Define All Images of this Template
 ***************************************/

/* File Paths used in JS Scripts */
var userXmlPath = "file:^sbcDevice/user/";                            //Path for user specific nav-Files
var systemXmlPath = "file:^sbcDevice/system/";                        //Path for system specific nav-Files

/* Images in Quickmenu from JS Scripts*/
var imageAlarmActive = '/file/sbcDevice/images/sbcDarkBlue/AlarmActive.svg';         //Image is shown if there are new Alarms
var imageAlarmAck = '/file/sbcDevice/images/sbcDarkBlue/AlarmAck.svg';                    //Image is shown if Alarms active but acknowledged
var imageAlarm = '/file/sbcDevice/images/sbcDarkBlue/Alarm.svg';                     //Image is shown if there are no Alarms
var imageTrend = '/file/sbcDevice/images/sbcDarkBlue/Trend.svg';                     //Image for the trend button
var imageUser = '/file/sbcDevice/images/sbcDarkBlue/User.svg';                       //Image for the user button
var imageSearch = '/file/sbcDevice/images/sbcDarkBlue/Search.svg';                   //Image for the search button
var imageInfo = '/file/sbcDevice/images/sbcDarkBlue/info.svg';                       //Image for the info button

/* Images in index.html */
var imageLogoNavBar = '/file/sbcDevice/images/logo_supervisor.svg';                  //Need to be defined in the system/*.nav File! Any Changes have no effect!
var imageLogoHeader = '/file/sbcDevice/images/Logo_sbc.svg';                         //Need to be defined in the system/*.nav File! Any Changes have no effect!
var imageMenuWrapper = '/file/sbcDevice/images/sbcDarkBlue/T_Icon_List.svg';         //Image for Menu Wrapper (Button to open the menu)
var imageLogoutButton = '/file/sbcDevice/images/sbcDarkBlue/Logout.svg';             //Image for the logout button

/* Images in start.html */
var imageStartHome = '/file/sbcDevice/images/csm_home-silde_image.jpg';              //Header Image from the startpage

/* Images in Menu */
var imageBackButton = '/file/sbcDevice/images/sbcDarkBlue/T_Icon_Left.svg';               //Arrow left for go-back navigation
var imageForwardButton = '/file/sbcDevice/images/sbcDarkBlue/T_Icon_Right.svg';      //Arrow right for go-forward navigation
var imageHomeNormal = '/file/sbcDevice/images/sbcDarkBlue/home.svg';                 //Home Image for first menu point
var imageHomeFocus = '/file/sbcDevice/images/sbcDarkBlue/homeLightblue.svg';         //Home Image if it is on focus
var imageArrowDown = '/file/sbcDevice/images/sbcDarkBlue/circleDown.svg';            //Arrow Down in Circle Image if Menu Point is expanded
var imageArrowRight = '/file/sbcDevice/images/sbcDarkBlue/circleRight.svg';               //Arrow Right in Circle Image if Menu Point is collapsed
var imagePlayNormal = '/file/sbcDevice/images/sbcDarkBlue/Play.svg';                 //Play Image for menu entries if mouseover
var imagePlayFocus = '/file/sbcDevice/images/sbcDarkBlue/PlayLightblue.svg';         //HPlay Image for menu entries if on focus

/* Images for weather on Daytime */
var imagesWeatherDay = {
     Clear: '/file/sbcDevice/images/weather/dayClear.svg',
     Sunny: '/file/sbcDevice/images/weather/daySunny.svg',
     MostlySunny: '/file/sbcDevice/images/weather/dayMostlySunny.svg',
     PartlySunny: '/file/sbcDevice/images/weather/dayPartlySunny.svg',
```

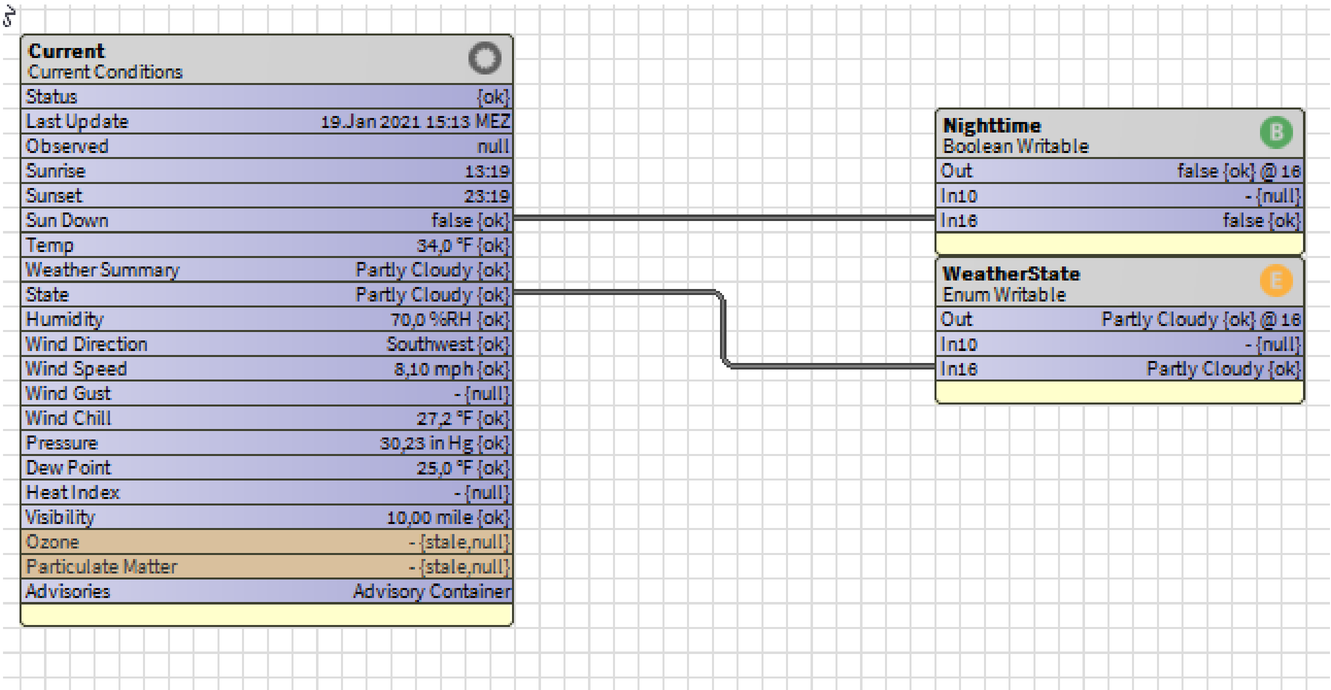
9. Connecting a weather station

Only possible from version 1.0.21 onwards.

The NAV files in the system path contain two new datapoints under Datapoints.

One is for the current Weather Status and one for Day and Night.

These can be linked to a weather function as follows:



Current
Current Conditions

Status	{ok}
Last Update	19.Jan 2021 15:13 MEZ
Observed	null
Sunrise	13:19
Sunset	23:19
Sun Down	false {ok}
Temp	34,0 °F {ok}
Weather Summary	Partly Cloudy {ok}
State	Partly Cloudy {ok}
Humidity	70,0 %RH {ok}
Wind Direction	Southwest {ok}
Wind Speed	8,10 mph {ok}
Wind Gust	- {null}
Wind Chill	27,2 °F {ok}
Pressure	30,23 in Hg {ok}
Dew Point	25,0 °F {ok}
Heat Index	- {null}
Visibility	10,00 mile {ok}
Ozone	- {stale,null}
Particulate Matter	- {stale,null}
Advisories	Advisory Container

Nighttime
Boolean Writable

Out	false {ok} @ 16
In10	- {null}
In16	false {ok}

WeatherState
Enum Writable

Out	Partly Cloudy {ok} @ 16
In10	- {null}
In16	Partly Cloudy {ok}

If a weather function is not required, the datapoint in the NAV file can be filled with “none” or “null”.

The first datapoint defines the weather status in the form of an EnumWritable.

The weather states are based on the example of the Niagara Weather Service.

In theory, any weather service can be connected, the only important thing is that the status is transmitted via a numerical value.

The weather status is linked to a keyword via the numerical value.

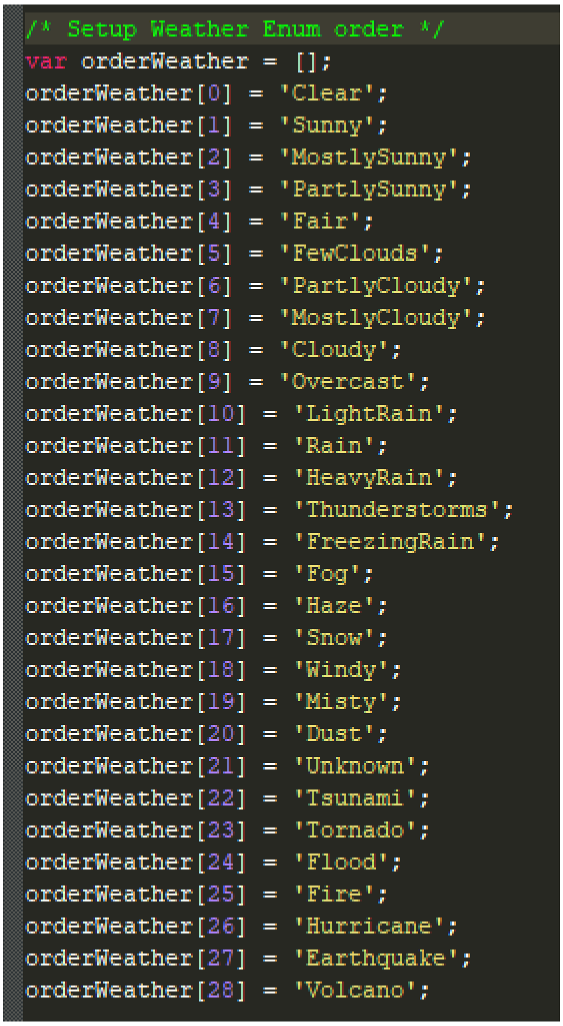
This keyword is then used to make a link to a weather symbol and a language-dependent text (for more details, see the sections below).

If the assignment of the weather status to the predefined numerical values is not identical, this can be adjusted or expanded.

There are several points that must be observed for this purpose, as outlined below.

9.1. Assignment of keywords to numerical values:

To be found in the “scripts/theme.js”



```
/* Setup Weather Enum order */
var orderWeather = [];
orderWeather[0] = 'Clear';
orderWeather[1] = 'Sunny';
orderWeather[2] = 'MostlySunny';
orderWeather[3] = 'PartlySunny';
orderWeather[4] = 'Fair';
orderWeather[5] = 'FewClouds';
orderWeather[6] = 'PartlyCloudy';
orderWeather[7] = 'MostlyCloudy';
orderWeather[8] = 'Cloudy';
orderWeather[9] = 'Overcast';
orderWeather[10] = 'LightRain';
orderWeather[11] = 'Rain';
orderWeather[12] = 'HeavyRain';
orderWeather[13] = 'Thunderstorms';
orderWeather[14] = 'FreezingRain';
orderWeather[15] = 'Fog';
orderWeather[16] = 'Haze';
orderWeather[17] = 'Snow';
orderWeather[18] = 'Windy';
orderWeather[19] = 'Misty';
orderWeather[20] = 'Dust';
orderWeather[21] = 'Unknown';
orderWeather[22] = 'Tsunami';
orderWeather[23] = 'Tornado';
orderWeather[24] = 'Flood';
orderWeather[25] = 'Fire';
orderWeather[26] = 'Hurricane';
orderWeather[27] = 'Earthquake';
orderWeather[28] = 'Volcano';
```

The order can be changed here by swapping the numbers (in square brackets) or keywords (text in quotation marks after the “=” sign).

New situations can also be added by expanding the list.

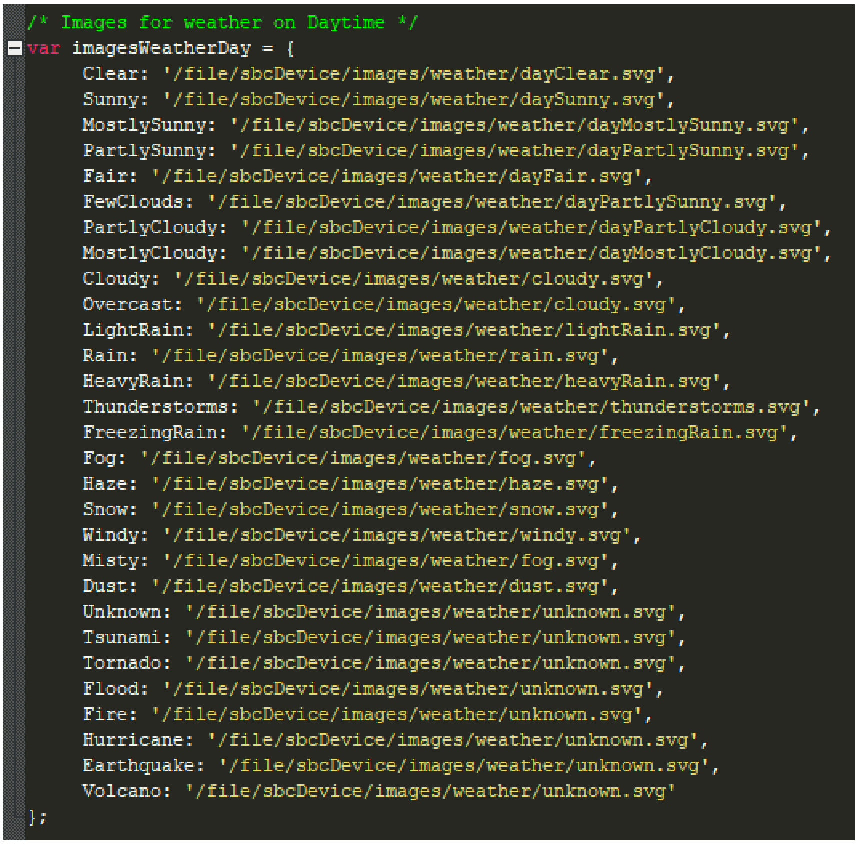
The only important thing here is that the number in square brackets is consecutive.

Duplications are not permitted.

9.2. Assignment of keywords to images

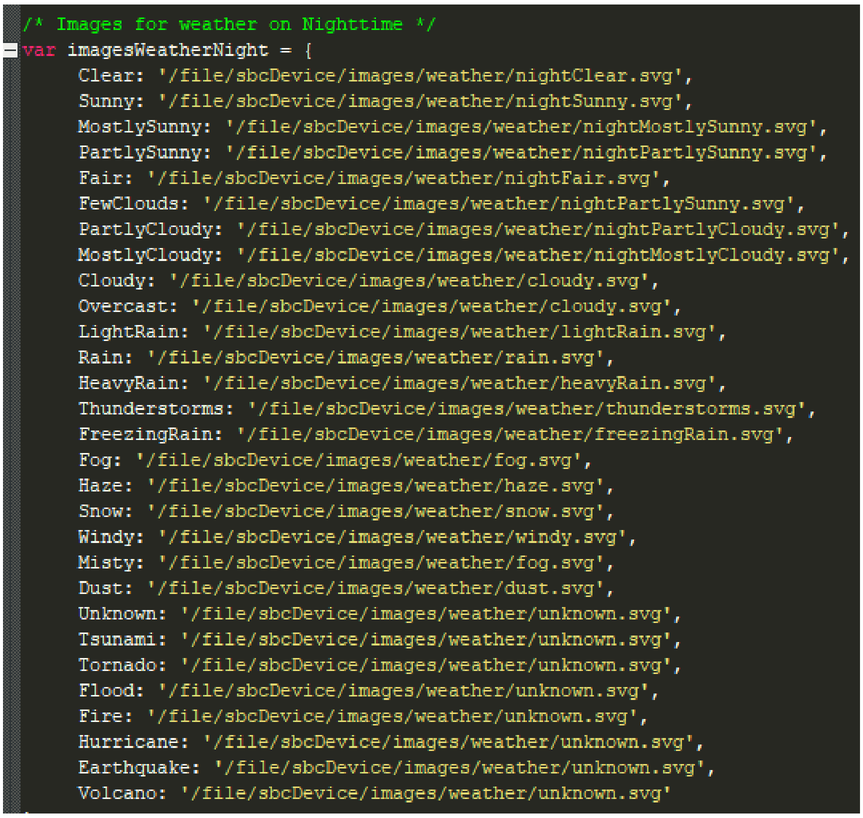
To be found in the “scripts/theme.js”

There is one area for images that are displayed during the daytime:



```
/* Images for weather on Daytime */
var imagesWeatherDay = {
    Clear: '/file/sbcDevice/images/weather/dayClear.svg',
    Sunny: '/file/sbcDevice/images/weather/daySunny.svg',
    MostlySunny: '/file/sbcDevice/images/weather/dayMostlySunny.svg',
    PartlySunny: '/file/sbcDevice/images/weather/dayPartlySunny.svg',
    Fair: '/file/sbcDevice/images/weather/dayFair.svg',
    FewClouds: '/file/sbcDevice/images/weather/dayPartlySunny.svg',
    PartlyCloudy: '/file/sbcDevice/images/weather/dayPartlyCloudy.svg',
    MostlyCloudy: '/file/sbcDevice/images/weather/dayMostlyCloudy.svg',
    Cloudy: '/file/sbcDevice/images/weather/cloudy.svg',
    Overcast: '/file/sbcDevice/images/weather/cloudy.svg',
    LightRain: '/file/sbcDevice/images/weather/lightRain.svg',
    Rain: '/file/sbcDevice/images/weather/rain.svg',
    HeavyRain: '/file/sbcDevice/images/weather/heavyRain.svg',
    Thunderstorms: '/file/sbcDevice/images/weather/thunderstorms.svg',
    FreezingRain: '/file/sbcDevice/images/weather/freezingRain.svg',
    Fog: '/file/sbcDevice/images/weather/fog.svg',
    Haze: '/file/sbcDevice/images/weather/haze.svg',
    Snow: '/file/sbcDevice/images/weather/snow.svg',
    Windy: '/file/sbcDevice/images/weather/windy.svg',
    Misty: '/file/sbcDevice/images/weather/fog.svg',
    Dust: '/file/sbcDevice/images/weather/dust.svg',
    Unknown: '/file/sbcDevice/images/weather/unknown.svg',
    Tsunami: '/file/sbcDevice/images/weather/unknown.svg',
    Tornado: '/file/sbcDevice/images/weather/unknown.svg',
    Flood: '/file/sbcDevice/images/weather/unknown.svg',
    Fire: '/file/sbcDevice/images/weather/unknown.svg',
    Hurricane: '/file/sbcDevice/images/weather/unknown.svg',
    Earthquake: '/file/sbcDevice/images/weather/unknown.svg',
    Volcano: '/file/sbcDevice/images/weather/unknown.svg'
};
```

As well as a separate area for images that are displayed during the night:



```
/* Images for weather on Nighttime */
var imagesWeatherNight = {
    Clear: '/file/sbcDevice/images/weather/nightClear.svg',
    Sunny: '/file/sbcDevice/images/weather/nightSunny.svg',
    MostlySunny: '/file/sbcDevice/images/weather/nightMostlySunny.svg',
    PartlySunny: '/file/sbcDevice/images/weather/nightPartlySunny.svg',
    Fair: '/file/sbcDevice/images/weather/nightFair.svg',
    FewClouds: '/file/sbcDevice/images/weather/nightPartlySunny.svg',
    PartlyCloudy: '/file/sbcDevice/images/weather/nightPartlyCloudy.svg',
    MostlyCloudy: '/file/sbcDevice/images/weather/nightMostlyCloudy.svg',
    Cloudy: '/file/sbcDevice/images/weather/cloudy.svg',
    Overcast: '/file/sbcDevice/images/weather/cloudy.svg',
    LightRain: '/file/sbcDevice/images/weather/lightRain.svg',
    Rain: '/file/sbcDevice/images/weather/rain.svg',
    HeavyRain: '/file/sbcDevice/images/weather/heavyRain.svg',
    Thunderstorms: '/file/sbcDevice/images/weather/thunderstorms.svg',
    FreezingRain: '/file/sbcDevice/images/weather/freezingRain.svg',
    Fog: '/file/sbcDevice/images/weather/fog.svg',
    Haze: '/file/sbcDevice/images/weather/haze.svg',
    Snow: '/file/sbcDevice/images/weather/snow.svg',
    Windy: '/file/sbcDevice/images/weather/windy.svg',
    Misty: '/file/sbcDevice/images/weather/fog.svg',
    Dust: '/file/sbcDevice/images/weather/dust.svg',
    Unknown: '/file/sbcDevice/images/weather/unknown.svg',
    Tsunami: '/file/sbcDevice/images/weather/unknown.svg',
    Tornado: '/file/sbcDevice/images/weather/unknown.svg',
    Flood: '/file/sbcDevice/images/weather/unknown.svg',
    Fire: '/file/sbcDevice/images/weather/unknown.svg',
    Hurricane: '/file/sbcDevice/images/weather/unknown.svg',
    Earthquake: '/file/sbcDevice/images/weather/unknown.svg',
    Volcano: '/file/sbcDevice/images/weather/unknown.svg'
```

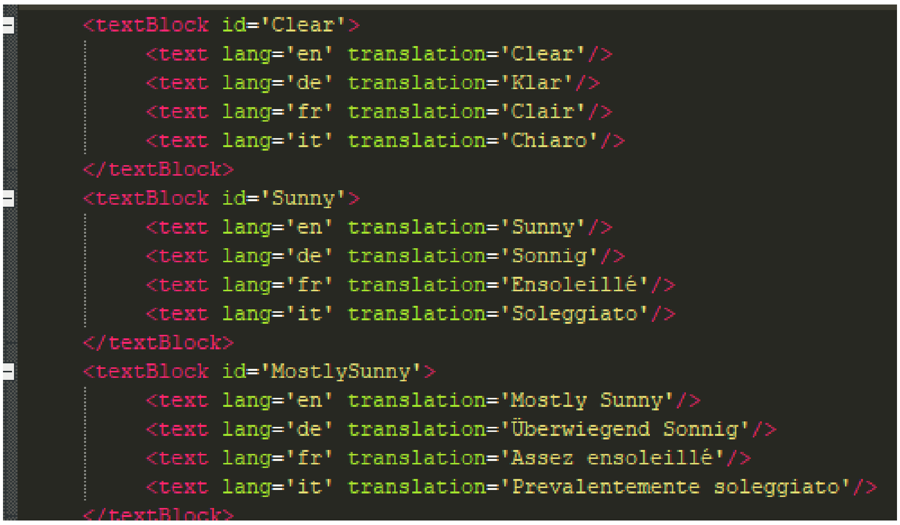
The first element is the keyword already mentioned in the previous chapter, followed by a colon. The path to the corresponding image is then appears between the quotation marks.

9.3. Assignment of keywords to a language-dependent text

Language-dependent texts are defined in the “system/lang.xml” file.

The keyword is specified under “id” in the “textBlock”.

Beneath that, the language is defined in the “text” tags under “lang” and the associated text under “translation”.



```
<textBlock id='Clear'>
    <text lang='en' translation='Clear'/>
    <text lang='de' translation='Klar'/>
    <text lang='fr' translation='Clair'/>
    <text lang='it' translation='Chiaro'/>
</textBlock>
<textBlock id='Sunny'>
    <text lang='en' translation='Sunny'/>
    <text lang='de' translation='Sonnig'/>
    <text lang='fr' translation='Ensoleillé'/>
    <text lang='it' translation='Soleggiato'/>
</textBlock>
<textBlock id='MostlySunny'>
    <text lang='en' translation='Mostly Sunny'/>
    <text lang='de' translation='Überwiegend Sonnig'/>
    <text lang='fr' translation='Assez ensoleillé'/>
    <text lang='it' translation='Prevalentemente soleggiato'/>
</textBlock>
```

Additions can be made by copying a “text block”.

Starting with “<textBlock id=‘keyword‘>” and ending with “</textBlock>”

The texts in the respective languages appear between these elements:

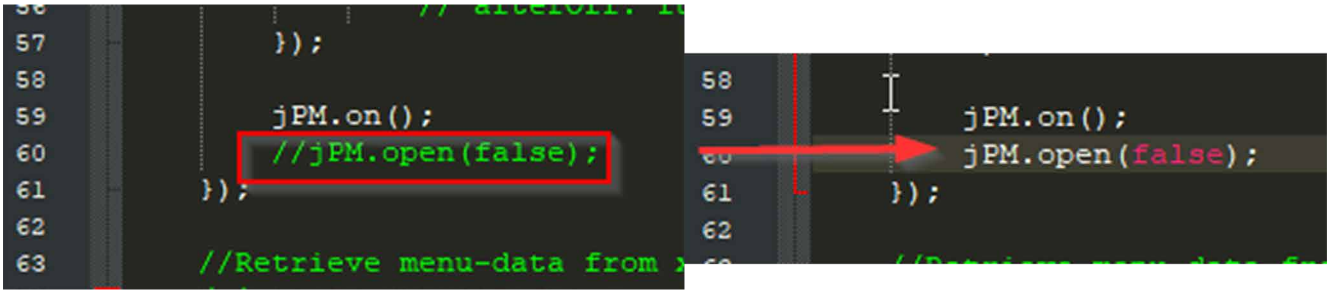
“<text lang=‘language‘ translation=‘text‘/>

10 Special changes to the menu

Various options can be changed under Scripts\createMenu.js.

This is also explained in the relevant section.

If you want the menu to be opened when you open the page in the browser, you must enter “//jPM. open(false);” remove the comment out flag, i.e. the two slashes at the start of the line (see the figure).



```
        });

        jPM.on();
        //jPM.open(false);
    });

    //Retrieve menu-data from x
```

```
        jPM.on();
        jPM.open(false);
    });
```

11. Special changes to colours or to the style

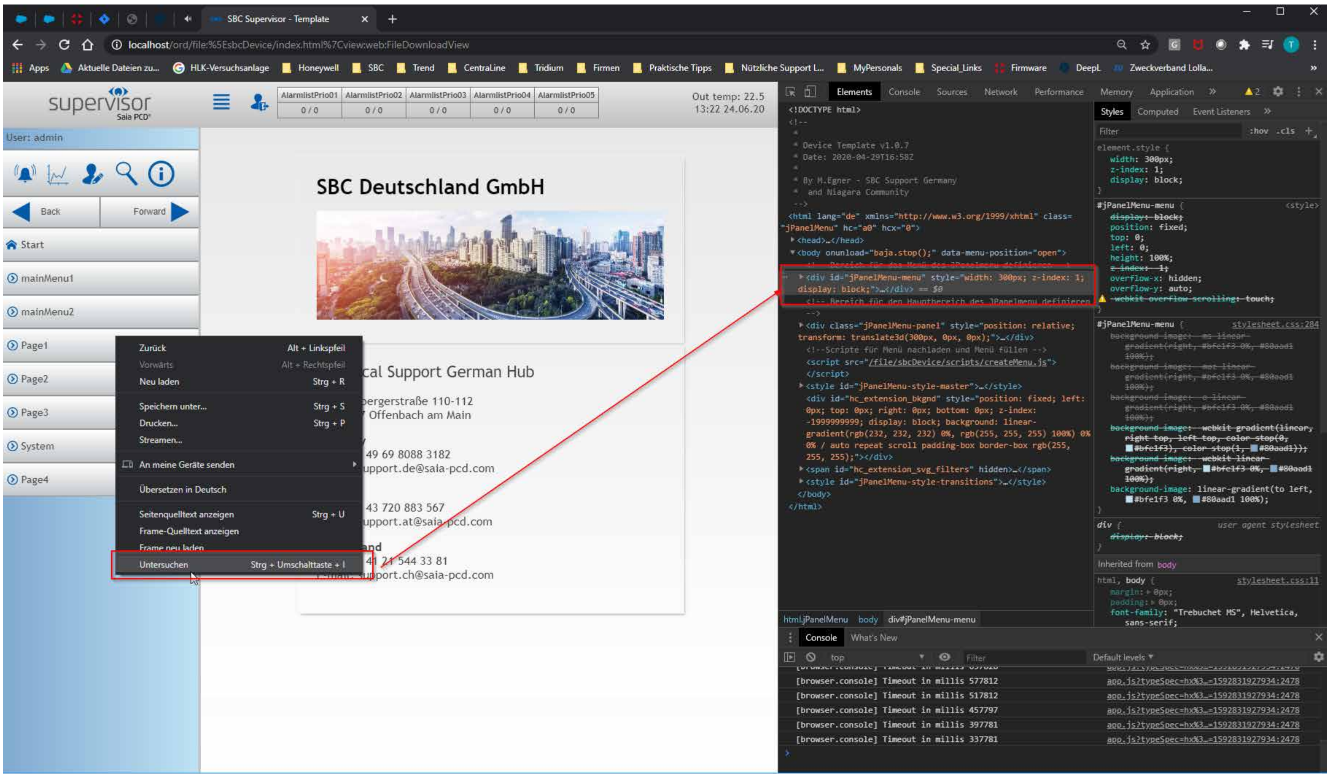
There are ways to change the appearance of the application.

You should have some knowledge of HTML and CSS to begin with or, if necessary, you should familiarise yourself with them.

Explanations for each function can be found on the Internet.

Open your browser and call up the Station template.

Switch to Developer mode (press F12 in Chrome) and go to the Elements tab. If you right click on an element in the menu on the right, you will automatically skip to the corresponding area in the code.

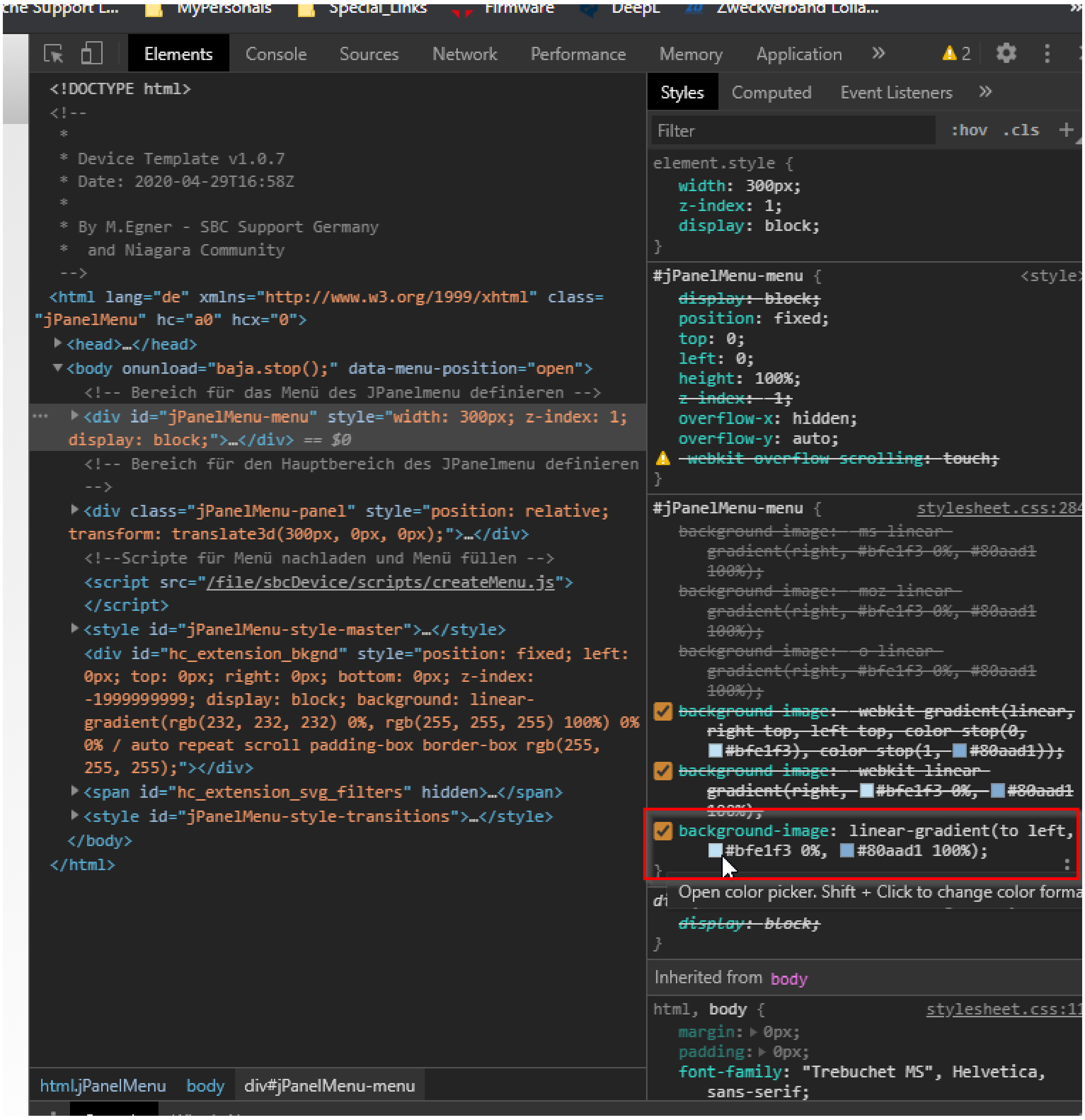


```
<!DOCTYPE html>
<!--
 *
 * Device Template v1.0.7
 * Date: 2020-04-29T16:58Z
 *
 * By M.Egner - SBC Support Germany
 *  and Niagara Community
-->
<html lang="de" xmlns="http://www.w3.org/1999/xhtml" class=
"jPanelMenu" hc="a0" hcx="0">
 ▸<head>…</head>
 ▾<body onunload="baja.stop();" data-menu-position="open">
    <!-- Bereich für das Menü des JPanelmenu definieren -->
  ▸<div id="jPanelMenu-menu" style="width: 300px; z-index: 1;
  display: block;">…</div> == $0
    <!-- Bereich für den Hauptbereich des JPanelmenu definieren
    -->
  ▸<div class="jPanelMenu-panel" style="position: relative;
  transform: translate3d(300px, 0px, 0px);">…</div>
    <!--Scripte für Menü nachladen und Menü füllen -->
    <script src="/file/sbcDevice/scripts/createMenu.js">
    </script>
  ▸<style id="jPanelMenu-style-master">…</style>
    <div id="hc_extension_bkgnd" style="position: fixed; left:
    0px; top: 0px; right: 0px; bottom: 0px; z-index:
    -1999999999; display: block; background: linear-
    gradient(rgb(232, 232, 232) 0%, rgb(255, 255, 255) 100%) 0%
    0% / auto repeat scroll padding-box border-box rgb(255,
    255, 255);"></div>
  ▸<span id="hc_extension_svg_filters" hidden>…</span>
  ▸<style id="jPanelMenu-style-transitions">…</style>
  </body>
</html>
```

```
element.style {
  width: 300px;
  z-index: 1;
  display: block;
}
#jPanelMenu-menu {                                    <style>
  ~~display: block;~~
  position: fixed;
  top: 0;
  left: 0;
  height: 100%;
  ~~z-index: -1;~~
  overflow-x: hidden;
  overflow-y: auto;
⚠ ~~-webkit-overflow-scrolling: touch;~~
}
#jPanelMenu-menu {                         stylesheet.css:284
  ~~background-image: -ms-linear-~~
    ~~gradient(right, #bfe1f3 0%, #80aad1~~
    ~~100%);~~
  ~~background-image: -moz-linear-~~
    ~~gradient(right, #bfe1f3 0%, #80aad1~~
    ~~100%);~~
  ~~background-image: -o-linear-~~
    ~~gradient(right, #bfe1f3 0%, #80aad1~~
    ~~100%);~~
  ~~background-image: -webkit-gradient(linear,~~
    ~~right top, left top, color-stop(0,~~
    ~~#bfe1f3), color-stop(1, #80aad1));~~
  ~~background-image: -webkit-linear-~~
    ~~gradient(right, #bfe1f3 0%, #80aad1~~
    ~~100%);~~
  background-image: linear-gradient(to left,
    #bfe1f3 0%, #80aad1 100%);
}
div {                                  user agent stylesheet
  ~~display: block;~~
}
Inherited from body
html, body {                                stylesheet.css:11
  margin: ▸ 0px;
  padding: ▸ 0px;
  font-family: "Trebuchet MS", Helvetica,
    sans-serif;
```

```
[browser.console] Timeout in millis 577812      app.js?typeSpec=hx%3…=1592831927934:2478
[browser.console] Timeout in millis 517812      app.js?typeSpec=hx%3…=1592831927934:2478
[browser.console] Timeout in millis 457797      app.js?typeSpec=hx%3…=1592831927934:2478
[browser.console] Timeout in millis 397781      app.js?typeSpec=hx%3…=1592831927934:2478
[browser.console] Timeout in millis 337781      app.js?typeSpec=hx%3…=1592831927934:2478
```

You can now adjust the colours directly on the right-hand side by making a colour selection and you will then see the result live:

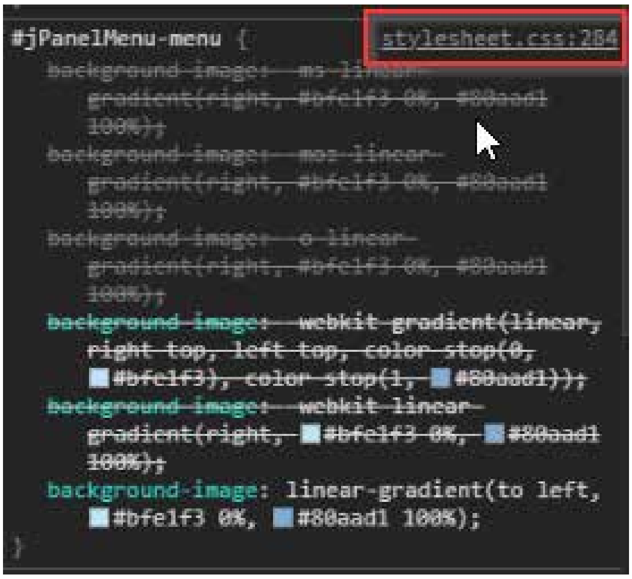

che Support L... MyPersonals Special_Links Firmware DeepL Zweckverband Lolla...
Elements Console Sources Network Performance Memory Application »
⚠ 2
Styles Computed Event Listeners »
Filter :hov .cls +
html.jPanelMenu body div#jPanelMenu-menu

```
<!DOCTYPE html>
<!--
 *
 * Device Template v1.0.7
 * Date: 2020-04-29T16:58Z
 *
 * By M.Egner - SBC Support Germany
 *  and Niagara Community
 -->
<html lang="de" xmlns="http://www.w3.org/1999/xhtml" class=
"jPanelMenu" hc="a0" hcx="0">
 ▶<head>…</head>
 ▼<body onunload="baja.stop();" data-menu-position="open">
    <!-- Bereich für das Menü des JPanelmenu definieren -->
 ···▶<div id="jPanelMenu-menu" style="width: 300px; z-index: 1;
    display: block;">…</div> == $0
    <!-- Bereich für den Hauptbereich des JPanelmenu definieren
    -->
   ▶<div class="jPanelMenu-panel" style="position: relative;
    transform: translate3d(300px, 0px, 0px);">…</div>
    <!--Scripte für Menü nachladen und Menü füllen -->
    <script src="/file/sbcDevice/scripts/createMenu.js">
    </script>
   ▶<style id="jPanelMenu-style-master">…</style>
    <div id="hc_extension_bkgnd" style="position: fixed; left:
    0px; top: 0px; right: 0px; bottom: 0px; z-index:
    -1999999999; display: block; background: linear-
    gradient(rgb(232, 232, 232) 0%, rgb(255, 255, 255) 100%) 0%
    0% / auto repeat scroll padding-box border-box rgb(255,
    255, 255);"></div>
   ▶<span id="hc_extension_svg_filters" hidden>…</span>
   ▶<style id="jPanelMenu-style-transitions">…</style>
  </body>
</html>
```

```
element.style {
  width: 300px;
  z-index: 1;
  display: block;
}
#jPanelMenu-menu {                          <style>
  ~~display: block;~~
  position: fixed;
  top: 0;
  left: 0;
  height: 100%;
  ~~z-index: 1;~~
  overflow-x: hidden;
  overflow-y: auto;
⚠ ~~-webkit-overflow-scrolling: touch;~~
}
#jPanelMenu-menu {                stylesheet.css:284
  ~~background-image: -ms-linear-
     gradient(right, #bfe1f3 0%, #80aad1
     100%);~~
  ~~background-image: -moz-linear-
     gradient(right, #bfe1f3 0%, #80aad1
     100%);~~
  ~~background-image: -o-linear-
     gradient(right, #bfe1f3 0%, #80aad1
     100%);~~
☑ ~~background-image: -webkit-gradient(linear,
     right top, left top, color-stop(0,
     #bfe1f3), color-stop(1, #80aad1));~~
☑ ~~background-image: -webkit-linear-
     gradient(right, #bfe1f3 0%, #80aad1
     100%);~~
☑ background-image: linear-gradient(to left,
     #bfe1f3 0%, #80aad1 100%);
}
 Open color picker. Shift + Click to change color forma
di
  ~~display: block;~~
}
Inherited from body
html, body {                       stylesheet.css:11
  margin: ▸ 0px;
  padding: ▸ 0px;
  font-family: "Trebuchet MS", Helvetica,
     sans-serif;
```

Simply enter the result in stylesheet.css and save it.

The location of the entry is shown here:

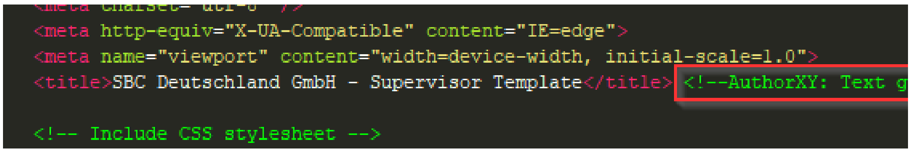


```
#jPanelMenu-menu {                stylesheet.css:284
  ~~background-image: -ms-linear-
     gradient(right, #bfe1f3 0%, #80aad1
     100%);~~
  ~~background-image: -moz-linear-
     gradient(right, #bfe1f3 0%, #80aad1
     100%);~~
  ~~background-image: -o-linear-
     gradient(right, #bfe1f3 0%, #80aad1
     100%);~~
  ~~background-image: -webkit-gradient(linear,
     right top, left top, color-stop(0,
     #bfe1f3), color-stop(1, #80aad1));~~
  ~~background-image: -webkit-linear-
     gradient(right, #bfe1f3 0%, #80aad1
     100%);~~
  background-image: linear-gradient(to left,
     #bfe1f3 0%, #80aad1 100%);
}
```

11.1. Mark changes

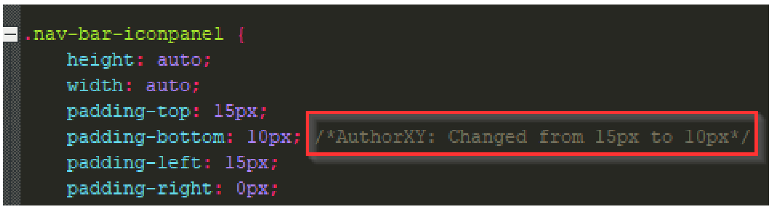
To make changes easier to find again later, it is advisable to add comments to them in the files.

Proceed as follows in .html, .xml and .nav files: “<!--Author: Comment -->”



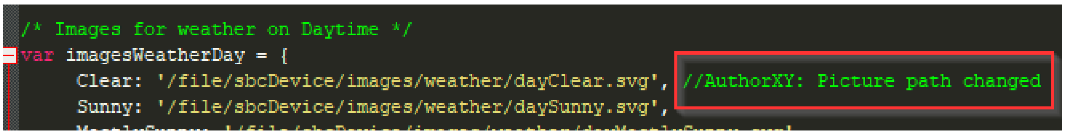
```
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>SBC Deutschland GmbH - Supervisor Template</title> <!--AuthorXY: Text g

<!-- Include CSS stylesheet -->
```

Proceed as follows in .css files: /*Author: Comment */



```
.nav-bar-iconpanel {
    height: auto;
    width: auto;
    padding-top: 15px;
    padding-bottom: 10px; /*AuthorXY: Changed from 15px to 10px*/
    padding-left: 15px;
    padding-right: 0px;
```

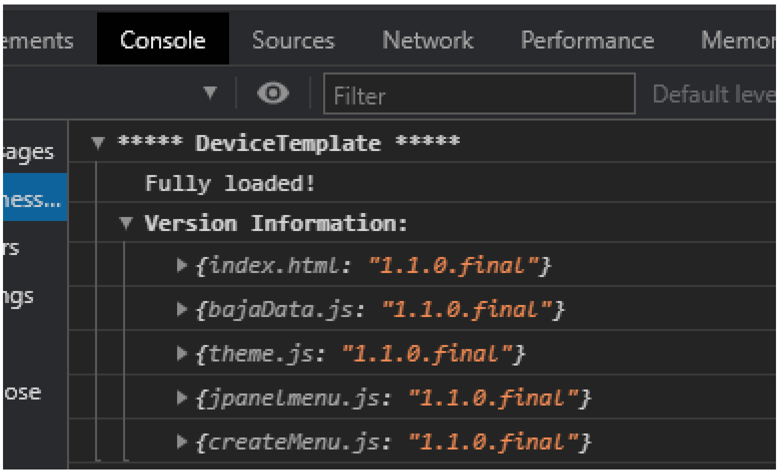
Proceed as follows in .js files: // Author: Comment



```
/* Images for weather on Daytime */
var imagesWeatherDay = {
     Clear: '/file/sbcDevice/images/weather/dayClear.svg', //AuthorXY: Picture path changed
     Sunny: '/file/sbcDevice/images/weather/daySunny.svg',
     MostlySunny: '/file/sbcDevice/images/weather/dayMostlySunny.svg',
```

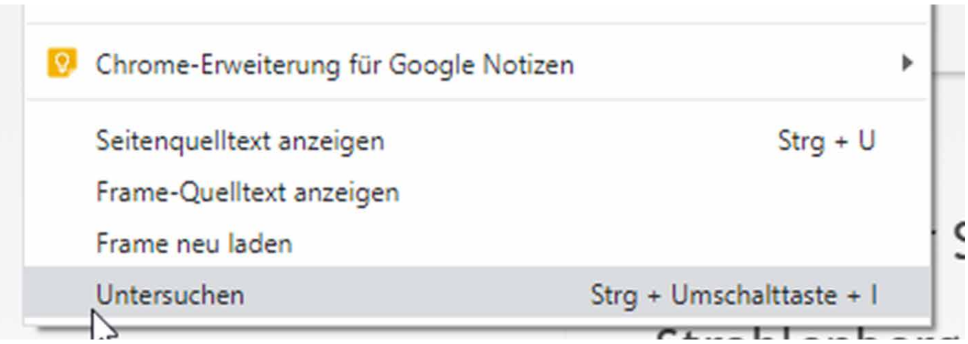
12. Debugging

In the event of problems, it is particularly important to get a dump/screenshot from the console for support purposes. Any errors may be seen here, but also notes on the version used (only from v1.1.0).

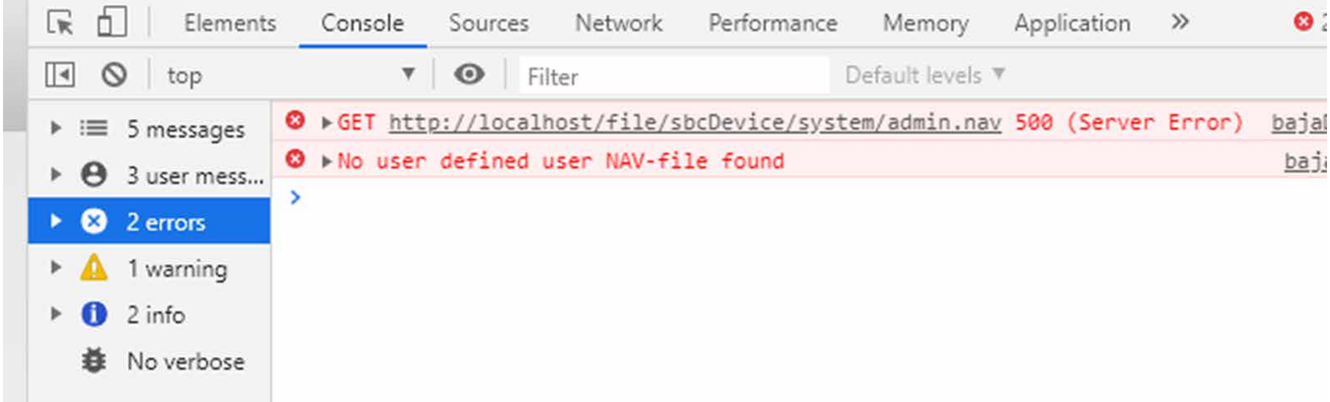


```
▼ ***** DeviceTemplate *****
    Fully loaded!
  ▼ Version Information:
      ▶{index.html: "1.1.0.final"}
      ▶{bajaData.js: "1.1.0.final"}
      ▶{theme.js: "1.1.0.final"}
      ▶{jpanelmenu.js: "1.1.0.final"}
      ▶{createMenu.js: "1.1.0.final"}
```

For example, you can use the console for debugging in Chrome:

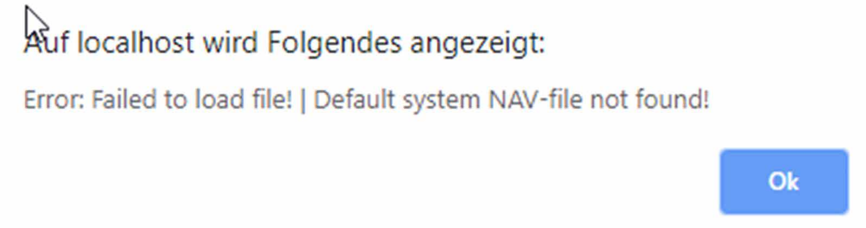

Chrome-Erweiterung für Google Notizen
Seitenquelltext anzeigen Strg + U
Frame-Quelltext anzeigen
Frame neu laden
Untersuchen Strg + Umschalttaste + I

That way you will get a reasonably clear error message if a problem is detected:



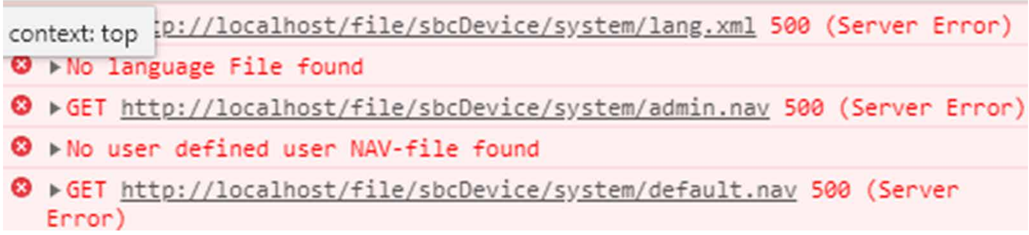
```
⊗ ▶GET http://localhost/file/sbcDevice/system/admin.nav 500 (Server Error)   bajaD
⊗ ▶No user defined user NAV-file found                                          baja
>
```

If no nav files or default files are found, then a popup appears:



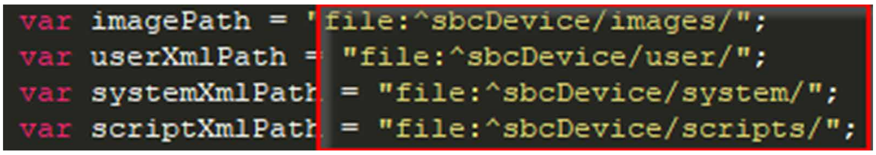
Auf localhost wird Folgendes angezeigt:

Error: Failed to load file! | Default system NAV-file not found!



```
context: top tp://localhost/file/sbcDevice/system/lang.xml 500 (Server Error)
⊗ ▶No language File found
⊗ ▶GET http://localhost/file/sbcDevice/system/admin.nav 500 (Server Error)
⊗ ▶No user defined user NAV-file found
⊗ ▶GET http://localhost/file/sbcDevice/system/default.nav 500 (Server
  Error)
```

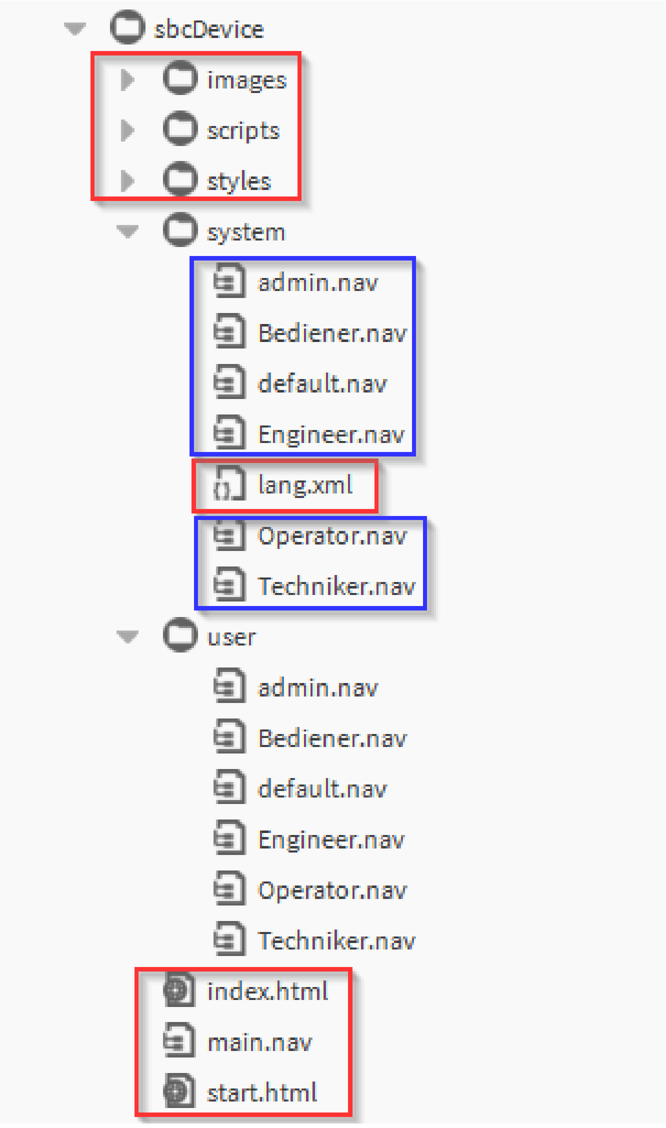
If a folder is renamed, the paths in the scripts/theme.js must be adapted!



```
var imagePath = "file:^sbcDevice/images/";
var userXmlPath = "file:^sbcDevice/user/";
var systemXmlPath = "file:^sbcDevice/system/";
var scriptXmlPath = "file:^sbcDevice/scripts/";
```

13. Update of the DeviceTemplate from an older version

13.1. Updating version v0.9.6 and older



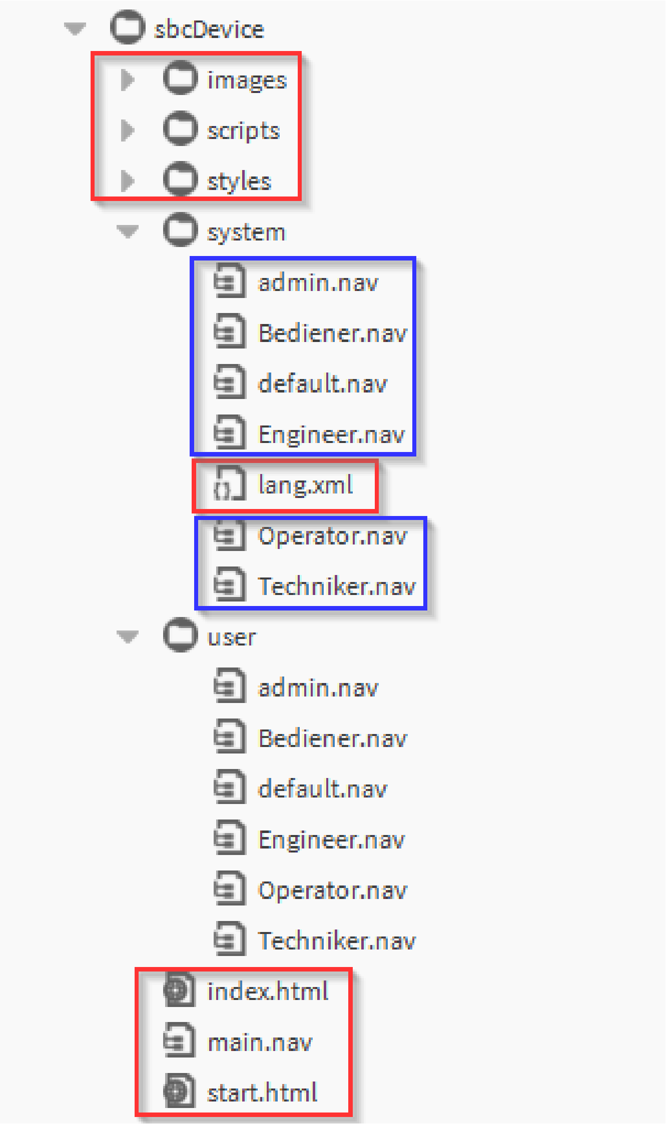
- sbcDevice
  - images
  - scripts
  - styles
  - system
    - admin.nav
    - Bediener.nav
    - default.nav
    - Engineer.nav
    - lang.xml
    - Operator.nav
    - Techniker.nav
  - user
    - admin.nav
    - Bediener.nav
    - default.nav
    - Engineer.nav
    - Operator.nav
    - Techniker.nav
  - index.html
  - main.nav
  - start.html

For an update, all files and folders, with the exception of the “Nav files” under “system” and “user”, are exchanged or overwritten (marked in red).
All files marked in blue no longer work with this version!
Changes to these files must be compared with the old file and transferred as necessary.

!	NOTE:	The update only affects the content of the “sbcDevice” folder! No other files may be overwritten!!

If special changes were made within the files marked in red, then these must be transferred to the new files.
In this case it is advisable to mark these changes with a comment so they can be found again later (see the end of chapter 11.1).

13.2. Update from version v0.9.7 to v1.0.8 to v1.1.0 or higher

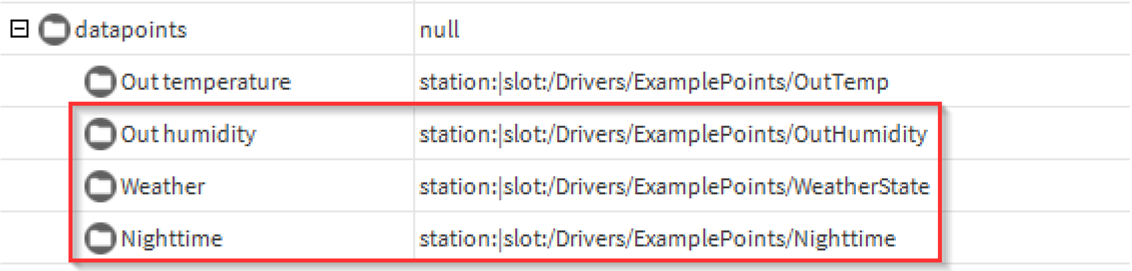


```
sbcDevice
    images
    scripts
    styles
    system
        admin.nav
        Bediener.nav
        default.nav
        Engineer.nav
        lang.xml
        Operator.nav
        Techniker.nav
    user
        admin.nav
        Bediener.nav
        default.nav
        Engineer.nav
        Operator.nav
        Techniker.nav
    index.html
    main.nav
    start.html
```

For an update, all files and folders, with the exception of the “Nav files” under “system” and “user”, are exchanged or overwritten (marked in red).
All files marked in blue will continue to work (downward compatible), but there are new sub-items in the NAV file under “Datapoints” for outdoor humidity and weather.

!	NB:	The update only affects the content of the “sbcDevice” folder! No other files may be overwritten!!

If these functions are required, these entries must be added or expanded.
These items allow you to incorporate outdoor humidity or a weather service in addition to the outdoor temperature.



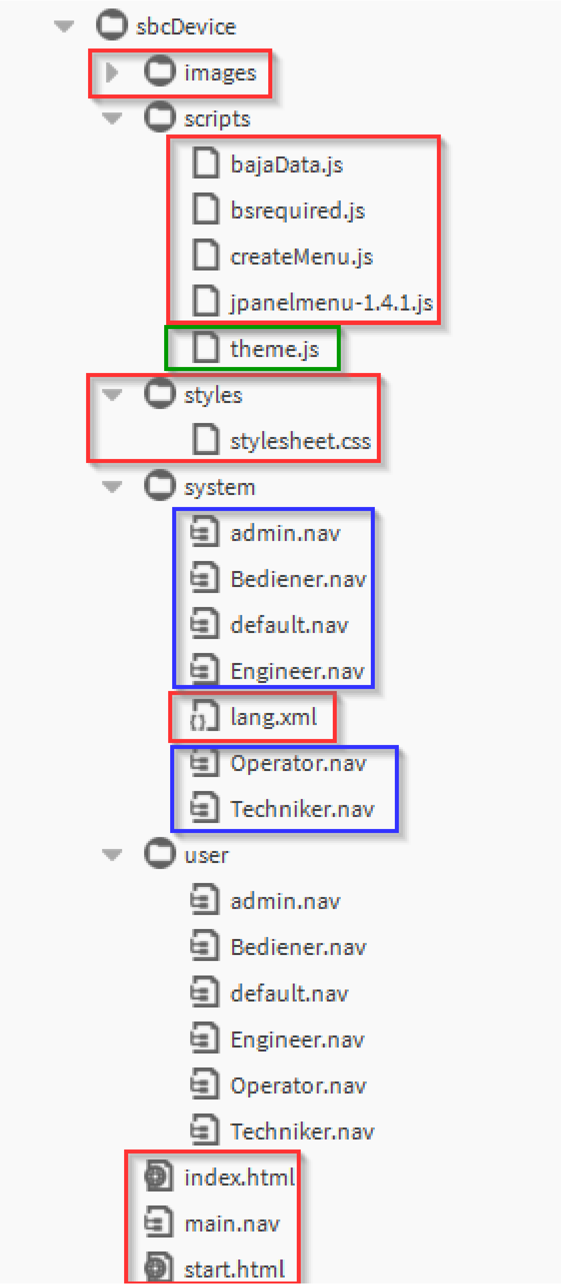
datapoints	null
Out temperature	station:|slot:/Drivers/ExamplePoints/OutTemp
Out humidity	station:|slot:/Drivers/ExamplePoints/OutHumidity
Weather	station:|slot:/Drivers/ExamplePoints/WeatherState
Nighttime	station:|slot:/Drivers/ExamplePoints/Nighttime

If special changes were made within the files marked in red, then these must be transferred to the new files.
In this case it is advisable to mark these changes with a comment so they can be found again later (see the end of chapter 11.1).

13.3. Update from version v1.0.9 to v1.1.0 or higher

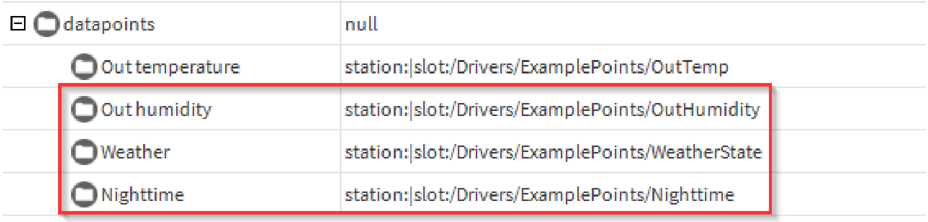
For an update, all files and folders, with the exception of the “Nav files” under “system” and “user” and the “theme.js” file under “scripts” are exchanged or overwritten (marked in red).
All files marked in blue will continue to work, but there are new sub-items in the NAV file under “Datapoints” for outdoor humidity and weather.

!	NOTE:	The update only affects the content of the “sbcDevice” folder! No other files may be overwritten!!



```
sbcDevice
  images
  scripts
    bajaData.js
    bsrequired.js
    createMenu.js
    jpanelmenu-1.4.1.js
    theme.js
  styles
    stylesheet.css
  system
    admin.nav
    Bediener.nav
    default.nav
    Engineer.nav
    lang.xml
    Operator.nav
    Techniker.nav
  user
    admin.nav
    Bediener.nav
    default.nav
    Engineer.nav
    Operator.nav
    Techniker.nav
  index.html
  main.nav
  start.html
```

If these functions are required, these entries must be added or expanded.
These items allow you to incorporate outdoor humidity or a weather service in addition to the outdoor temperature.



⊟ datapoints	null
Out temperature	station:|slot:/Drivers/ExamplePoints/OutTemp
Out humidity	station:|slot:/Drivers/ExamplePoints/OutHumidity
Weather	station:|slot:/Drivers/ExamplePoints/WeatherState
Nighttime	station:|slot:/Drivers/ExamplePoints/Nighttime

Changes to the “scripts/theme.js” file (marked in green) may need to be transferred to the new file.
If there have been no changes, this can be overwritten directly.
If special changes were made within the files marked in red, then these must be transferred to the new files.
In this case it is advisable to mark these changes with a comment so they can be found again later (see the end of chapter 11.1).

Saia-Burgess Controls AG
Bahnhofstrasse 18 | 3280 Murten, Switzerland
T +41 26 580 30 00 | F +41 26 580 34 99
www.saia-pcd.com

support@saia-pcd.com | www.sbc-support.com

2021-04-26 | Technical data and information subject to change.