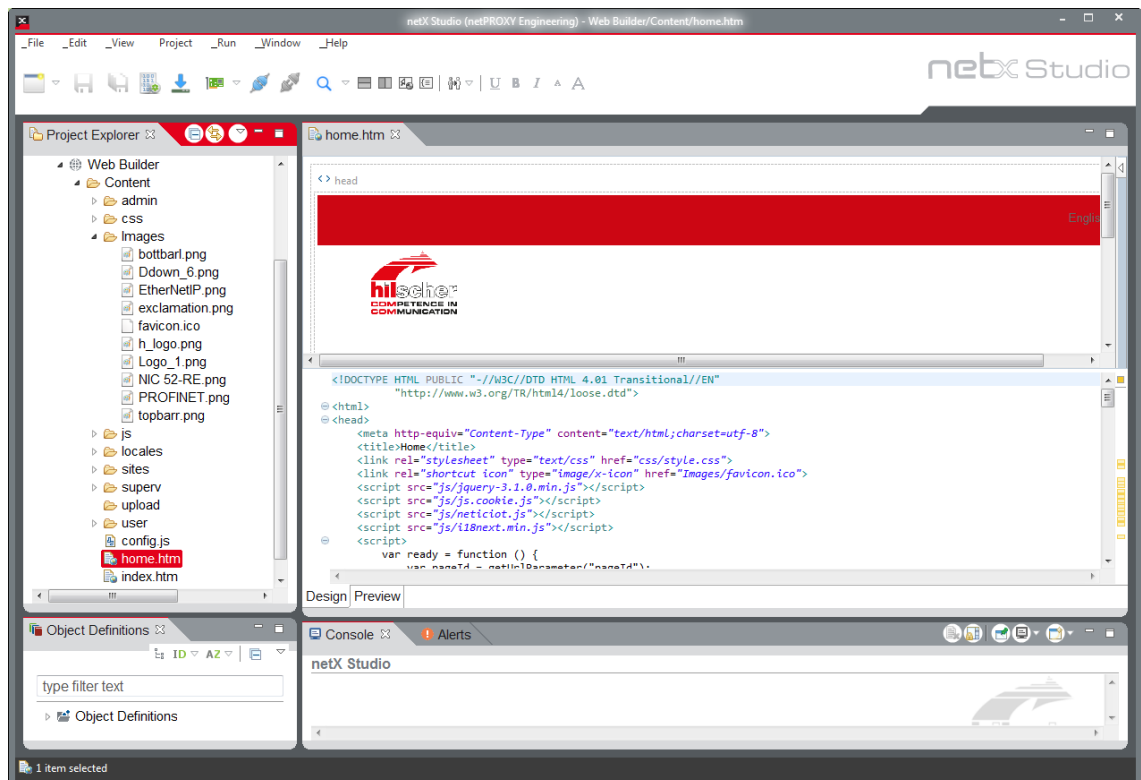


Application note
netX Studio Engineering Tool
Web Builder



Hilscher Gesellschaft für Systemautomation mbH
www.hilscher.com

DOC160207AN01EN | Revision 1 | English | 2016-12 | Released | Public

Table of contents

1	Introduction	3
1.1	About this Document	3
1.1.1	Description of the contents	3
1.1.2	List of revisions	3
1.1.3	Conventions in this document.....	4
1.2	Legal notes.....	5
1.2.1	Copyright.....	5
1.2.2	Important notes	5
1.2.3	Exclusion of liability	6
1.2.4	Warranty	6
1.2.5	Export regulations	7
2	Description of Web Builder for servX WebServer	8
3	Examples of adding own content.....	10
3.1	Overview	10
3.2	Substituting company logo and adding hyperlink to own company website	12
3.3	Changing page layout in Cascading Style Sheet (CSS)	17
3.4	Changing text in HTML page	21
4	Using Product Macros to customize web pages of product variants.....	23
4.1	Overview	23
4.2	Application example of Product Macros	27
5	Checking web content locally in browser before download.....	34
6	Access rights for web pages	37
7	Integrating netPROXY objects in web pages	40
	List of Figures	45
	List of Tables.....	47
	Contacts.....	48

1 Introduction

1.1 About this Document

1.1.1 Description of the contents

This document describes functions and application of the **Web Builder**, which is a tool for designing and managing HTML web pages for the **servX WebServer** in netIC IOT communication modules.

Here you will find step-by-step instructions for typical use cases like adapting standard layouts and customizing the content of the web pages. This manual also explains the management of access rights and how to integrate netPROXY object elements into a web page.

The **Web Builder** is part of the Hilscher **netX Studio Engineering Tool**. Please note that this Application Note does not describe how to use the netX Studio Engineering Tool itself; fundamental information about how to operate the Engineering Tool is provided in a separate operating instruction manual called *netX Studio Engineering Tool - Device development*, DOC160103OlxxEN.

1.1.2 List of revisions

Index	Date	Change
1	2016-12-07	Created

Table 1: List of revisions

1.1.3 Conventions in this document

Notes, operation instructions and results of operation steps are marked as follows:

Notes



Note:

<important note>



Note:

<simple note>



Folgen

<note, where to find further information>

Operation instructions

1. <operational step>

➤ <instruction>

➤ <instruction>

2. <operational step>

➤ <instruction>

➤ <instruction>

Results

↻ <intermediate result>

⇒ <final result>

1.2 Legal notes

1.2.1 Copyright

© Hilscher Gesellschaft für Systemautomation mbH

All rights reserved.

The images, photographs and texts in the accompanying material (user manual, accompanying texts, documentation, etc.) are protected by German and international copyright law as well as international trade and protection provisions. You are not authorized to duplicate these in whole or in part using technical or mechanical methods (printing, photocopying or other methods), to manipulate or transfer using electronic systems without prior written consent. You are not permitted to make changes to copyright notices, markings, trademarks or ownership declarations. The included diagrams do not take the patent situation into account. The company names and product descriptions included in this document may be trademarks or brands of the respective owners and may be trademarked or patented. Any form of further use requires the explicit consent of the respective rights owner.

1.2.2 Important notes

The user manual, accompanying texts and the documentation were created for the use of the products by qualified experts, however, errors cannot be ruled out. For this reason, no guarantee can be made and neither juristic responsibility for erroneous information nor any liability can be assumed. Descriptions, accompanying texts and documentation included in the user manual do not present a guarantee nor any information about proper use as stipulated in the contract or a warranted feature. It cannot be ruled out that the user manual, the accompanying texts and the documentation do not correspond exactly to the described features, standards or other data of the delivered product. No warranty or guarantee regarding the correctness or accuracy of the information is assumed.

We reserve the right to change our products and their specification as well as related user manuals, accompanying texts and documentation at all times and without advance notice, without obligation to report the change. Changes will be included in future manuals and do not constitute any obligations. There is no entitlement to revisions of delivered documents. The manual delivered with the product applies.

Hilscher Gesellschaft für Systemautomation mbH is not liable under any circumstances for direct, indirect, incidental or follow-on damage or loss of earnings resulting from the use of the information contained in this publication.

1.2.3 Exclusion of liability

The software was produced and tested with utmost care by Hilscher Gesellschaft für Systemautomation mbH and is made available as is. No warranty can be assumed for the performance and flawlessness of the software for all usage conditions and cases and for the results produced when utilized by the user. Liability for any damages that may result from the use of the hardware or software or related documents, is limited to cases of intent or grossly negligent violation of significant contractual obligations. Indemnity claims for the violation of significant contractual obligations are limited to damages that are foreseeable and typical for this type of contract.

It is strictly prohibited to use the software in the following areas:

- for military purposes or in weapon systems;
- for the design, construction, maintenance or operation of nuclear facilities;
- in air traffic control systems, air traffic or air traffic communication systems;
- in life support systems;
- in systems in which failures in the software could lead to personal injury or injuries leading to death.

We inform you that the software was not developed for use in dangerous environments requiring fail-proof control mechanisms. Use of the software in such an environment occurs at your own risk. No liability is assumed for damages or losses due to unauthorized use.

1.2.4 Warranty

Although the hardware and software was developed with utmost care and tested intensively, Hilscher Gesellschaft für Systemautomation mbH does not guarantee its suitability for any purpose not confirmed in writing. It cannot be guaranteed that the hardware and software will meet your requirements, that the use of the software operates without interruption and that the software is free of errors. No guarantee is made regarding infringements, violations of patents, rights of ownership or the freedom from interference by third parties. No additional guarantees or assurances are made regarding marketability, freedom of defect of title, integration or usability for certain purposes unless they are required in accordance with the law and cannot be limited. Warranty claims are limited to the right to claim rectification.

1.2.5 Export regulations

The delivered product (including the technical data) is subject to export or import laws as well as the associated regulations of different countries, in particular those of Germany and the USA. The software may not be exported to countries where this is prohibited by the United States Export Administration Act and its additional provisions. You are obligated to comply with the regulations at your personal responsibility. We wish to inform you that you may require permission from state authorities to export, re-export or import the product.

2 Description of Web Builder for servX WebServer

The purpose of the integrated **Web Builder** of the **netX Studio Engineering Tool** is the design and management of web content (HTML pages, CSS, Java Script and image files) for the servX WebServer. The servX WebServer is part of firmware for Hilscher devices like e. g. the netIC IOT DIL-32 communication module. It allows you to access an automation device in the field via its Ethernet interface by standard web browser and HTTP.

Parameters and IO data of devices containing firmware with integrated servX WebServer and netPROXY Server configured by the netX Studio Engineering Tool can be read and written via HTTP.

Standard functions of the WebServer are:

- Displaying general device information like e. g. version, serial number, MAC address, maintenance notes (Home)
- Configuring integrated MQTT client (Configuration > MQTT Client) and OPC-UA server (Configuration > OPC UA Server) [if enabled]
- Diagnosis of IP settings (Diagnosis > IP-Settings), network status (Diagnosis > Communication) and Ethernet connection (Diagnosis > Ethernet)
- Reading and writing of device parameter and netPROXY objects (Diagnosis > Object Data)
- Downloading device description file from the device (Download)
- User management (Administration > User/Password Management)
- Resetting device (Administration > Reset Device)

Functions and design of the web pages running on the WebServer can be edited and managed by the **Web Builder**.

Within netX Studio, the **Web Builder** serves as a kind of content management system for the WebServer pages. By using the "product macros" of netX Studio, you can assign customized web pages and graphics (e. g. containing individual brand names and pictures) to different variants of your product/device. After the build process, customized web content can be downloaded to individual target devices along with other data that has been created/managed in netX Studio (like e. g. firmware, configuration data, device and manufacturer IDs etc.)

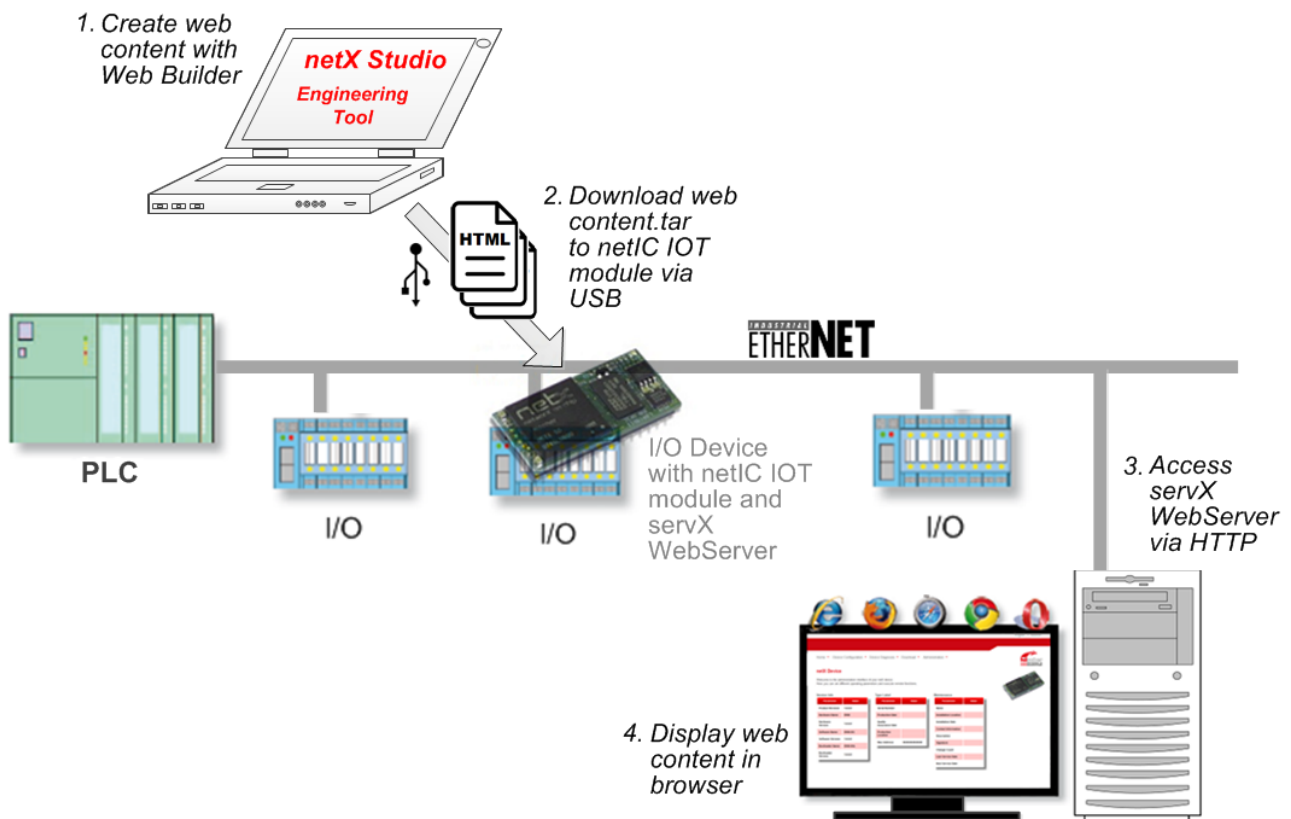


Figure 1: Using Web Builder to create web content for servX WebServer

3 Examples of adding own content

3.1 Overview

This section provides step-by-step instructions for typical customizations you might want to make for the standard content provided by the Web Builder. These are:

- Substituting company logo and adding hyperlink to internet homepage
- Changing welcome text on start page
- Changing background design of the navigation pane, resizing standard text-font and changing color of headings in cascading style sheet (CSS)

Start page (home page) before customization:

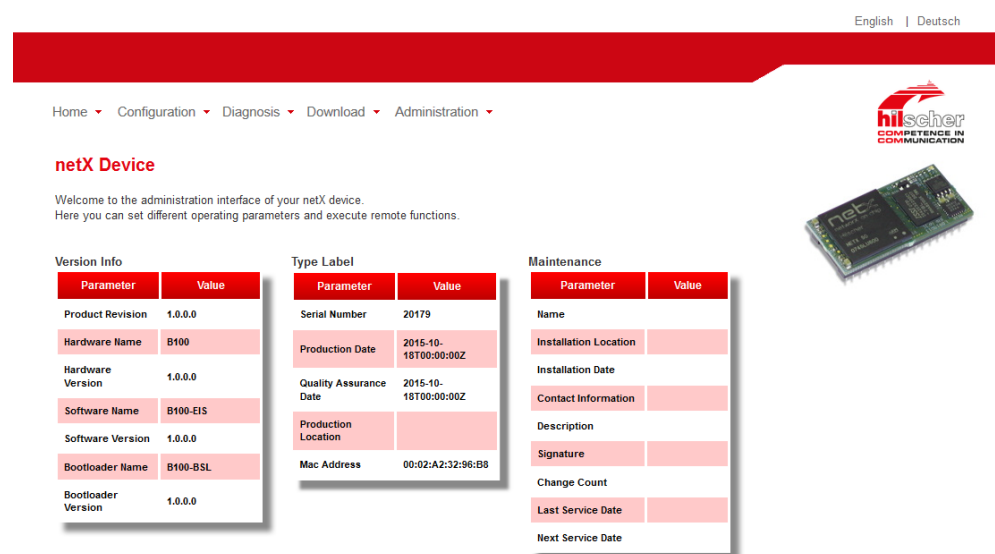


Figure 2: Start page without customization

Start page after customization:


Home ▾ Configuration ▾ Diagnosis ▾ Download ▾ Administration ▾

English | Deutsch

MUSTERMANN
AUTOMATION
★ ★ ★

B100-EIS

Welcome to the web configuration interface of the B100-EIS module for EtherNet/IP communication.
Here you can set different operating parameters and execute remote functions.



Version Info

Parameter	Value
Product Revision	1.0.0.0
Hardware Name	B100
Hardware Version	1.0.0.0
Software Name	B100-EIS
Software Version	1.0.0.0
Bootloader Name	B100-BSL
Bootloader Version	1.0.0.0

Type Label

Parameter	Value
Serial Number	20179
Production Date	2015-10-18T00:00:00Z
Quality Assurance Date	2015-10-18T00:00:00Z
Production Location	
Mac Address	00:02:A2:32:96:B8

Maintenance

Parameter	Value
Name	
Installation Location	
Installation Date	
Contact Information	
Description	
Signature	
Change Count	
Last Service Date	
Next Service Date	

Figure 3: Start page after customization



Note:
Please note that this chapter provides only exemplary instructions and that there are – depending on your skills in web design – other possible ways of proceeding.

3.2 Substituting company logo and adding hyperlink to own company website

By default, all web pages generated by the **Web Builder** carry the Hilscher logo. This logo is referenced in the **home.htm** HTML file, which contains or references all standard elements that shall be displayed on every web page, like e. g. the navigation menu, footer and header texts and said company logo.

This section shows how you can substitute the Hilscher logo with your own company logo (if available as graphic file) and how to insert a hyperlink leading to a web page in the world wide web – in this example a fictive page which could be your own company's homepage.

1. Open project

- Start the **netX Studio Engineering Tool** and open the project which you want to edit its web pages (in this example, this is the **Series B** project).
- In the **Project Explorer**, scroll down to the **Web Builder** object.

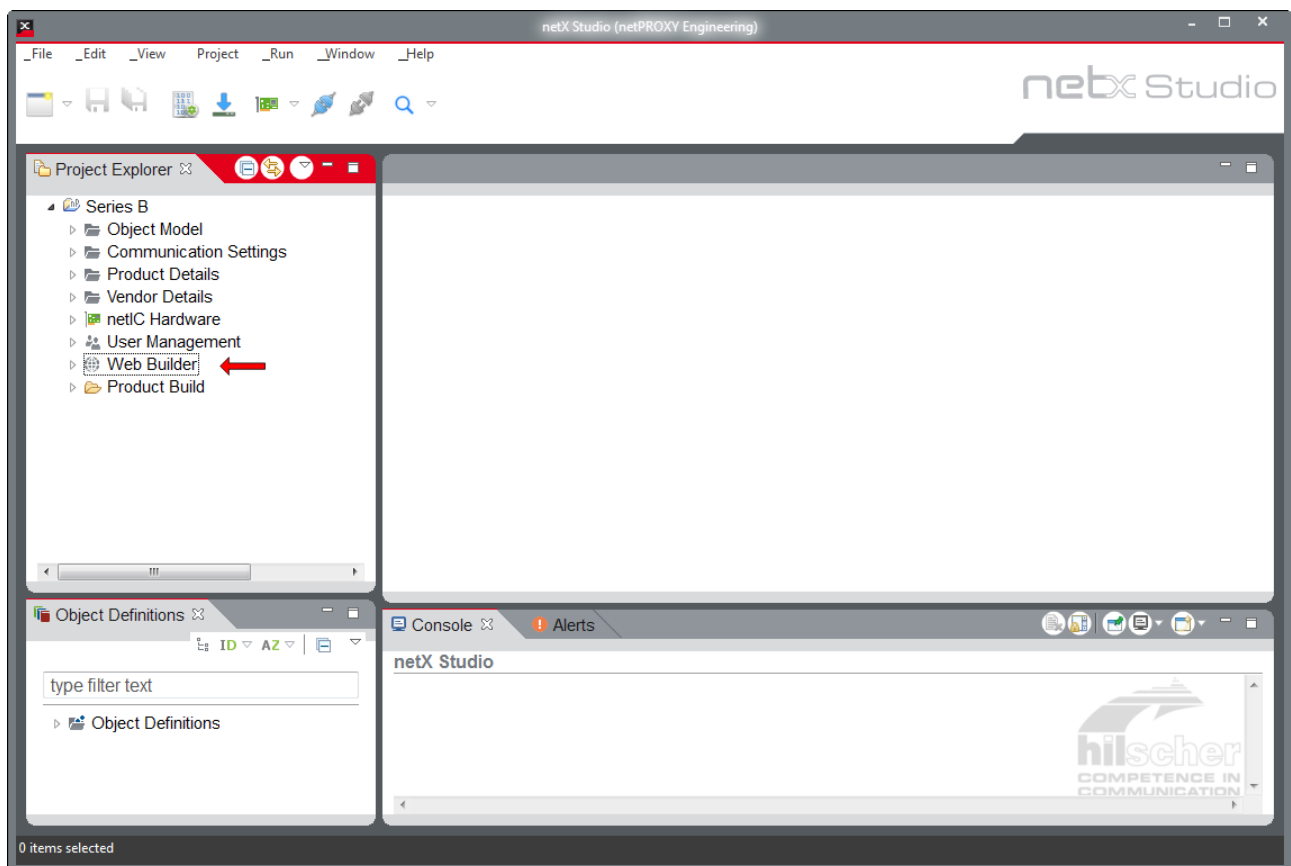


Figure 4: Web Builder in opened project

2. Import graphic file containing own logo.
 - In the **Project Explorer**, select folder **Web Builder > Content > Images**.
 - Choose **Add Files** from the context menu.

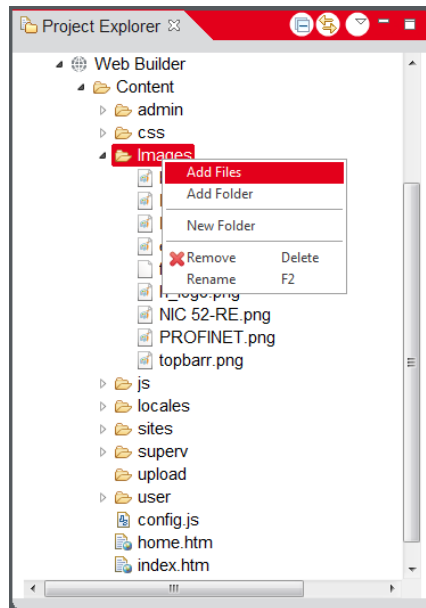


Figure 5: Add file

- The Windows file selection dialog opens.
- In your file system, choose the graphic file containing your company logo then click **Open**.
- The graphic file is imported and added to the **Images** folder. In this example this is the **Logo_1.png** file.

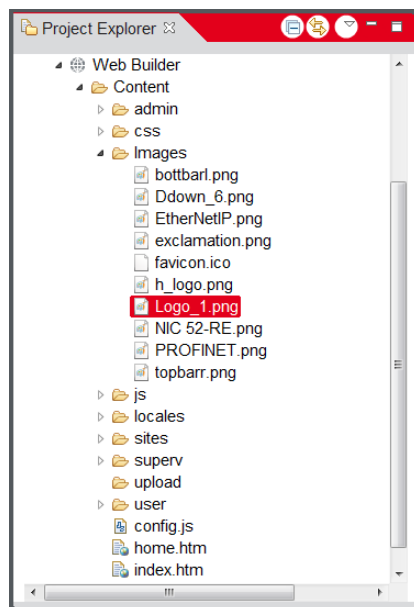


Figure 6: New logo graphic file in Project Explorer

3. Open HTML file.

- In folder **Web Builder > Content**, double-click **home.htm** file.
- The file is opened in the editor. In case the netX Studio GUI has been set to default layout (this can be done in the **Window > Default Layout** menu), the editor is split into an upper (**Design Page**) for layout preview and a lower **Source Page** for editing the actual source code:

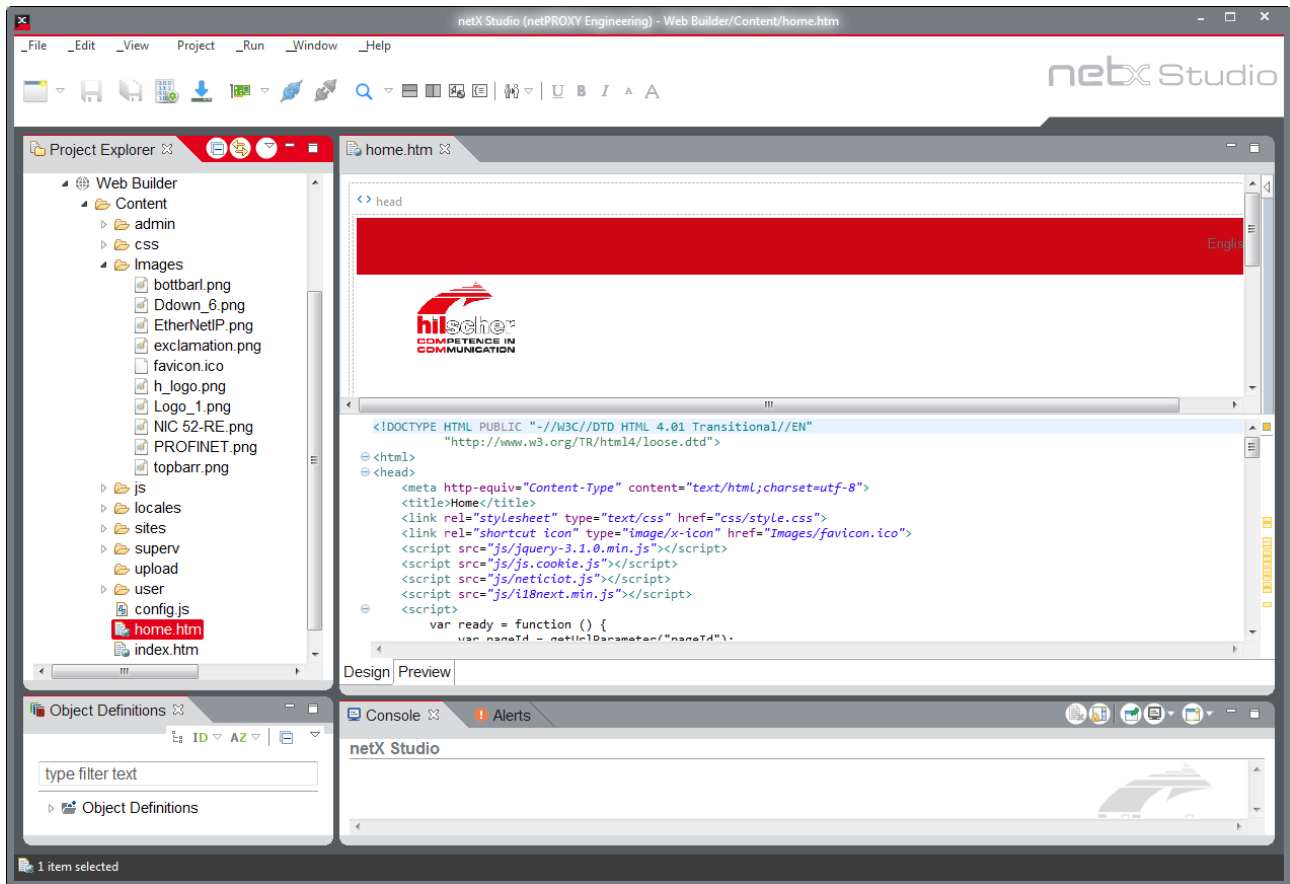


Figure 7: home.htm page in editor

4. Add hyperlink to own company website.

- In the Source Page (lower part of the editor window), scroll to the element which references the logo and the hyperlink. To avoid laboriously searching the source code, you can alternatively click on the wanted object (in this case the Hilscher logo) in the Design Page in the upper part of the editor. The Source Page then automatically jumps to the right section and highlights the corresponding source code element:

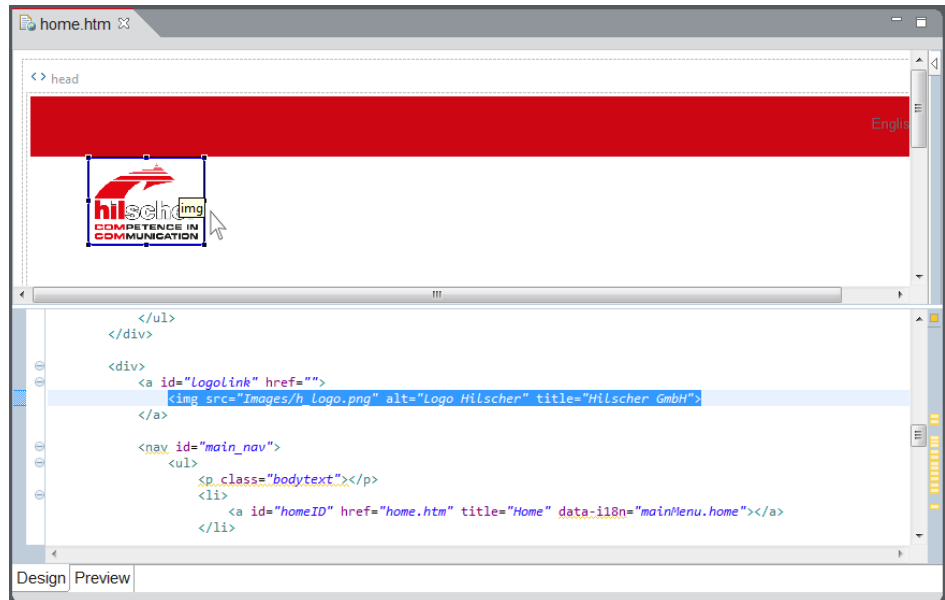


Figure 8: Auto focus on source code element of Logo

- In the `` hyperlink tag, insert your web address (e. g. `"http://www.mustermann-automation.com"`) into the empty quotation marks behind the `href=""` attribute.



Figure 9: Adding own web address

5. Reference own graphic file.

- In the `` tag, replace the name of the old graphics file `h_logo.png` with the name of the new graphics file which you have imported in step 1 (i. e. `Logo_1`).
- Replace the alternative text `alt="Logo Hilscher"` with the text that should be displayed in case the image cannot be displayed by the browser (e. g. `"Logo Mustermann Automation"`).

- In the `title="Hilscher GmbH"` title attribute, replace `"Hilscher GmbH"` with the text that should be displayed as "tooltip" when the user of the web page hovers over the image/logo with his/her mouse (e. g. `"Go to Mustermann website"`).
- ⇒ The element containing the hyperlink and the logo should now look as follows (also, the new logo should now already be displayed in the layout preview in the upper part of the editor):



Figure 10: Changed image attributes

- In the menu, choose **File > Save** or press **Ctrl + S** on your keyboard to save the changes to the **home.htm** file.
- ⇒ You have replaced the Hilscher logo with your own logo and added hyperlink to your company website.

3.3 Changing page layout in Cascading Style Sheet (CSS)

Layout and design of the web pages generated by the **Web Builder** are by default determined by a Cascading Style Sheet named `style.css`.

The CSS file defines e. g. framing, spacing, text formats (font, size, color) and designs of headers and footers etc.

This sections shows how to change the design of the header by removing the background image, assigning gray color and moving the menu pane a bit upwards. This section also shows how to enlarge the standard font-size from 10 pt to 12 pt and how to change the color of first-level headings (H1) from red to blue.

1. Open CSS file.

➤ In the **Web Builder > Content > css** folder, double-click **style.css** file.

➤ The file is opened in the editor.

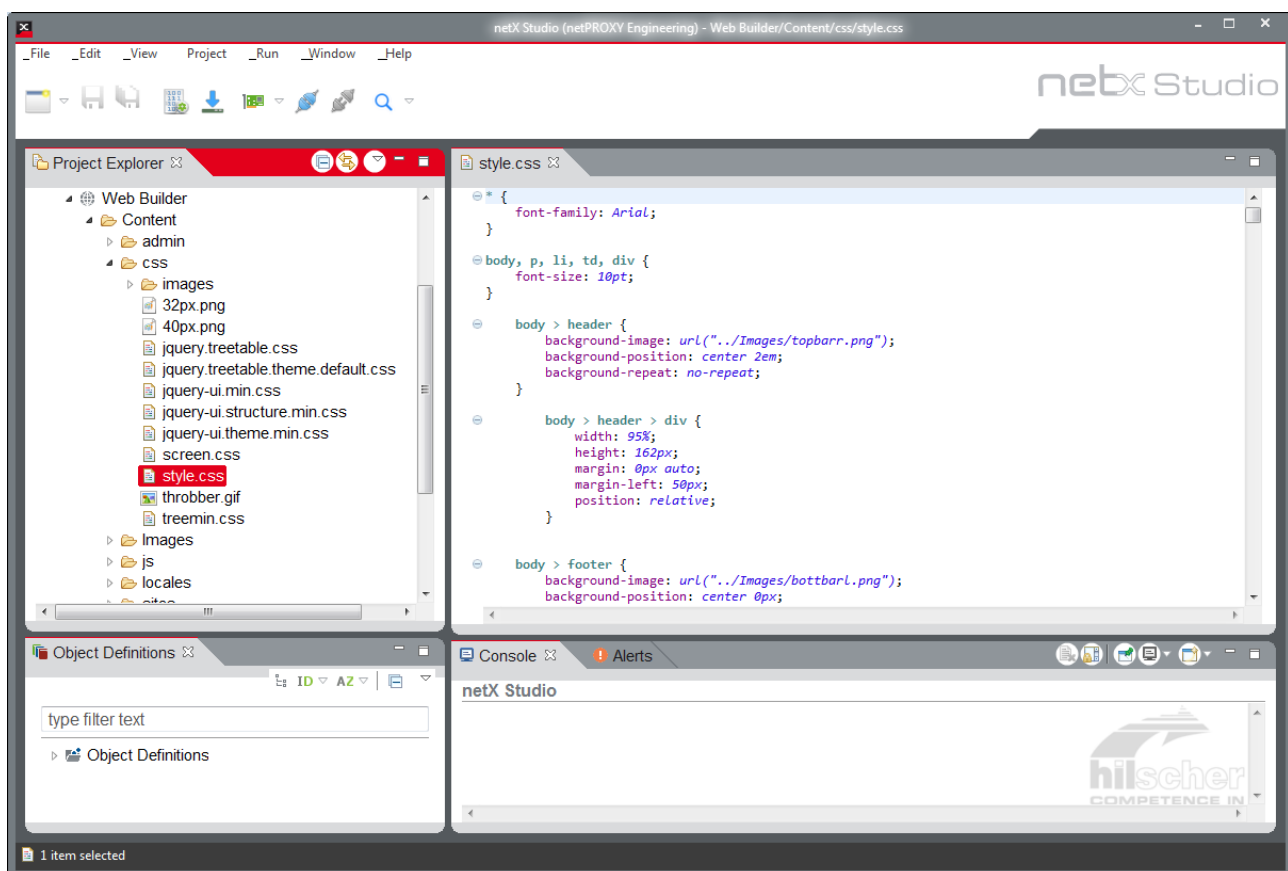


Figure 11: CSS file

2. Substitute header background image with gray color definition.

➤ In section

```
body > header {
background-image: url("../Images/topbarr.png");
background-position: center 2em;
background-repeat: no-repeat;
}
```

replace the code referencing the background-image

```
background-image: url("../Images/topbarr.png");
```

with a color definition for a gray background area, e. g.

```
background-color: #E6E6E6;
```

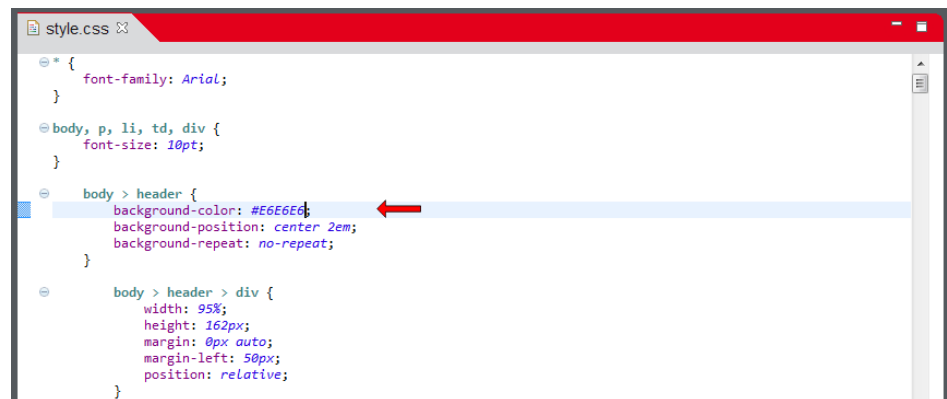


Figure 12: Replacing background image with gray area

3. Enlarge font size.

➤ In the `/** CONTENT */` area of the style sheet, scroll to the declaration of the standard font size

```
body, p, li, td, div {
font-size: 10pt;
}
```

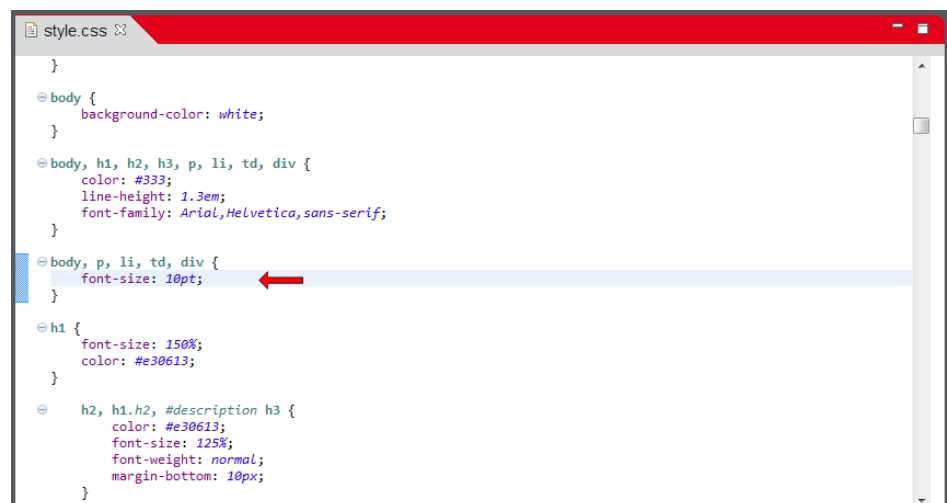


Figure 13: Standard font size declaration

➤ Change the `font-size` value from 10pt to 12pt.

4. Change color of first-level headings (H1)

- In the declaration of the H1 heading (in the paragraph below)

```
h1 {  
  font-size: 150%;  
  color: #e30613;  
}
```

change the color definition `color: #e30613` to a different value, e. g. to `#0000FF` for blue.

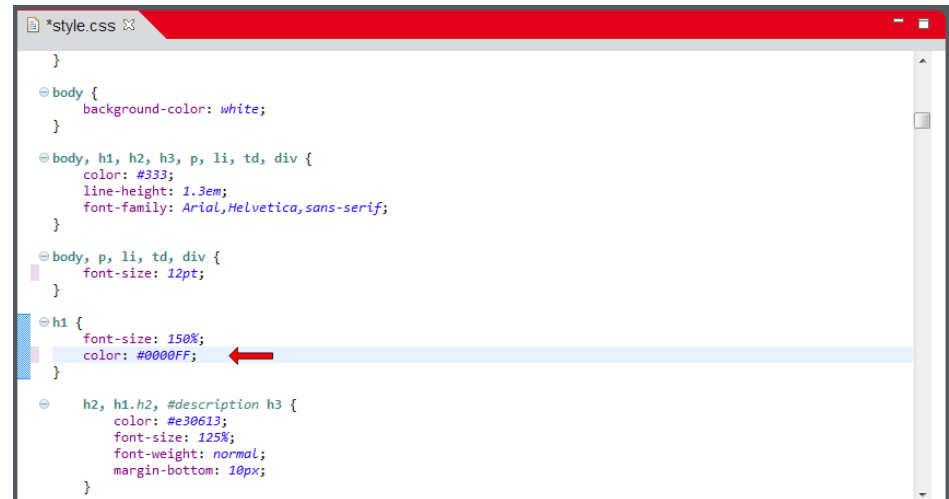


Figure 14: Changing color of H1

5. Move menu pane upwards.

- In the style sheet, scroll down to the section declaring the position of the navigation pane:

```
#main_nav {  
  position: absolute;  
  bottom: 40px;  
  left: 0;  
}
```

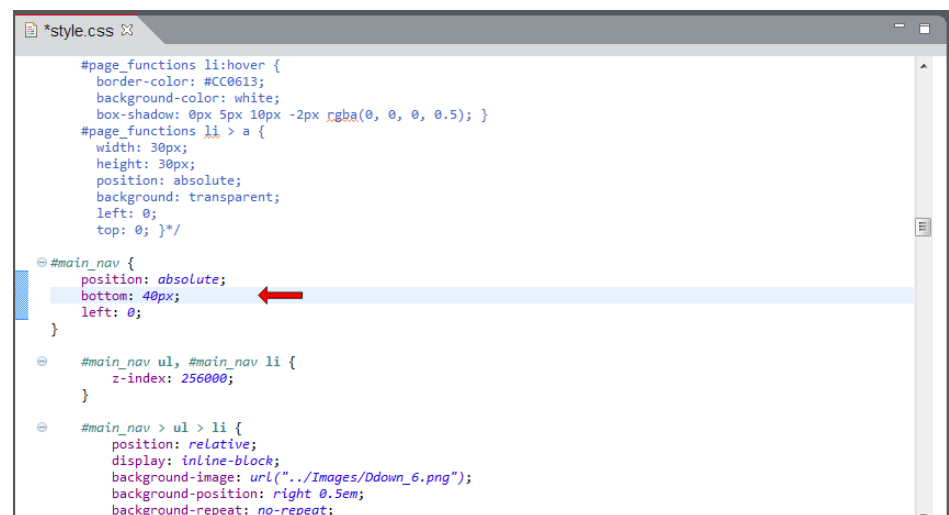


Figure 15: Changing position of navigation pane

- Increase the `bottom: 40px;` value to e. g. `120px;`
(this increases the distance of the navigation menu from the lower border of the header area for 80 pixels to the upward direction)
- In the menu, choose **File > Save** or press **Ctrl + S** on your keyboard to save the changes to the `style.css` file.

3.4 Changing text in HTML page

This section shows how to change the welcome text on the **info.htm** page. The **info.htm** page is integrated in the **home.htm** page as an inline frame ("iframe element"). The **home.htm** page is the first page to be displayed after calling up the **index.htm**.

1. Open HTML page.
 - In the folder **Web Builder > Content > sites**, open **info.htm** file.
 - The file is opened in the editor.

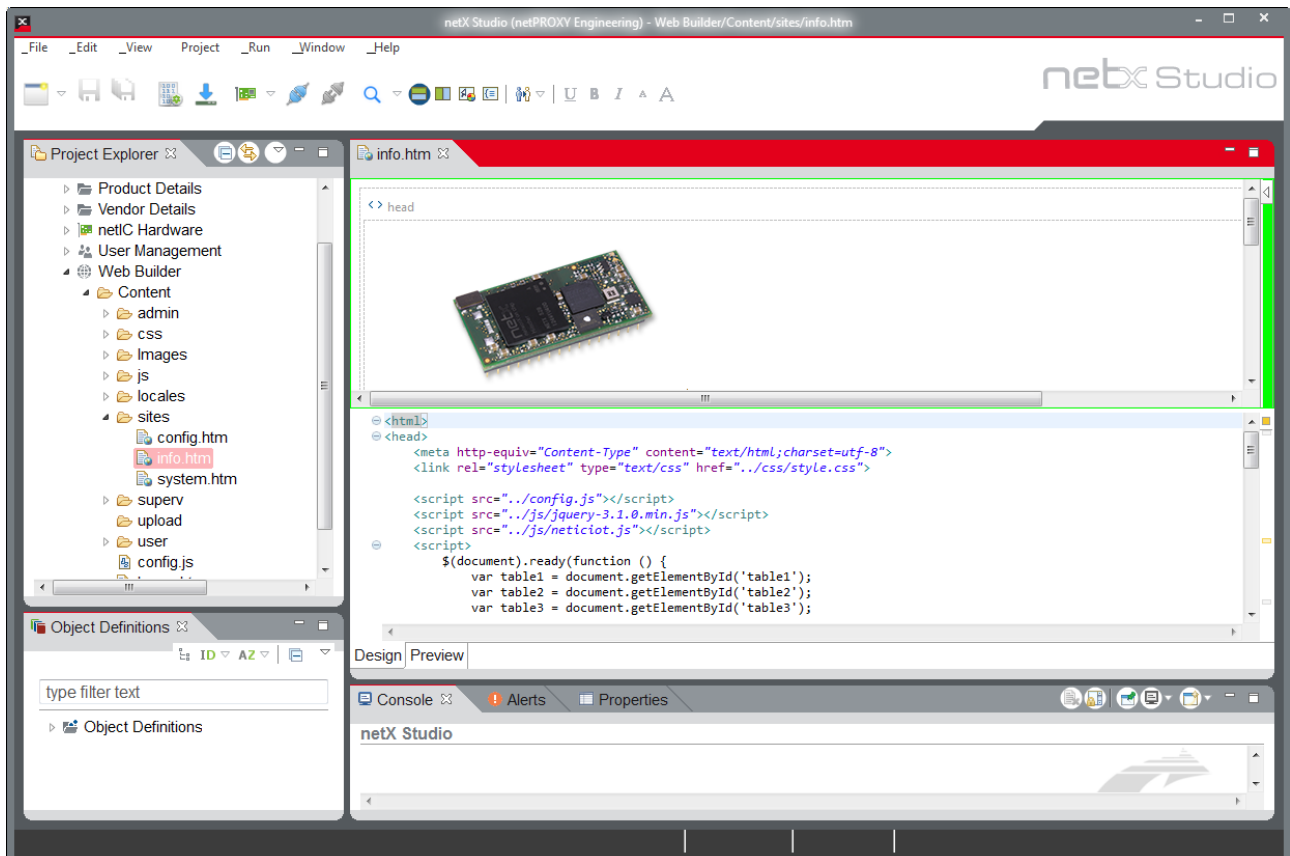


Figure 16: Opened info.htm file

2. Enter new text.

- In the Source Page (lower part of the editor), scroll down to the first paragraph that you want to change, e. g. **netX Device** heading (as an alternative, you can click on the text in the upper part of the editor in order to highlight the corresponding text in the HTML code).

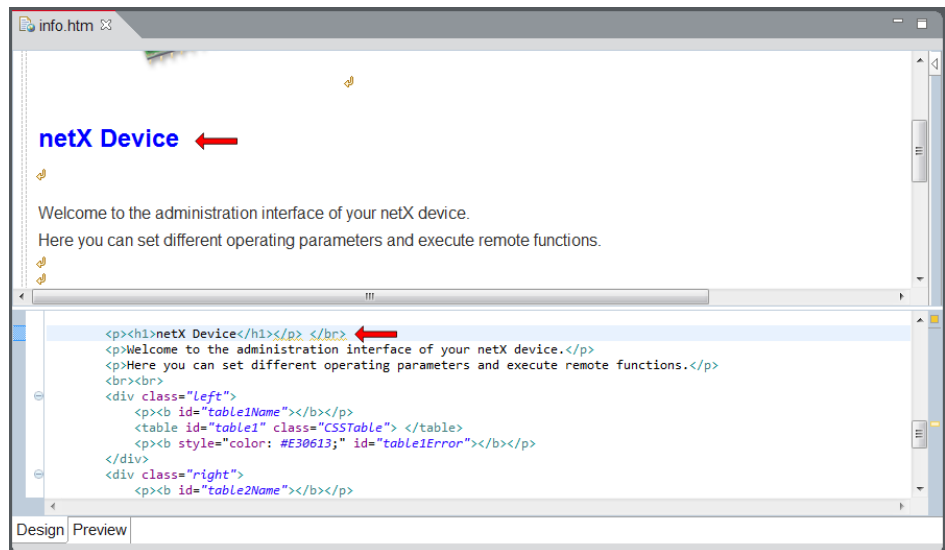


Figure 17: Changing heading text

- In the `<p><h1>netX Device</h1></p> </br>` tag, replace the name **netX Device** by an individual device name, e. g. **B100-EIS**. Change other text passages according to your needs.

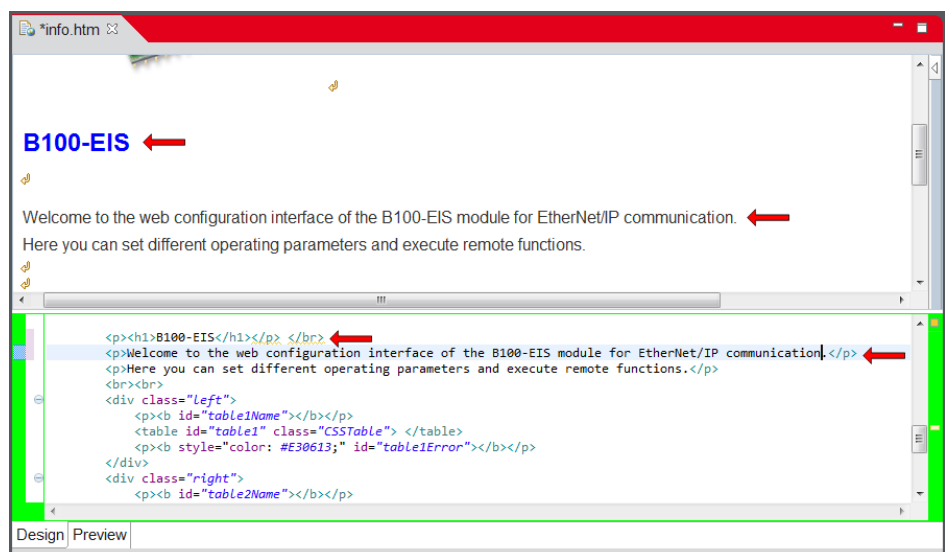


Figure 18: Changing welcome text

- In the menu, choose **File > Save** or press **Ctrl + S** on your keyboard to save the changes to the **info.htm** file.
- ⇒ You have changed parts of the layout and the contents of the default web pages shipped with the Web Builder.

4 Using Product Macros to customize web pages of product variants

4.1 Overview

During the build process, all content created and managed by the **Web Builder** (HTML, graphics, JavaScript and style sheet files) is compressed and packed into a **content.tar** archive file.

For each product variant contained in the project, a separate **content.tar** file is generated, which will automatically be stored in the **Product Build > Products > [name of product variant] > Web Builder** directory:

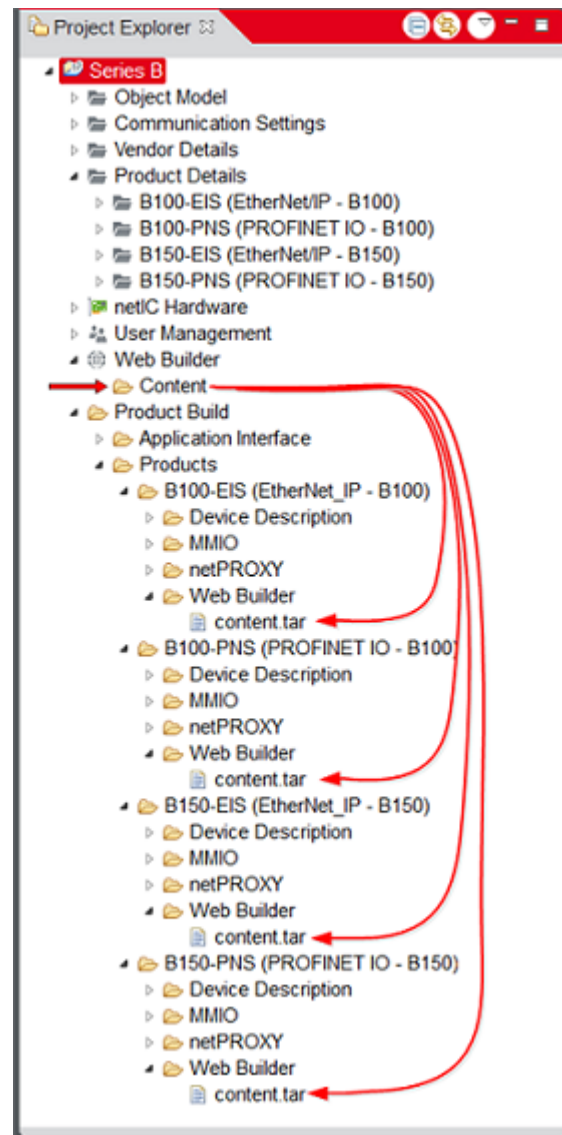


Figure 19: Content.tar files in Project Explorer

The **content.tar** file can then be downloaded from netX Studio into the target device via USB (on its own or together with other product configuration data). Instructions for this can be found in the operating instruction manual *netX Studio Engineering Tool – Device development*, DOC160103OIXXEN in section *Downloading product files*.

As long as you haven't defined any **Product Macros** yet, the contents of the **content.tar** files is identical for all product variants; i. e. each **content.tar** contains exactly the same HTML pages, graphics and style sheets.

If you want to assign individual content (customized texts and images within a web page) to individual product variants, you need to use the **Product Macros** of the **netX Studio Engineering Tool**.

A **Product Macros** element will automatically be added to every product variant that you create in the **Product Details** folder.

The element contains an empty list, in which you can enter product-specific variables that shall be used for this product variant during the build process.

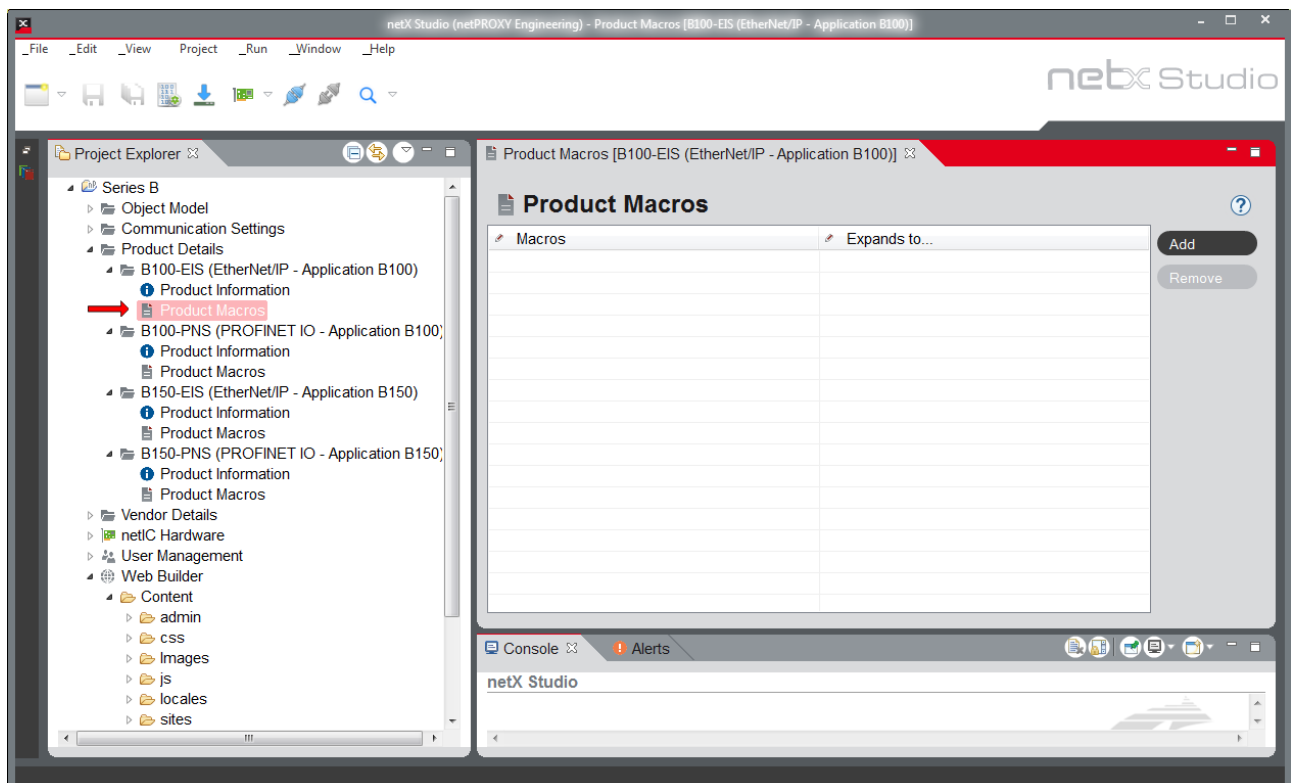


Figure 20: Empty Product Macros element

The **Macros** column is to be filled with the variables, and the **Expands to...** column is to be filled with the values that shall be used as actual values for this product variant.

A variable must begin and end with a percentage symbol (%), e. g. `%DEVICE_NAME%`.

There are no special rules or restrictions for the corresponding value in the **Expands to...** column. Here you can enter any character strings like a name like `B100-EIS Adapter Module` or a reference to a certain graphics file or style sheet.

The variables can be inserted into the source code of any HTML file of the Web Builder. For example, you could insert a `%STYLESHEET%` variable for an individual layout into the header declaration of the **home.htm** page, a `%DEVICE_NAME%` variable for individual product names into the welcome text and an `%IMAGE%` variable for individual product pictures.

During the build process, these variables will be replaced by the product-specific character strings or paths to individual file directories, thus creating customized home pages for each product variant featuring individual color layouts (`%STYLESHEET%`), product/device names (`%DEVICE_NAME%`) and product picture photos (`%IMAGE%`).

The following picture depicts the flow of the Product Macros:

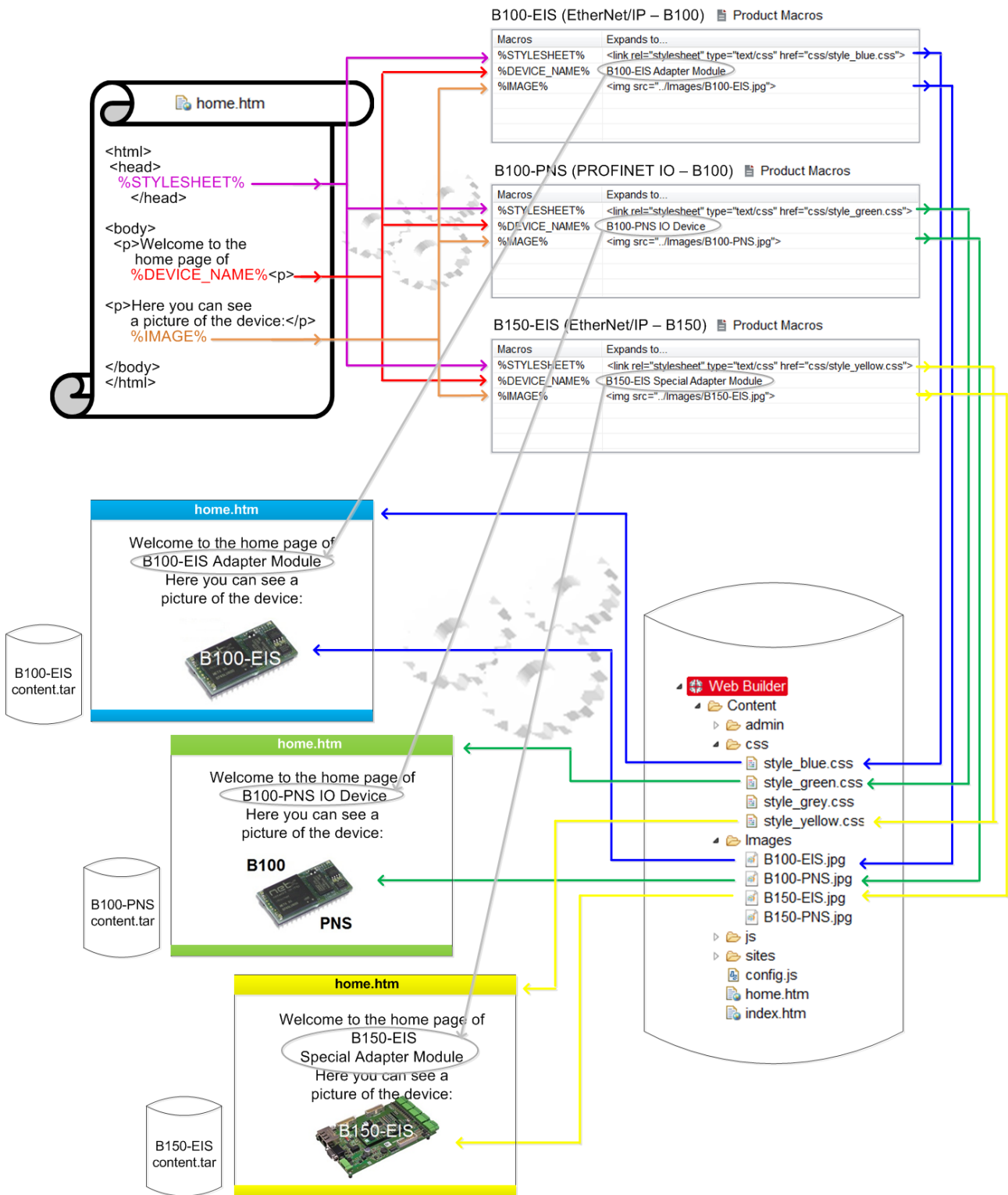


Figure 21: Product Macros and variables in build process

4.2 Application example of Product Macros

Prerequisites

- You have created separate Cascading Style Sheets for each product variant (e. g. **style_blue.css**, **style_green.css**, **style_yellow.css** and **style_grey.css**) and imported them into the **Web Builder > Content > css** folder.
- You have created separate graphic files for each product variant (e. g. **B100-EIS.jpg**, **B150-EIS.jpg**, and **B150-PNS.jpg**) and imported them into the **Web Builder > Content > Images** folder.



Note:

To import files into the **Web Builder**, select the import target folder in the **Project Explorer** then choose **Add Files** from the context menu.

Step-by-step instructions

1. Insert variable for CSS style sheet in HTML code of **home.htm** file.
 - In folder **Web Builder > Content**, double-click **home.htm** file.
 - The file is opened in the editor.

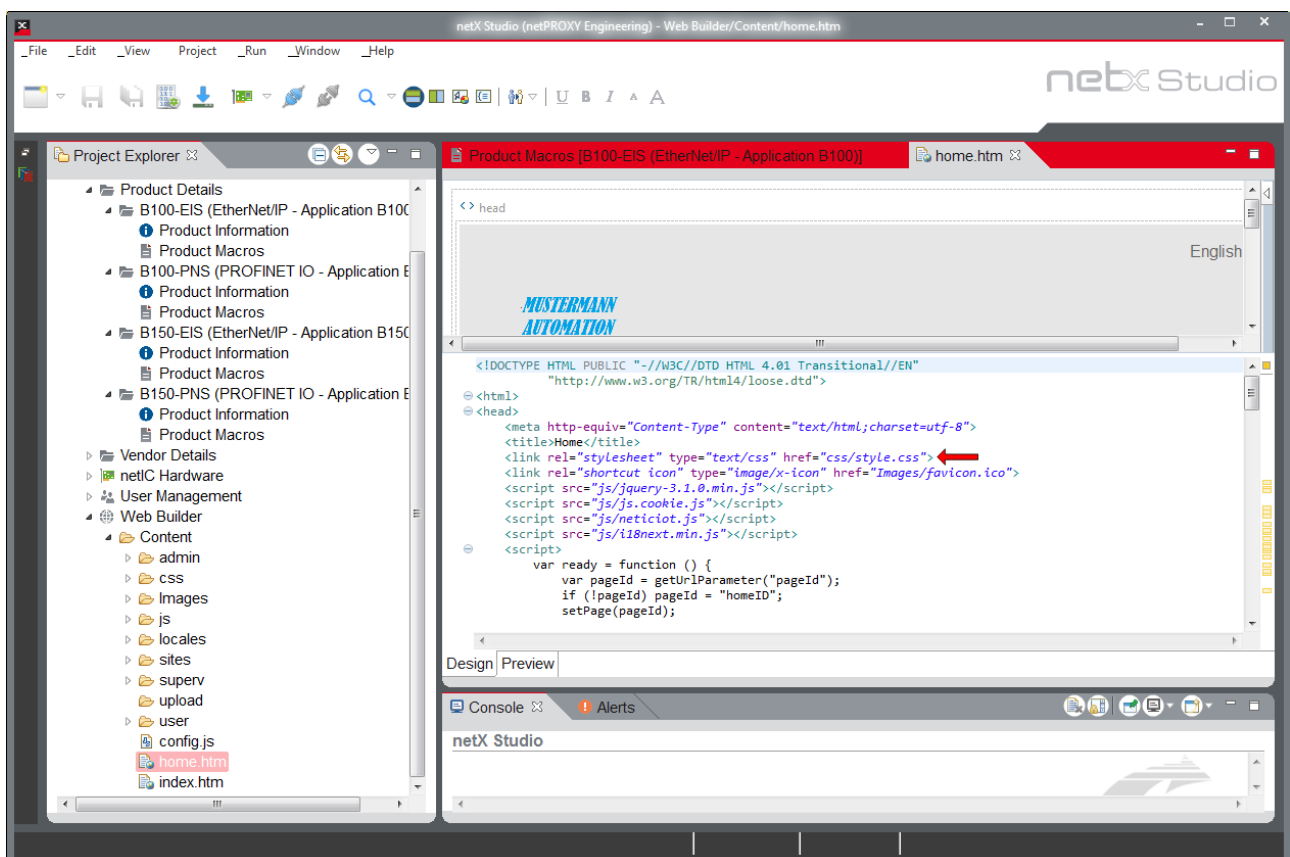


Figure 22: Opened home.htm file

- In the `<head>` element, replace the link to the CSS style sheet `<link rel="stylesheet" type="text/css" href="css/style.css">` with the variable `%STYLESHEET%`.



Figure 23: Inserting variable for CSS style sheet

- Press **Ctrl + S** on your keyboard to save the changes to the **home.htm** file.
2. Insert variable for device name in HTML code of **info.htm** file.
 - In the folder **Web Builder > Content > sites**, double-click **info.htm** file.
 - The file is opened in the editor.

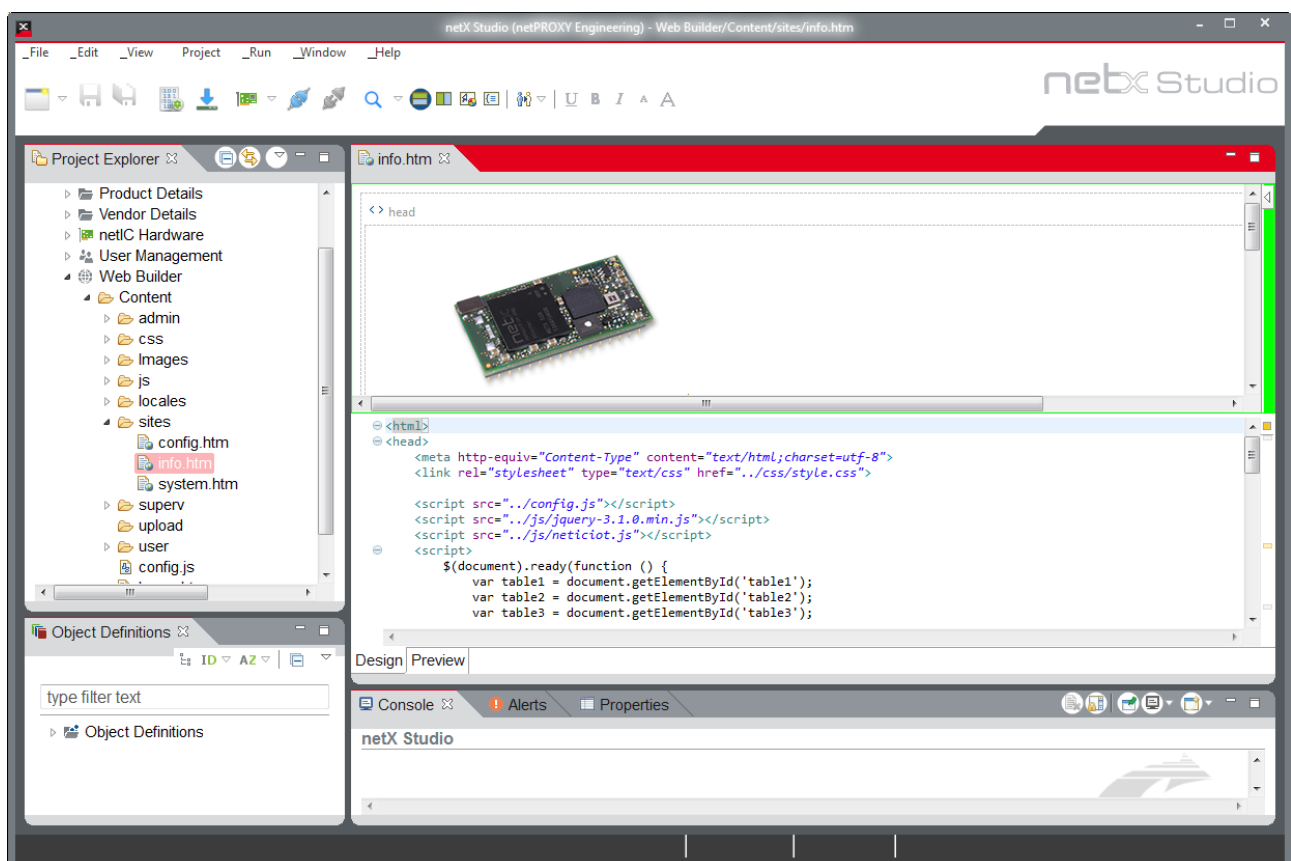


Figure 24: Info page

- In the Source Page (lower part of the editor), scroll down to the paragraphs containing the device name.

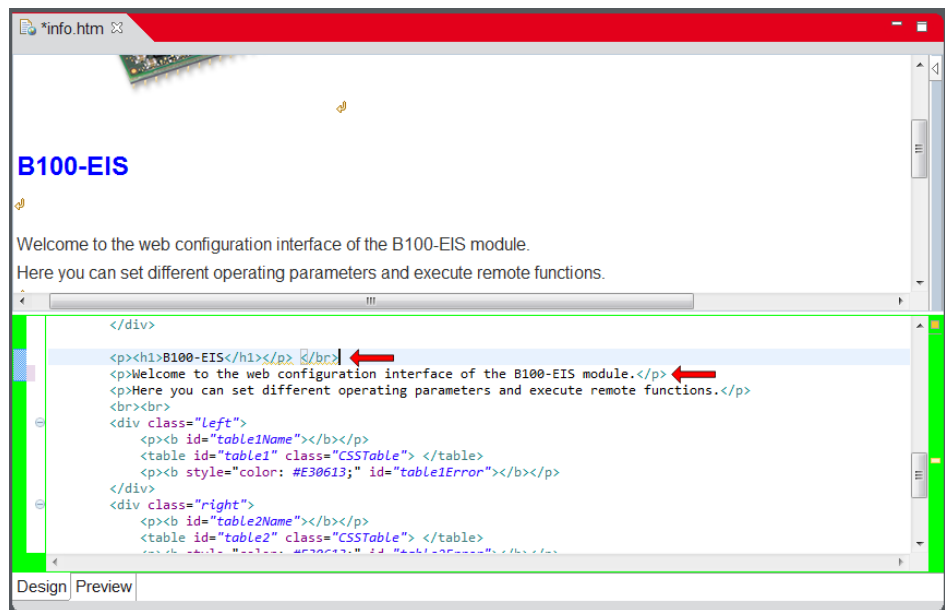


Figure 25: Device name in HTML code

- Replace the device name (in this example B100-EIS in the <h1> heading and B100-EIS module in the following paragraph) with the %DEVICE_NAME% variable.

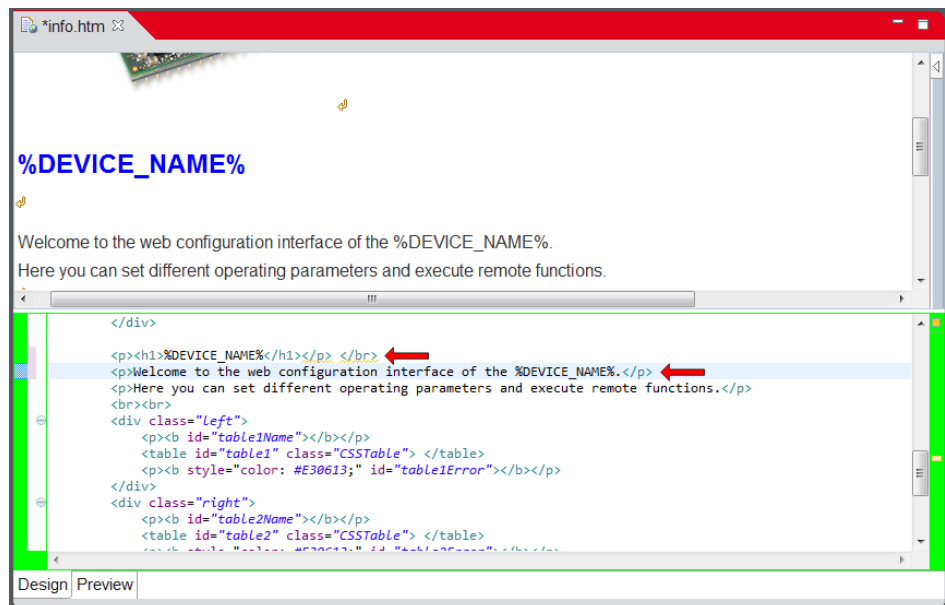


Figure 26: Inserted device name variable

3. Insert variable for product image.

- Scroll to the code line containing the reference of the device image.

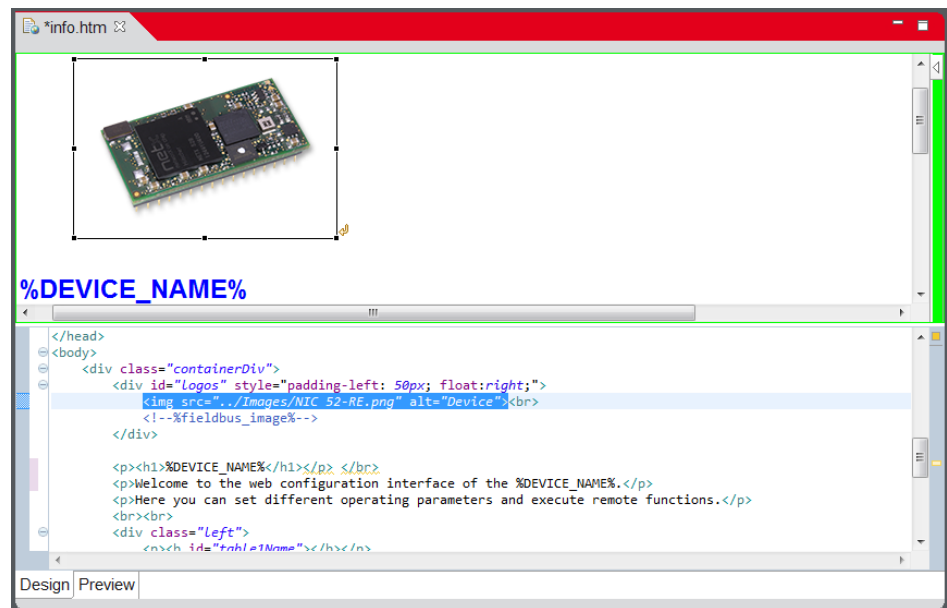


Figure 27: Reference of device picture

- Replace the link to the product image `` with the `%IMAGE%` variable.

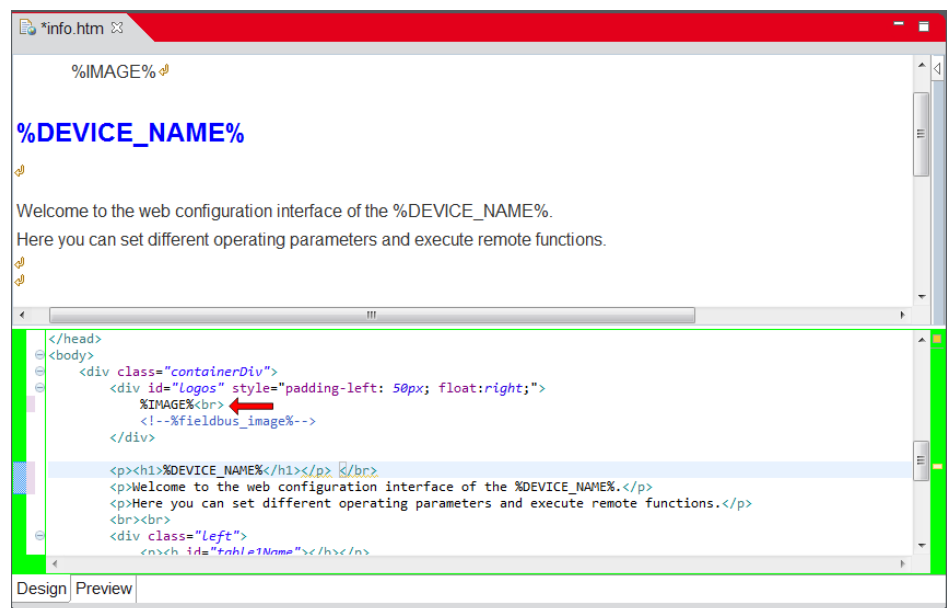


Figure 28: Inserted device picture variable

- Press **Ctrl + S** on your keyboard to save the changes to the **info.htm** file.

4. Open Product Macros element of first product variant.
 - In the **Product Details** > [Product name] folder, double-click **Product Macros** element.
 - An empty list of Product Macros for the product variant opens in the editor.
5. Define variables of first product variant.
 - Click **Add** button to define a variable.

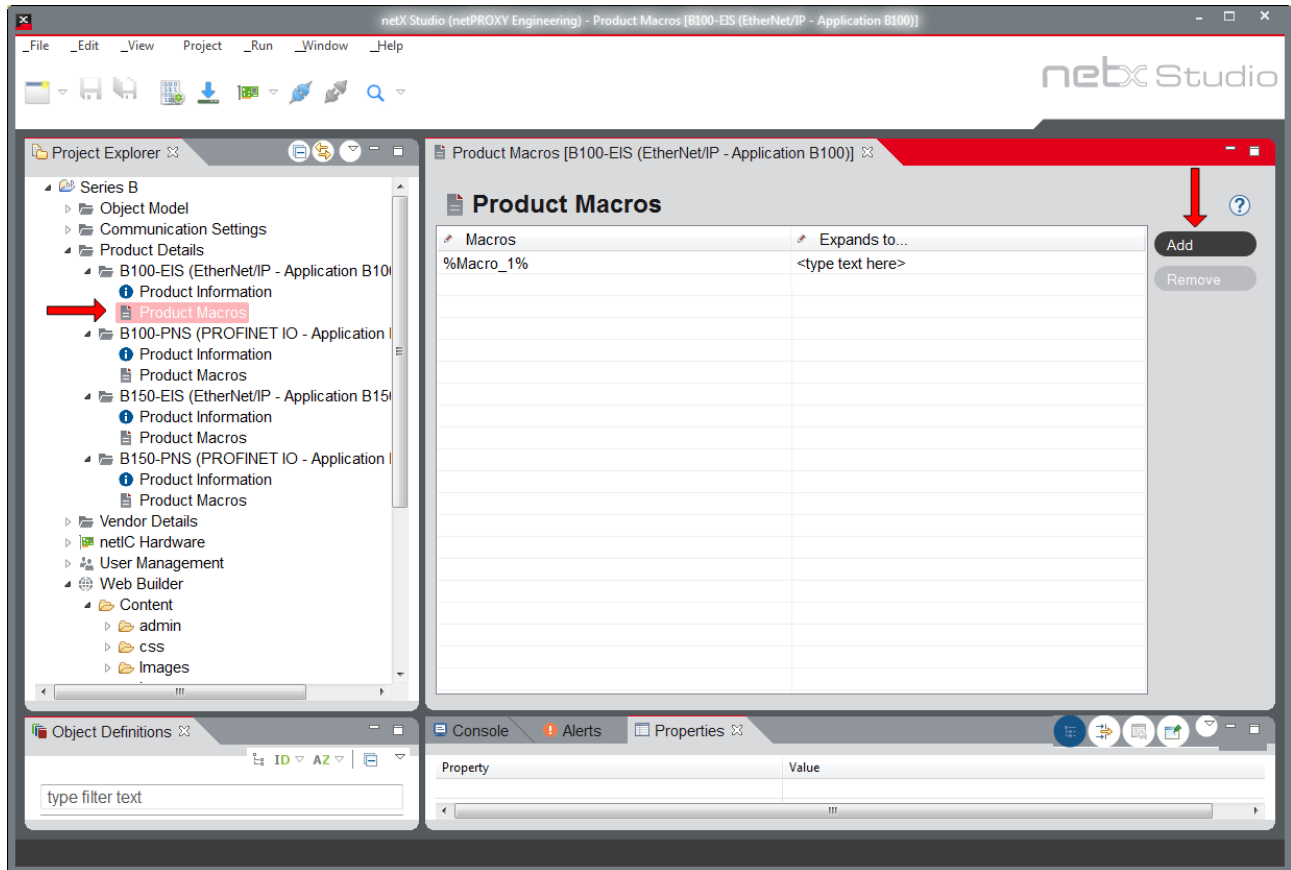


Figure 29: Open macro

- The editor inserts a dummy variable named %Macro_1% in the **Macros** column, and a <type text here> place-holder text in the **Expands to...** column.
- In the **Macros** column, replace the dummy variable %Macro_1% with the %STYLESHEET% variable. In the **Expands to...** column, enter the link to the CSS style sheet which you want to use for this product variant, e. g. <link rel="stylesheet" type="text/css" href="css/style_blue.css">.
- Click **Add** button again, then add %DEVICE_NAME% variable. In the **Expands to...** field enter the device/product name you want use for this product variant, e. g. B100-EIS Adapter Module.

- Then add the %IMAGE% variable for the product picture and the path to the image file you want to use for this product variant, e. g. ``.

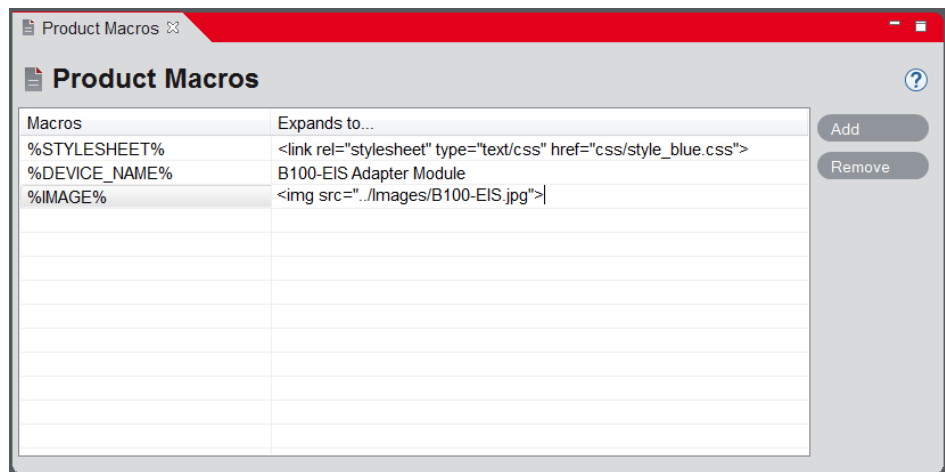


Figure 30: Variables for product B100-EIS

- You have defined the "macros" respectively variables for the first product variant.

6. Define variables for second product variant.

- Open the **Product Macros** element of the second product variant, click **Add** button and add the `%STYLESHEET%`, `%DEVICE_NAME%` and `%IMAGE%` variables in the **Macros** column. In the **Expands to...** column enter the path to the style sheet, the device name and the path to image file you want to use for this product variant:

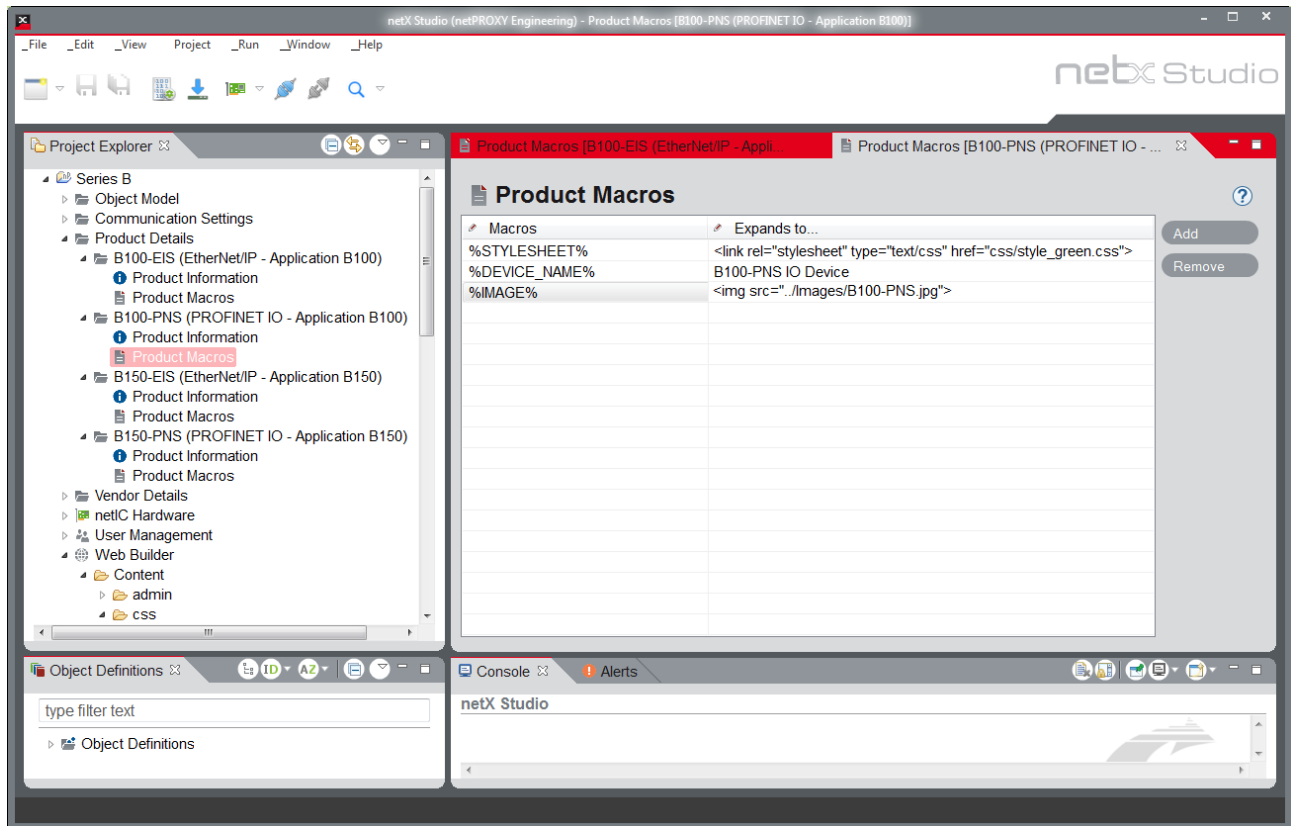


Figure 31: Variables for product B100-PNS

- You have defined the "macros" respectively variables for the second product variant.
7. Define variables for remaining product variants.
- Repeat the procedure described above for the Product Macros of the other products of your project.

**Important:**

When using variables in commonly used resources like the **home.htm** page, make sure to define a value for this variable in the Product Macros of all product variants. Otherwise the variable won't be resolved during the build process and the variable will be displayed on the web page of the corresponding product variant (missing the value) in its original string (e. g. `%DEVICE_NAME%` in the welcome text on the **home.htm** page). Also, style sheets and images referenced by variables would be completely missing in the build output of product variants without properly defined Product Macros.

- ⇒ You have defined Product Macros and inserted them into the web content. You can now start the build process.

5 Checking web content locally in browser before download

After the build process, you can view and display your produced web pages locally in your standard web browser on your development PC before downloading them into the target device. For this, netX Studio offers quick access to the **content.tar** file via Windows Explorer.

In the Windows Explorer, you can copy the TAR file to a local directory on your PC, unpack it and open the web pages locally in your browser.

If your project features multiple products (product variants) and if you have used Product Macros to customize the web pages of these variants, you can repeat the subsequently described procedure for each product and thus check whether the variables have been properly resolved.

To check your output, proceed as follows:

1. Copy content archive to local directory.
 - After the build process has been finished, open **Product Build > Products > [Product name] > Web Builder** folder in the **Project Explorer**.
 - Select **Web Builder** folder and choose **Show in Windows Explorer** from the context menu.

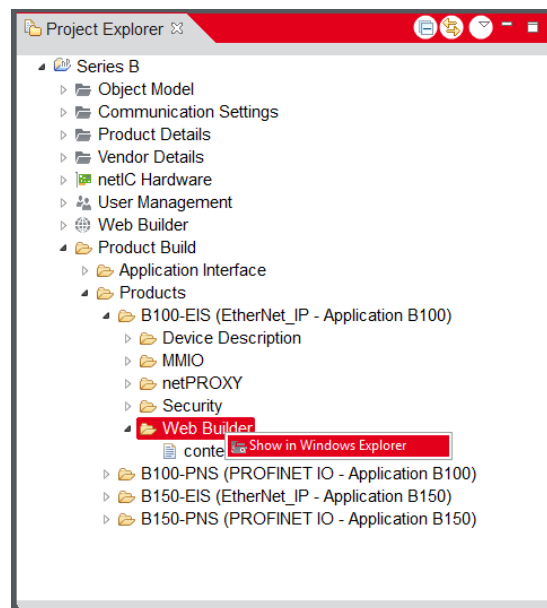


Figure 32: Opening folder in Windows Explorer

- The Windows Explorer opens and shows the selected folder in your file system:

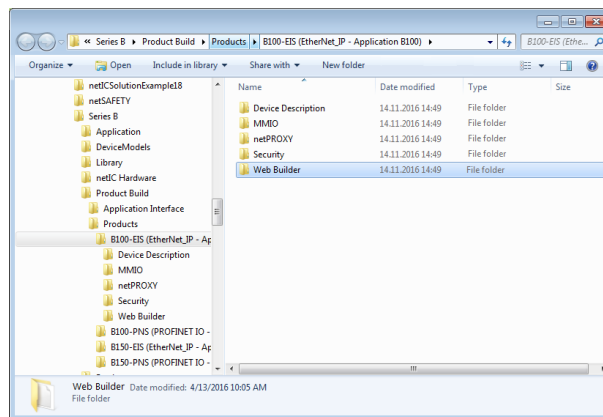


Figure 33: Product Build directory in Windows Explorer

- Open **Web Builder** folder and copy the **content.tar** file to a suitable location, e. g. to your desktop.
2. Unpack archive file.
- Use an archive tool supporting the TAR format (like e. g. 7-Zip) to unpack the **content.tar** file.
 - The unpacked **content.tar** folder contains the "processed" elements of the **Content** folder of the **Web Builder**. If you have used Product Macros, the variables within the HTML pages have been replaced by the values defined for this product variant.

Name	Date modified	Type	Size
admin	17.10.2016 11:45	File folder	
css	17.10.2016 11:45	File folder	
Images	17.10.2016 11:45	File folder	
js	17.10.2016 11:45	File folder	
locales	17.10.2016 11:45	File folder	
sites	17.10.2016 11:45	File folder	
superv	17.10.2016 11:45	File folder	
user	17.10.2016 11:45	File folder	
config.js	17.10.2016 11:45	JScript-Skriptdatei	7 KB
content.tar	17.10.2016 11:45	TAR-Datei	1.130 KB
home.htm	17.10.2016 11:45	Firefox HTML Doc...	5 KB
index.htm	17.10.2016 11:45	Firefox HTML Doc...	1 KB

Figure 34: Unzipped content folder

3. Display web pages.

- Double-click **index.htm** or **home.htm** file.
- ⇒ The start page (home page) of the web content is displayed in your standard web browser.

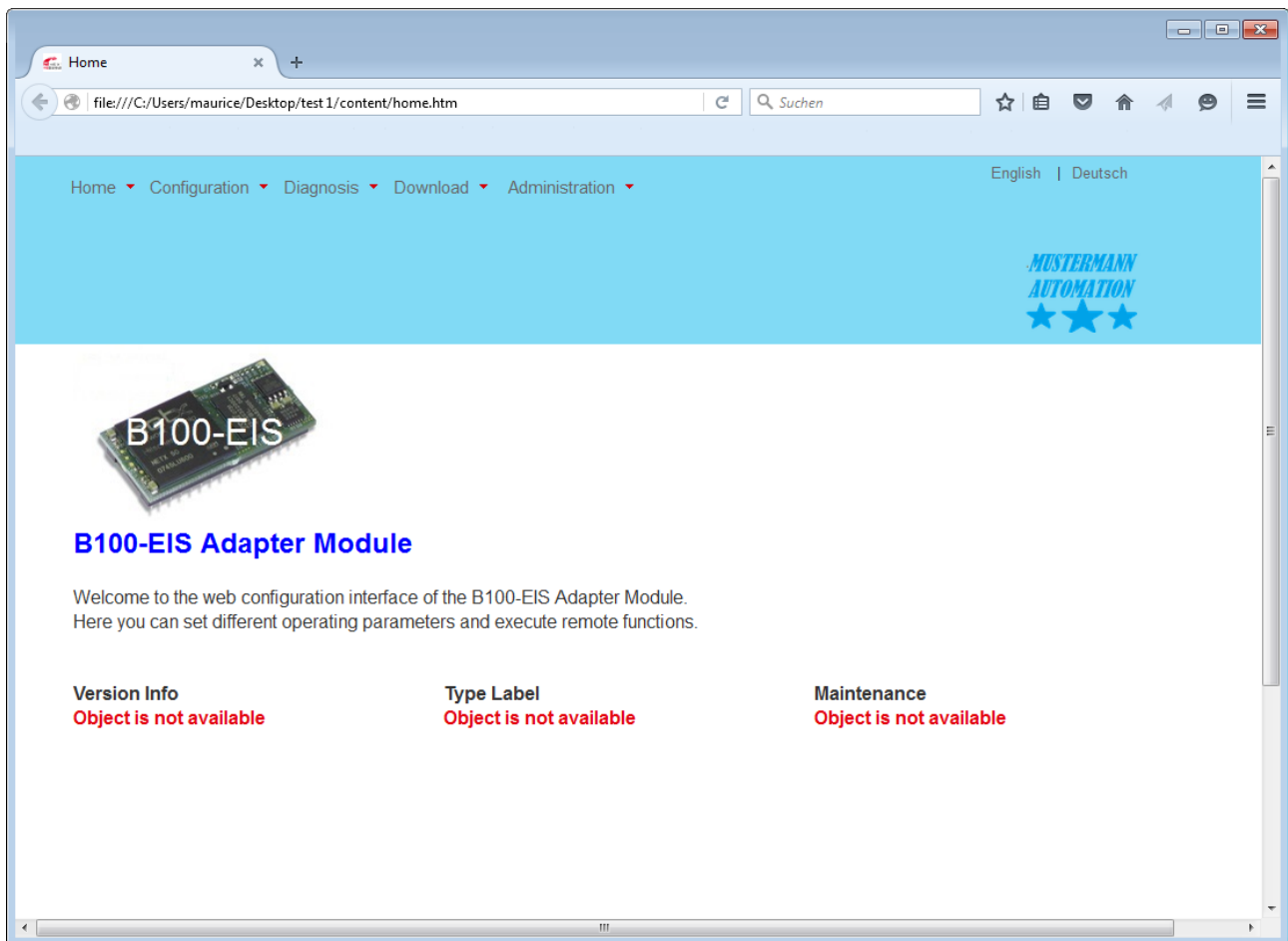


Figure 35: Locally displayed home page of B100-EIS product

**Note:**

Certain features cannot be displayed when you open the web pages locally on your PC (see "Object is not available" note in the figure above). These are the data objects that the servX WebServer would normally read via Java-Script from netPROXY-Server within the target device. Because you have opened the content only "locally" on your PC (without WebServer or netPROXY-Server) these data objects are not available for display.

- Repeat the procedure for the **content.tar** files of the other products/product variants if necessary.

6 Access rights for web pages

Overview

Access rights of users for individual web pages (and thus for certain functions of the WebServer) are controlled by membership of the user in one or several user groups (also called "roles"). The following groups/roles are defined for accessing the WebServer and the FTP server:

- Administrator
- Supervisor
- User

Users and groups

Users can be assigned to groups in the **User Accounts** element, which is located in the **User Management** folder of the netX Studio Engineering Tool.

Instructions for this can be found in the operating instruction manual *netX Studio Engineering Tool – Device development*, DOC160103OIXXEN in section *Configuring user management*.



Note:

A master user possessing all access rights is pre-defined in netX Studio. This predefined master user has the user name `admin` and password `admin`. It is strongly recommended to instantly change the default `admin` password for this master user! (The master user and its `admin` user name cannot be deleted or edited.)

Please note that after having configured and downloaded a project to a target device via netX Studio, the integrated WebServer also allows user management (adding new users, assigning groups/roles and passwords) on the device itself. The **User Accounts** web page on the WebServer can be accessed in the **Administration > User/PasswordManagement** menu (you must be member of the **Administrator** group to be allowed to access this page).

Access rights of user groups defined by folders

The access rights of user groups for individual web pages are defined by the directory in which a page is stored in the netX Studio Engineering Tool. Pages stored in the **admin** folder can be accessed on the WebServer only by members of the **Administrator** group, pages stored in the **superv** folder only by members of the **Supervisor** group and pages stored in the **user** folder only by members of the **User** group. Pages which are not stored in any of the above mentioned folders are freely accessible (e. g. the **home.htm** page):

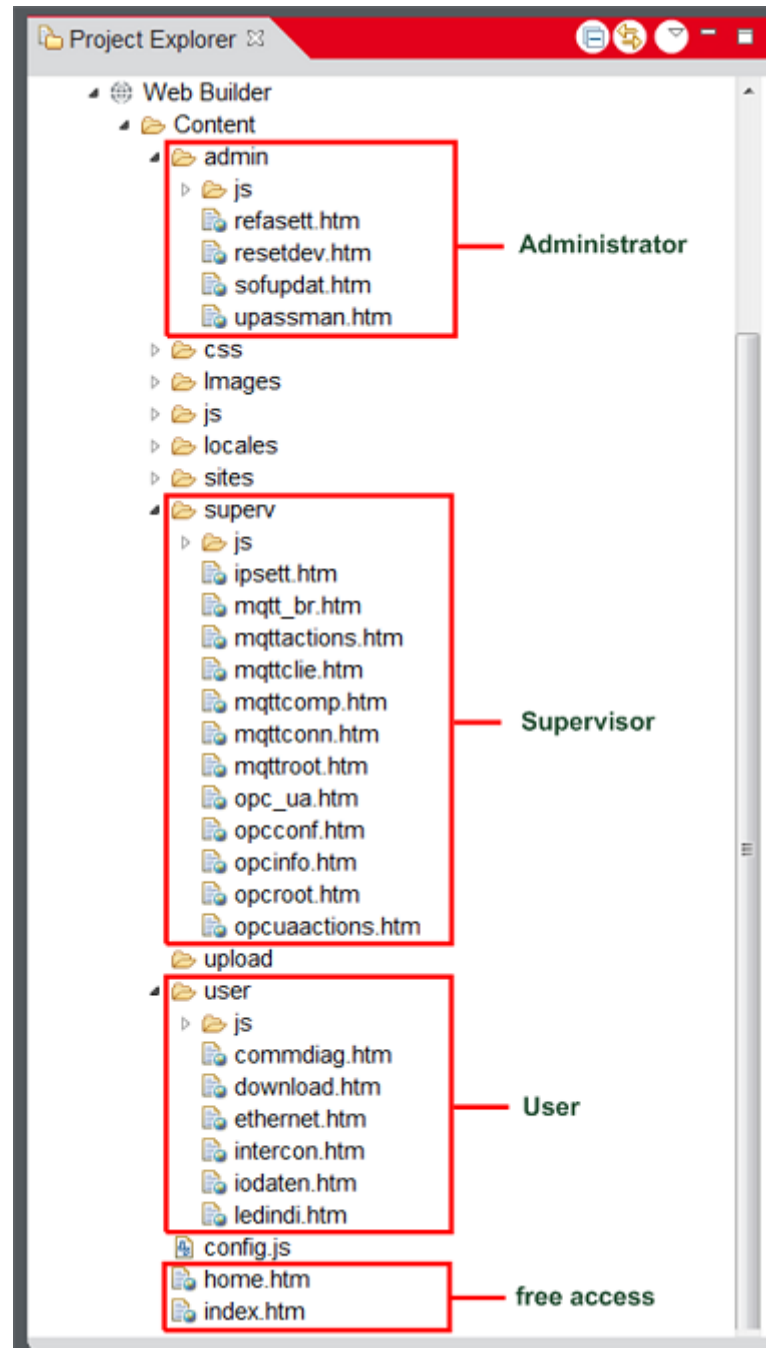


Figure 36: Assignment of HTML pages to user groups in Web Builder

If you want to create a new web page in the Web Builder, you can determine the access right for this page by storing it in one of these folders.

The following table shows which web page can by default be accessed by which user group after having built and downloaded the web content to the target device (i. e. if you do not change the standard storage locations of the web pages shown in the picture above):

User group (role)	Allows access of	Navigation in menu of WebServer
Administrator	Start page (landing page) with version info, type label ("virtual device type label") and maintenance notes	Home
	User management	Administration > User/Password Management
	Device reset	Administration > Device Reset
Supervisor	Start page (landing page) with version info, type label ("virtual device type label") and maintenance notes	Home
	IP settings	Diagnosis > IP Settings
	Configuration of MQTT client	Configuration > MQTT Client
	Configuration of OPC UA server	Configuration > OPC UA Server
User	Start page (landing page) with version info, type label ("virtual device type label") and maintenance notes	Home
	Diagnosis of network status	Diagnosis > Communication
	Diagnosis of Ethernet connection	Diagnosis > Ethernet
	Reading and writing of object data	Diagnosis > Object Data
	Download of device description file	Download

Table 2: Access rights of user groups (default)

The start (home) page of the WebServer can be freely accessed by everyone, even if you do not belong to any of these user groups (if you are "anonymus"). As soon as you try to open other pages from the home page a login dialog appears. If you then login as a user not possessing the required access rights (i. e. if you do not belong to the corresponding group) access will be denied.

7 Integrating netPROXY objects in web pages

Overview

The WebServer is able to read and write netPROXY objects (e. g. IO data of actors and sensors) via netPROXY server and JavaScript, and to dynamically integrate them in its web pages.

In its state of delivery, the standard content of the Web Builder is equipped with a web page featuring such object data. After downloading the web content to the target device, this page can be accessed from your browser by choosing **Diagnosis > Object Data** from the navigation menu:

Home ▾ Configuration ▾ Diagnosis ▾ Download ▾ Administration ▾

Object Data

Object Data - Object 0x40020001 - Instance: 1

Parameter	Value	Write
value	7	<input type="text"/>
state	1	<input type="text"/>
mode	1	<input type="text"/>
limit_high	9	<input type="text"/>
limit_low	2	<input type="text"/>

Last refresh: 15:11:28

Figure 37: Example of web page featuring netPROXY objects

Prerequisites

Note that objects which shall be accessed by the WebServer must be labeled accordingly in the **Object Definitions** of the **netX Studio Engineering Tool** beforehand [see position (1) in the following figure]. You must also add a **Web Server** element to your project in the **Communication Settings** and select the corresponding label in the **Filter** field of the **Web Server** element [see position (2) in the figure after next].



Instructions for this can be found in the operating instruction manual *netX Studio Engineering Tool – Device development*, DOC160103OIXXEN in section *Adding WebServer, OPC UA, MQTT*.

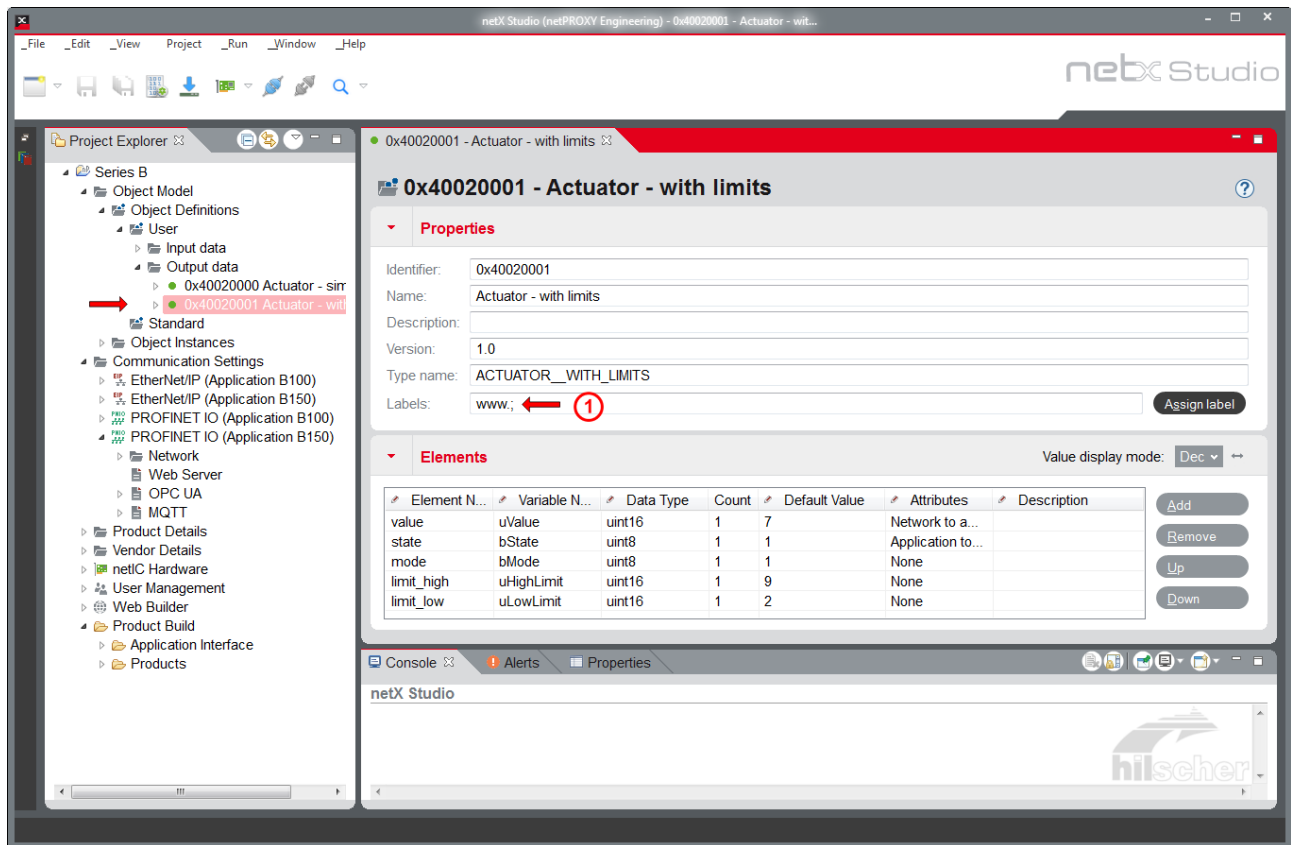


Figure 38: Labelling object for WebServer access

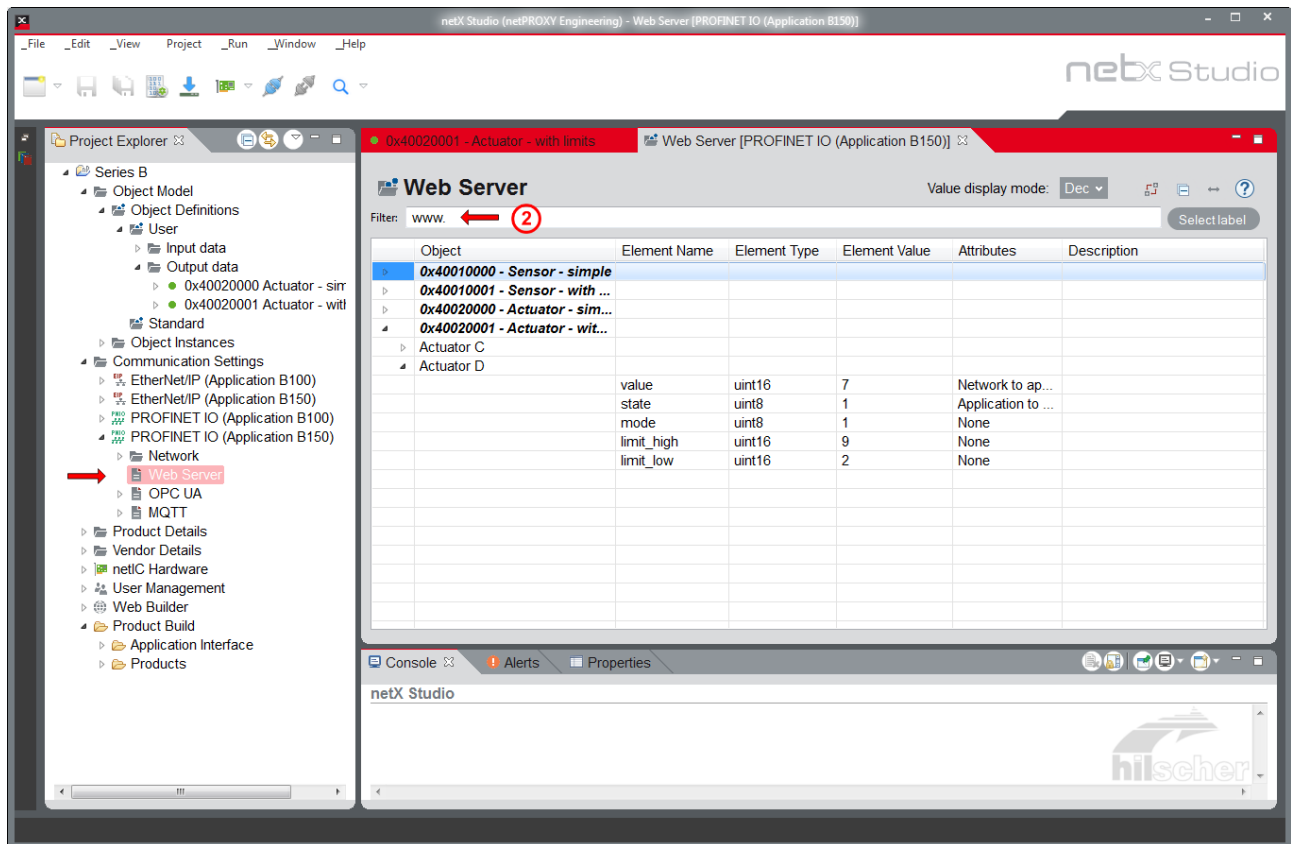


Figure 39: Adding WebServer element to product and add label to filter

JavaScript functions

The subsequently listed JavaScript functions are used for integrating netPROXY object elements into a web page:

- reading netPROXY object element (function readDataValues with op=read)
- writing netPROXY object element (function doWrite with op=write)
- reading netPROXY object description (function updateTable with op=getD)

These functions are defined in the **neticiot.js** JavaScript file, which is stored in the **Web Builder > Content > js** folder:

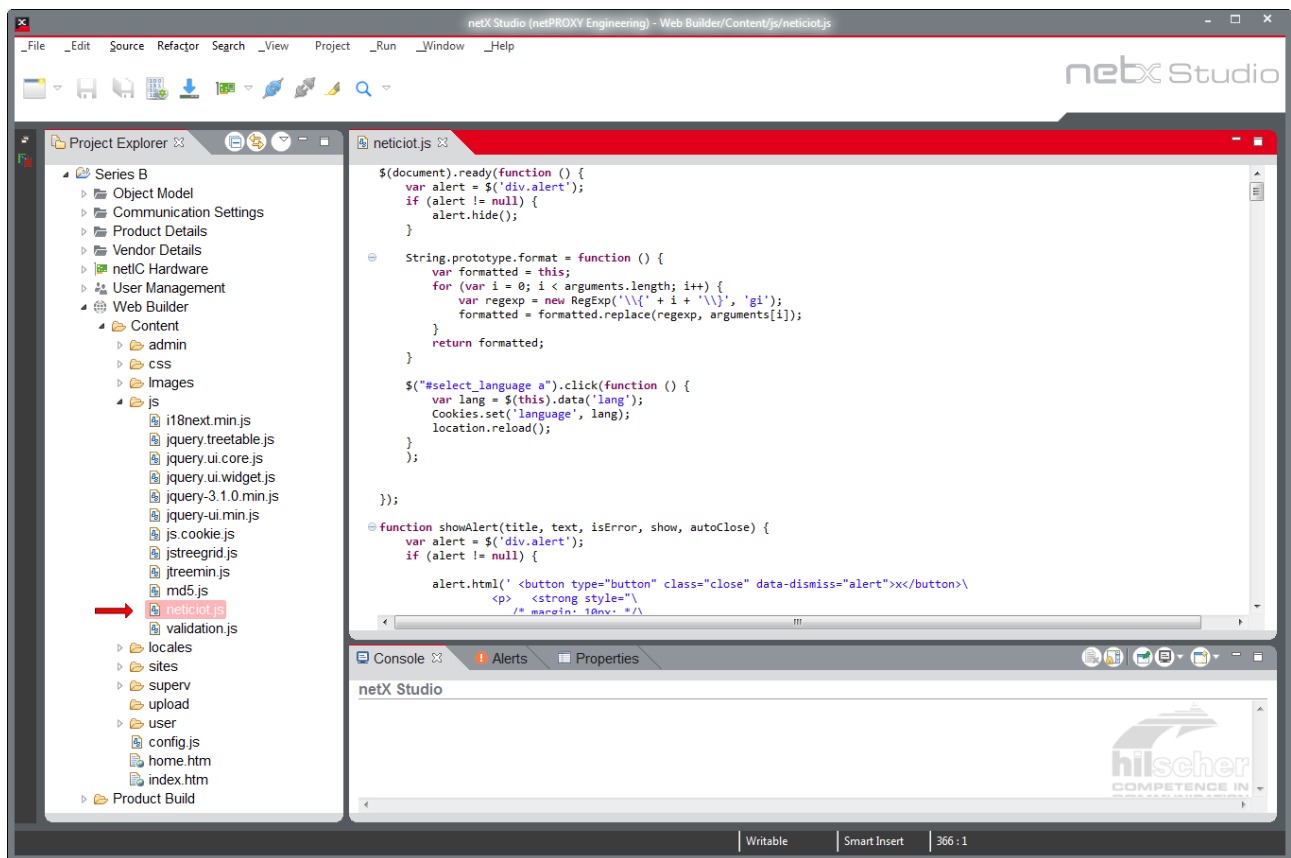


Figure 40: neticiot.js Java Script file

Reading netPROXY object element

This function is defined in function `readDataValues` (line 366 ff.) with `op=read`:

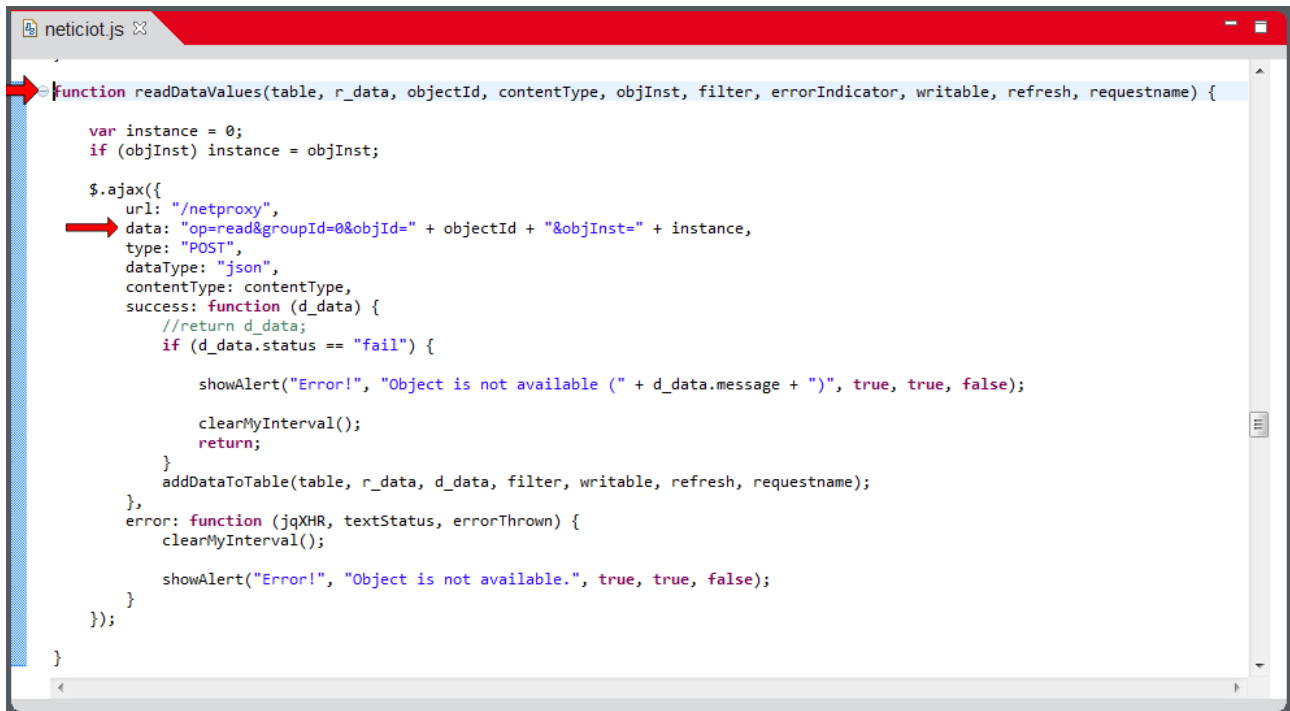


Figure 41: Code in JavaScript for reading netPROXY object elements

Writing netPROXY object elements

This function is defined in function `doWrite` (line 599 ff.) with `op=write`:

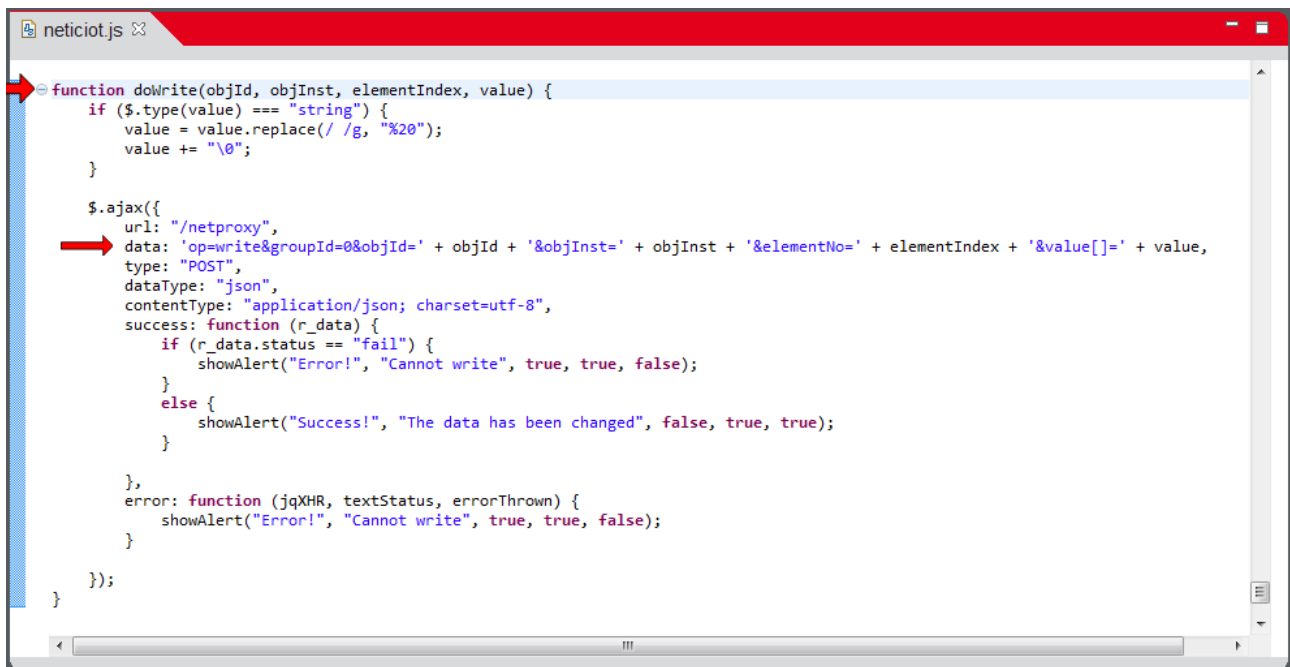


Figure 42: Code in JavaScript for writing netPROXY object elements

Reading netPROXY object description

This function is defined in function `updateTable` (line 121 ff.) with `op=getD`:



Figure 43: Code in JavaScript for reading netPROXY object descriptions

List of Figures

Figure 1:	Using Web Builder to create web content for servX WebServer	9
Figure 2:	Start page without customization.....	10
Figure 3:	Start page after customization.....	11
Figure 4:	Web Builder in opened project	12
Figure 5:	Add file.....	13
Figure 6:	New logo graphic file in Project Explorer.....	13
Figure 7:	home.htm page in editor	14
Figure 8:	Auto focus on source code element of Logo	15
Figure 9:	Adding own web address	15
Figure 10:	Changed image attributes	16
Figure 11:	CSS file.....	17
Figure 12:	Replacing background image with gray area	18
Figure 13:	Standard font size declaration	18
Figure 14:	Changing color of H1	19
Figure 15:	Changing position of navigation pane	19
Figure 16:	Opened info.htm file	21
Figure 17:	Changing heading text.....	22
Figure 18:	Changing welcome text	22
Figure 19:	Content.tar files in Project Explorer.....	23
Figure 20:	Empty Product Macros element	24
Figure 21:	Product Macros and variables in build process	26
Figure 22:	Opened home.htm file	27
Figure 23:	Inserting variable for CSS style sheet	28
Figure 24:	Info page.....	28
Figure 25:	Device name in HTML code	29
Figure 26:	Inserted device name variable.....	29
Figure 27:	Reference of device picture.....	30
Figure 28:	Inserted device picture variable.....	30
Figure 29:	Open macro	31
Figure 30:	Variables for product B100-EIS	32
Figure 31:	Variables for product B100-PNS	33
Figure 32:	Opening folder in Windows Explorer	34
Figure 33:	Product Build directory in Windows Explorer.....	35
Figure 34:	Unzipped content folder.....	35
Figure 35:	Locally displayed home page of B100-EIS product.....	36
Figure 36:	Assignment of HTML pages to user groups in Web Builder.....	38
Figure 37:	Example of web page featuring netPROXY objects	40
Figure 38:	Labelling object for WebServer access	41
Figure 39:	Adding WebServer element to product and add label to filter	41
Figure 40:	neticiot.js Java Script file	42

Figure 41: Code in JavaScript for reading netPROXY object elements 43

Figure 42: Code in JavaScript for writing netPROXY object elements 43

Figure 43: Code in JavaScript for reading netPROXY object descriptions 44

List of Tables

Table 1:	List of revisions	3
Table 2:	Access rights of user groups (default)	39

Contacts

HEADQUARTERS

Germany

Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
D-65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: info@hilscher.com

Support

Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

SUBSIDIARIES

China

Hilscher Systemautomation (Shanghai) Co. Ltd.
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support

Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France

Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support

Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

India

Hilscher India Pvt. Ltd.
Pune
Phone: +91 8888 750 777
E-Mail: info@hilscher.in

Italy

Hilscher Italia S.r.l.
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support

Phone: +39 02 25007068
E-Mail: it.support@hilscher.com

Japan

Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support

Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Korea

Hilscher Korea Inc.
Seongnam, Gyeonggi, 463-400
Phone: +82 (0) 31-789-3715
E-Mail: info@hilscher.kr

Switzerland

Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support

Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA

Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support

Phone: +1 630-505-5301
E-Mail: us.support@hilscher.com