



**AutomationBrowser
Android Touch Panel Edition
Version 2**



iniNet . Erfolgreiche Kunden
E-mail: info@ininet.ch, www.ininet.ch, Tel: +41 61 -716 96 26

Contents

LICENSE AGREEMENT	3
ABBREVIATIONS	4
MMI	4
*.PRJ	4
VIEW	4
TEQ (*.TEQ)	4
PAINTER	4
PPO	4
CONTAINER	4
INSTALLATION	5
SPIDERCONTROL™ AUTOMATION BROWSER FOR ANDROID™	5
ATTENTION	5
INSTALLATION ON YOUR ANDROID TOUCH PANEL	5
FOR INDUSTRIAL TOUCH PANEL.....	5
WHY THE AUTOMATIONBROWSER?	7
FIRST STEPS	9
ANDROID SETTINGS:	11
FULL-SCREEN:	11
IMMERSIVE-MODE:	11
NAVIGATION BAR:	11
AUTO-SCALE:	11
PAN & ZOOM:	12
ANTI-ALIAS:	12
ANDROID KEYPAD:	12
EXPORT:	12
IMPORT:	13
PIN CODE:	13
CLOSE:	13
EXTENDING THE AUTOMATIONBROWSER	14
AUTOMATIONBROWSER WITH SEAMLESS NAVIGATION FROM PLC TO CAMERA	14
AUTOMATIONBROWSER WITH PROGRAMMABLE LOGIC TO JUMP, BLOCK TOUCH AND SWITCH BACKLIGHT	14
HTTP/HTTPS	15
VNC URL	16
RTSP URL FOR VIDEO STREAMING	17
AUTO-START MODE	18
SINGLE STATION MODE	18
FRAMESET IN HTML FILE	20
HOME SCREEN	21
CODESYS MICROBROWSER FAQ	22

License Agreement

READ THE TERMS OF THIS AGREEMENT AND ANY PROVIDED SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT") CAREFULLY BEFORE OPENING THE SOFTWARE MEDIA PACKAGE. BY OPENING THE SOFTWARE MEDIA PACKAGE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THESE TERMS, PROMPTLY RETURN THE UNUSED SOFTWARE TO YOUR PLACE OF PURCHASE.

1. LICENSE TO USE

ININET GRANTS YOU A NON-EXCLUSIVE AND NON-TRANSFERABLE LICENSE LIMITED TO THE INTERNAL USE OF THE ACCOMPANYING SOFTWARE AND DOCUMENTATION AND ANY ERROR CORRECTIONS PROVIDED BY ININET SOLUTIONS GMBH (COLLECTIVELY "SOFTWARE").

2. GOVERNING LAW

THIS AGREEMENT SHALL BE GOVERNED BY THE LAWS OF KANTON BASELSTADT, SWITZERLAND.

3. DISCLAIMER OF WARRANTY

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OR MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH ININET SOLUTIONS MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. GOOD DATA PROCESSING PROCEDURE DICTATES THAT ANY PROGRAM BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE RELYING ON IT. THE USER MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM. ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

4. RESTRICTIONS

SOFTWARE IS CONFIDENTIAL AND COPYRIGHT. THE TITLE OF ANY SOFTWARE AND ALL ASSOCIATED INTELLECTUAL PROPERTY RIGHTS ARE RETAINED BY ININET SOLUTIONS GMBH AND/OR ITS LICENSORS. YOU MAY NOT MAKE COPIES OF SOFTWARE OTHER THAN A SINGLE COPY OF SOFTWARE FOR ARCHIVAL PURPOSES. YOU MAY NOT MODIFY, DECOMPILE, AND REVERSE ENGINEER SOFTWARE. SOFTWARE IS NOT DESIGNED OR LICENSED FOR USE IN ON-LINE CONTROL OF AIRCRAFT, AIR TRAFFIC, AIRCRAFT NAVIGATION OR AIRCRAFT COMMUNICATIONS; OR IN THE DESIGN, CONSTRUCTION, OPERATION OR MAINTENANCE OF ANY NUCLEAR FACILITY. YOU WARRANT THAT YOU WILL NOT USE SOFTWARE FOR THESE PURPOSES.

5. LIMITATION OF LIABILITY

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ININET SOLUTIONS GMBH OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF ININET SOLUTIONS GMBH HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

IN NO EVENT WILL ININET SOLUTIONS GMBH'S LIABILITY TO YOU, WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, EXCEED THE AMOUNT PAID BY YOU FOR SOFTWARE UNDER THIS AGREEMENT. THE FOREGOING LIMITATIONS WILL APPLY EVEN IF THE ABOVE STATED WARRANTY FAILS ITS ESSENTIAL PURPOSE.

6. TERMINATION

THIS AGREEMENT IS EFFECTIVE UNTIL TERMINATED. YOU MAY TERMINATE THIS AGREEMENT AT ANY TIME BY DESTROYING ALL COPIES OF SOFTWARE. THIS AGREEMENT WILL TERMINATE IMMEDIATELY WITHOUT NOTICE FROM ININET SOLUTIONS GMBH IF YOU FAIL TO COMPLY WITH ANY PROVISION OF THIS AGREEMENT. UPON TERMINATION, YOU MUST DESTROY ALL COPIES OF SOFTWARE.

Windows 10 is a registered trade mark of Microsoft Inc.

Mac OS is a registered trade mark of Apple Inc.

Pentium is a registered trade mark by Intel Inc.

CODESYS is a registered trade mark of CODESYS GmbH

Abbreviations

MMI

Man Machine Interface, e.g. a SpiderControl™ views displayed at a touch panel or browser.

***.prj**

File extension for a SpiderControl™ project file generated by the SpiderControl™ EDITOR.
A SpiderControl™ project comprises everything to form a MMI on a touch panel or in a browser.

View

A view is what the user of a MMI sees at one moment inside a window or a browser. A *.teq file implements a view.

TEQ (*.teq)

File extension for a SpiderControl™ view file generated by the SpiderControl™ EDITOR.

Painter

A painter is a graphic object, which is used by the SpiderControl™ EDITOR. This object is programmed in JAVA. Several painters were packed into a Applet which resides on a embedded system.

PPO

Stands for Process Point. A process point is a variable of the user application that should be made visible towards the MMI.

Container

A container is a local variable, which has a scope within the actual applet/view. Containers are used to exchange values between different painters in a view or between different views of the same applet

Installation

SpiderControl™ Automation Browser for Android™

The SpiderControl Automation Browser app for Android is available for download on this page:

<https://www.ininet.ch/public/MicroBrowser/Android/automb.html>

This Browser is intended to be used on Industrial Touch Panel with Android version 5.0 (Lollipop) and higher, for ARM or x86 processor. The SpiderControl Automation Browser app supports web visualisation designed with SpiderControl Editor or any of the OEM editors, CODESYS V.2, CODESYS V.3, but it can also open any standard HTML 5 page.

Attention

Please, pay special attention that this edition of the Browser is not intended to be used by customers on their smartphone or tablet from public market. Since this app requires an Activation code, depending on the Hardware. An update of your Android may cause the loss of the license key. Also, when you change your phone, you will have to buy a new license. Therefore, we suggest to install the SpiderControl MicroBrowser app from Play™ Store with your Google account, for this purpose.

Installation on your Android Touch Panel

Open this page on your Android device with a standard web browser, and click on the droid icon.

Download SpiderControl Automation Browser (Open link above)

Once the download is finished, open the APK file stored in your Download folder and follow instructions to complete the installation.

Note: Some authorizations can be asked to allow the installation of an APK file out from the official store.

For Industrial Touch Panel

If you wish the Automation Browser to be the main app of your device, then Automation Browser Home Screen edition is the right choice. It will automatically start on reboot and every time you press the Home button it will show the app again.

Download SpiderControl Automation Browser (Open link above)

After installation, when you press the Android Home button, you will be asked to choose for the default launcher(*). Press the Home button again to see the ALWAYS option.

Once you have chosen the Automation Browser (ALWAYS), it will automatically be launched on the next start and you will not see the default Android Home screen anymore! To give you the choice again, you will have to clear the "Open defaults" setting in the Automation Browser app

Open the Android Settings (app menu in the top right corner) and depending on the Android version you will have to go

Apps > Automation Browser > Open by default > CLEAR DEFAULTS
Storage & USB > Apps > Automation Browser > (i) icon > Open by default > CLEAR DEFAULTS

Then, press on the Android Home button again.

(*) If you don't see the popup to select the default launcher, it is probably because another launcher is already selected as default. In that case you will have to clear the "Open defaults" settings in the current launcher app. The name of the app is often something like "Launcher" or "Google Now Launcher", ...

Why the AutomationBrowser?

Brief Summary

- Support Panel Operation, Kiosk Mode
- One Browser always works, also with legacy HMI which is no longer supported in standard Browser
- Easy application installation, setup and maintenance
- The user is restricted to the desired URL
- The user has a list of all available PLC's/servers
- No exit to OS level, but allow for certain OS configurations (e.g. IP address)
- Interact with additional input devices: On-Screen-Keyboard, RFID, Scanner
- Have a better performance with CODESYS V3.x clients even on slower HW
- Allow for remote controlled functions: PLC can switch on light on panel
- Involve and control other programs on panel without leaving the UI context
- Offer a platform for I4.0/IIoT integration which can be used by factory level staff (no IT pro's needed)

Detailed Description

Web-based HMIs are today standard in automation for operation and monitoring. Through the available browsers – on a wide variety of devices –

Web technologies offer a high degree of simplification and modularity in the development of graphical user interfaces. The same technology also allows operation in the operator panel, on the PC, smartphone or tablet. But: so far, so good – where are still unresolved problems today? Web-based HMIs can be easily stored directly on controllers, because integrated embedded Web servers are available almost everywhere today.

A currently major obstacle are older Web HMIs, which are used on many installed controllers that are based on Java Applets and are no longer supported by popular browsers. These include, for example CODESYS Webvisu V2.x or even older SpiderControl™ OEM versions on PLC by Phoenix Contact, SAIA-Burgess, Panasonic and many more.

Another problem will arise when you have several automation systems in one system and the operator station has to jump back and forth between the various Web servers, so that the operator can even see all relevant information there. From technical point of view, this is not a problem. To switch from one server to another, the stored URL link will do. In practice, however, this can be complex and often problematic. That means you have to deposit the possible URL jumps in the HMIs and on all Web servers in advance. It is also a considerable effort and in some cases it's not possible at all, if the Web HMI was developed by a third company. Another problem will be the log-ins.

Usually there are several user levels in an operation for which one has to identify first. But if you jump from one Web server to another, this information is lost and the login procedure will start again. Another important issue is pop-up messages that should be displayed to the user immediately if there is a problem with the system. However, if this happens while the panel displays HMI from another Web server, the user does not get this error. The SpiderControl™

AutomationBrowser by iniNet Solutions is developed to solve this problems and contains also other useful functionalities for automation. For example

a so-called station list can be created directly in the AutomationBrowser, in which all connected controllers are stored with their URL. This list can be displayed at any time, even if the browser is currently located on an HTML page of a controller. It is not necessary to make any changes to existing Web HMIs of controllers. The difference to a „favorites“ list – as well known from a conventional browser: as the browser on the panel is expected to operate in „kiosk mode“ (full screen), it cannot access all normal functions of the browser menu. Only the station selection and, if necessary, a “back” button will be displayed. Also, the station list can be displayed in a touch-screen-friendly, large format in order to increase the user-usability. This start page can even be customized by a programmer. To solve the problem of compatibility with older web HMIs, several browsers have been integrated into AutomationBrowser. Invisible to the user, the AutomationBrowser simply selects the right viewer by itself, which makes it easy to integrate older Web visualizations like CODESYS V2.

.

First Steps

After the successful installation, the following Window appears:

Automation Browser

Stations

- Local HMI
http://127.0.0.1:8080/
- WebVisu
http://192.168.1.1/plc/webvisu.htm

Local HMI

Control Unit

Local HMI
http://127.0.0.1:8080/

Connect

Auto Start

9

Choose one station from the list to be opened on startup. Button turns green for this station

www.spidercontrol.net
Version: 1.0.8.4

Automation Browser

Stations

- Local HMI
http://127.0.0.1:8080/
- WebVisu
http://192.168.1.1/plc/webvisu.htm

Local HMI

Control Unit

Local HMI
http://127.0.0.1:8080/

Auto Detect

Auto Start (10s)

Select whether the AutomationBrowser should use HTML5 client or the MicroBrowser. Toggle between AutoDetect/HTML5/MicroBrowser mode

Select one station from the list to start automatically. Toggle different delays to wait before doing so (e.g. wait until PLC has booted)

www.spidercontrol.net
Version: 1.0.8.4

The AutomationBrowser contains MicroBrowser and Chromium

The MicroBrowser is able to display

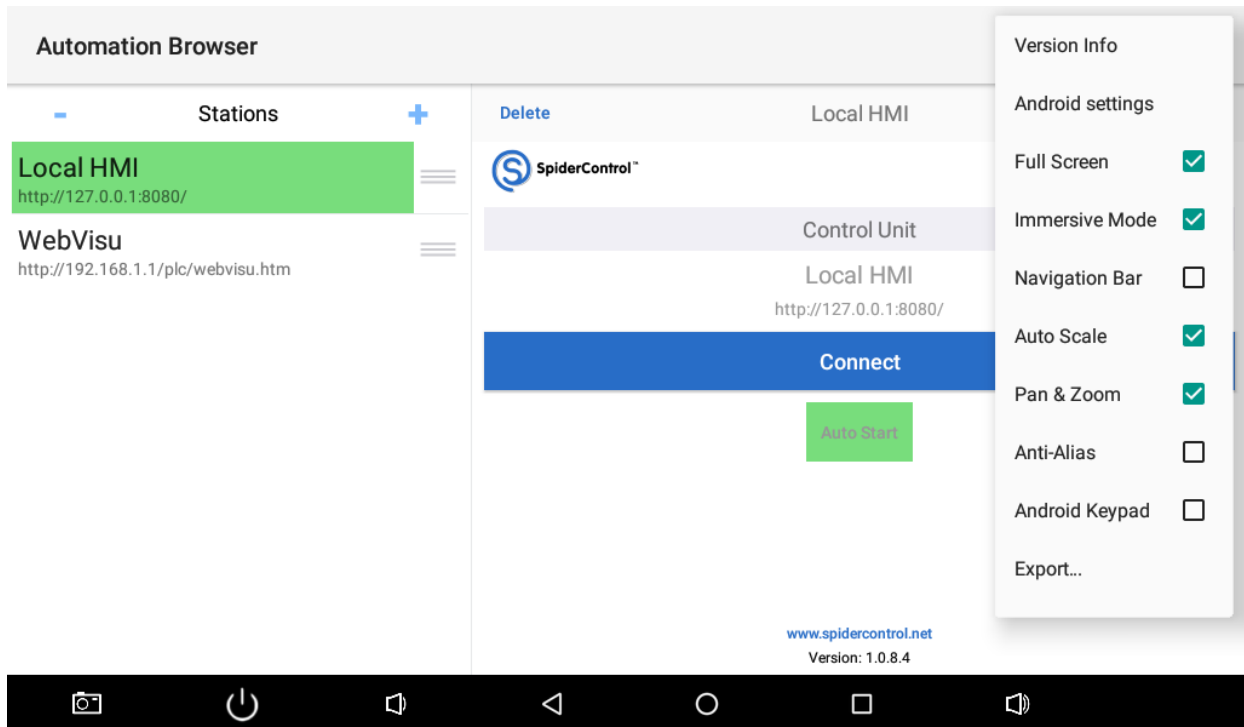
- All Web-HMI which are designed with SpiderControl PC HMI Editor or an OEM version of it
- CODESYS WebVisu V2.3
- CODESYS WebVisu V3.x

For all other content, the AutomationBrowser will use the integrated Chromium HTML5 Web-Client.

When opening an URL, the AutomationBrowser will first analyze the HTML page and then open automatically either the MicroBrowser or Chromium (when in Auto Mode). Older SpiderControl HMI projects as well as the CODESYS Webvisu V2.3 used Java Applets, which are no longer supported in any Browser. The MicroBrowser is able to display these HMI without a Java VM by using a native implementation. CODESYS Webvisu V3.x can be visualized both with the MicroBrowser as well as with Chromium HTML5. The MicroBrowser offers better performance as well as other possibilities, so for this type of HMI the AutomationBrowser will open Chromium when in Auto-Mode or the MicroBrowser when forced to MicroBrowser Mode.

When you display a newer HTML5 based SpiderControl HMI, the Auto mode will open the MicroBrowser, but you can force it to use Chromium by selecting the HTML5 mode.

Main Menu (Upper right 3 dots)



Android Settings:

Enter Android Settings to modify IP Address and other network related parameters. This is useful when the AutomationBrowser is in Home Screen mode and the user has no access to other Apps.

Full-screen:

Full-screen is only used to remove the status bar in the top of the screen

Immersive-Mode:

Immersive-Mode is used to remove both status bar and the task bar in the bottom of the screen (restore task bar with a swipe up from the bottom of the screen)

Navigation Bar:

This is only used in HTML 5 mode to show a control bar in the top of the screen, to navigate with previous and next buttons, to refresh the view or to come back in the home view

Auto-scale:

- MicroBrowser: The view will automatically be re-scaled to fit the screen (isotropic, keep width/height ratio)
- HTML 5: It's not really re-scaled on the view, since it depends on the HTML page. But it will activate an option of the WebView control to fit the screen.

Pan & Zoom:

- MicroBrowser: enable/disable panning and zooming of the view.
- HTML 5: Unused, this option is handled in the HTML code

Anti-Alias:

- MicroBrowser: Improve the rendering with anti-alias if the device does not already support this feature in hardware accelerator. In most devices we don't need to activate this option.
- HTML 5: Unused

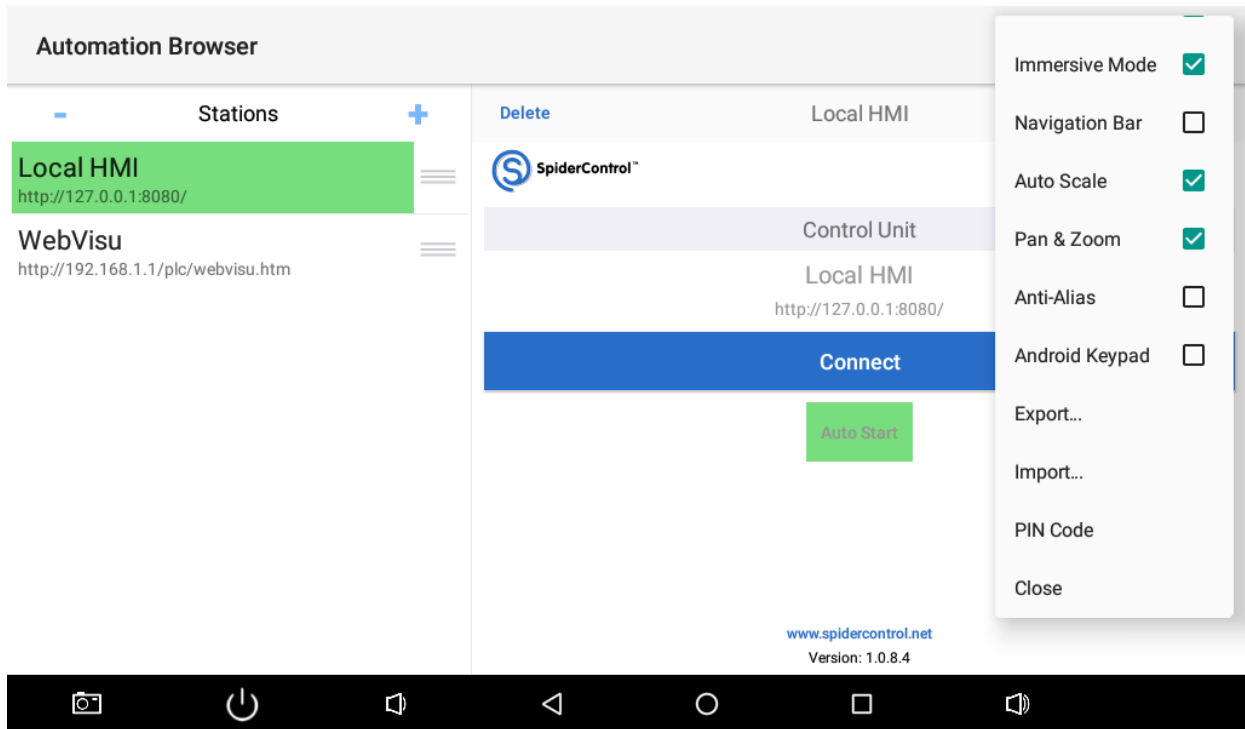
Android Keypad:

- MicroBrowser: Either show the Android Keypad to edit a value or use the Keypad/alphapad TEQ files
- HTML 5: Unused

Export:

Export station list in the into the
Download/AutomationBrowser/MB_STATION.xml

Scroll the menu list for more...



Import:

Import station list from Download/AutomationBrowser/MB_STATION.xml

PIN Code:

PIN code is used to prevent any modification from operator user. The password is needed to change anything. This feature allows for locking up the operator panel to avoid a user to modify settings or to exit the App ('Kiosk Mode')

Close:

Close the menu

Extending the AutomationBrowser

The version that you have installed so far covers the basic functionality. If you need a customized behaviour, the AutomationBrowser can be extended with the SpiderPLC components. In the following, we present two examples of such extensions. If you have an application which could be solved using this approach, please contact us.

AutomationBrowser with seamless navigation from PLC to camera

The SpiderControl AutomationBrowser allows for a seamless navigation from a HTML5 Webvisu on a PLC to a web-cam which is linked directly with its H264 / rtsp:// URL (which is supported by almost any webcam). A fully customized user interface in the control panel offers the option of providing a self-contained, consistent operation.

<https://www.youtube.com/watch?v=ohQA5tl2A8E>

AutomationBrowser with programmable logic to jump, block touch and switch backlight

The SpiderControl AutomationBrowser can be programmed with an integrated functional block logic to force a jump to a specific URL, to block the touch screen or to switch on/off the back light. The integrated SpiderPLC can be programmed with any standard Browser and connects to external PLC using standard protocols like OPC UA, Modbus, ISO-on-TCP and more. Like this, a Web-Panel can be used to display several Web-HMI's, but still be controlled by a connected PLC.

<https://www.youtube.com/watch?v=2klVhvjNuk8>

HTTP/HTTPS

The Automation Browser supports both http and https URL, with or without credentials.

```
http://[user:password@]hostname/...  
https://[user:password@]hostname/...
```

If credentials are not specified in the URL and the server prompts for HTTP authentication, it will show an authentication dialog, to enter the username and password.

If you choose "Save Password" in this dialog, it will still show the authentication dialog the next time, to allow the user to use the stored information or enter another username/password.

If credentials are specified in the URL, the HTTP authentication dialog will not be displayed, unless the credentials are wrong.

In that case, you will have to update or clear the credentials from the URL.

If the server uses as untrusted SSL certificate or self-made certificate, the user will be prompted to accept it and continue loading the page, or not.

Press YES (ALWAYS) to permanently save your choice.

Press CLEAR DATA in the Android Settings of the app, to restore the default settings and clear all stored credentials.

VNC URL

VNC URL is something like vnc://192.168.1.123/

vnc://hostname[:port]/[bpp[.depth]]/[password]

Optional parameters:

- port, default is 5900
- bpp is either 8, 16 (565) or 32 (888), 0 use default parameter from server
- depth is optional and depends on the bpp. Supported values are 16.15 (555), 8.6 (64 colors) or 8.3 (8 colors)
- password if need must be defined after the pixel format, to use default pixel format, use bpp 0
- If supported on the VNC server side, the best choice is 16 bpp (565), for example

vnc://192.168.1.2/16/password

Notes:

- A Floating Button is used to open and close the Android Keypad,
- The Floating Button is visible only if Android Keypad option is enabled in the menu.
- A long press on the Floating Button moves the button.
- The Auto-Start in MicroBrowser mode is recommended to be used instead of Auto detection
- If the connection is lost and the Auto Start is defined with a delay (3s or more), it will return to the countdown page

RTSP URL for video streaming

```
rtsp://[user:password@]hostname/[live0][?caching=MILLISECONDS]
```

Depending on the camera you will have to specify /live0, /live1, ... or something like that, after the hostname

The optional network caching parameter in the rtsp URL, allows you to reduce the latency from the live stream, but may cause artifacts or refresh issues if the parameter is too short

The default value is 200 ms, for example, to set 50 ms

```
rtsp://192.168.1.123/live0?caching=50
```

Auto-Start mode

* Without countdown (start immediately)

- 1) Auto start with auto detection will shortly show the setup page and open either the MicroBrowser or HTML5 view
- 2) Auto start with MicroBrowser will directly jump to the MicroBrowser if the server is reachable (*)
- 3) Auto start with HTML5 will directly jump to the WebView, if the server is reachable

(*) If the server is not reachable or if the connection is lost, it will jump to the setup page.

* With a countdown of N seconds (Auto Start 3s, 10s, 15s, 30s, 45s, 60s, 90s or 120s) on startup

- 1) Auto start with auto detection will shortly show the setup page and either open the MicroBrowser or HTML5 view after a delay of N seconds
- 2) Auto start with MicroBrowser will jump to the MicroBrowser view if the server is reachable after a delay of N seconds (**)
- 3) Auto start with HTML5 will jump to the WebView, if the server is reachable after a delay of N seconds

(**) If the server is not reachable or if the connection is lost, it will do endless retries every 10 seconds.

To abort the endless retries and return to the setup page, press 5 times in the top-left corner.

Or press the back button if available.

Instead of showing the loading message "Loading...", you can set a boot image to show on startup, loaded from
/sdcard/Download/bootscreen.png

Single station mode

Single station mode is used to directly start the specified URL in the XML app setting file, skipping the setup page.

It's a mode where the end user will not see the setup page.

File: automb.xml

```
<settings>
  <name>My Visu</name>
  <start_url>http://localhost/Visu.html</start_url>
  <start_mode>MicroBrowser</start_mode>
  <auto_start_delay_sec>3</auto_start_delay_sec>
  <use_android_keypad>>false</use_android_keypad>
</settings>
```

--> Procedure with loading file from External Storage

- 1) Export files from Automation Browser into external storage (USB, SD card, ...)
- 2) Copy automb.xml into:
 <ExternalStorage>/Download/AutomationBrowser/automb.xml
(for Android < 10)

<ExternalStorage>/Android/data/net.spidercontrol.automb/files/automb.xml (for Android >= 10)

(The folder should already exist on the external storage)

3) Import files in Automation Browser (menu)

Once the XML file has been loaded with <start_url>, it will automatically start with the specified URL and settings.

Pressing back button will close the app.

In this mode, you don't see the standard station list, menu, config, etc...

To restore the standard mode, remove the XML file from external storage (or remove the external storage)

--> Procedure with loading file from Internal Storage

1) Export files from Automation Browser into internal storage

2) Copy automb.xml into:

<InternalStorage>/Download/AutomationBrowser/automb.xml

(for Android < 10)

<InternalStorage>/Android/data/net.spidercontrol.automb/files/automb.xml (for Android >= 10)

(The folder should already exist on the internal storage)

3) Import files in Automation Browser (menu)

To restore the standard mode, remove the XML file (automb.xml) from internal storage

All settings that can be defined in XML file:

```
<settings>
  <name>My Visu</name>
  <start_url>http://192.168.1.123/Visu.html</start_url>
  <start_mode>MicroBrowser</start_mode>
  <auto_start_delay_sec>3</auto_start_delay_sec>
  <show_navig_bar>false</show_navig_bar>
  <full_screen>true</full_screen>
  <immersive>false</immersive>
  <auto_scale>true</auto_scale>
  <pan_zoom>true</pan_zoom>
  <use_android_keypad>false</use_android_keypad>
  <use_external_storage>false</use_external_storage>
  <pin_code></pin_code>
</settings>
```

<start_mode> is either an integer or a string

0: Auto (not supported for auto start)

1: HTML5

2: MicroBrowser

Frameset in HTML file

Only few configurations are supported to see either 2, 3, 4 or 6 frames
An URL can be defined for each frame, either a http URL for HTML5
WebView or a RTSP URL for video streaming,
something like:

File: frameset2.html

```
<FRAMESET cols="50%,50%">
  <FRAME src="http://192.168.1.20/html/frame1.html">
  <FRAME src="http://192.168.1.20/html/frame2.html">
</FRAMESET>
```

File: frameset4.html

```
<FRAMESET rows="50%,50%" cols="50%,50%">
  <FRAME src="http://192.168.1.20/html/frame1.html">
  <FRAME src="http://192.168.1.20/html/frame2.html">
  <FRAME src="rtsp://192.168.1.10/live0">
  <FRAME src="rtsp://192.168.1.11/live1">
</FRAMESET>
```

File: frameset3.html

```
<FRAMESET cols="50%,50%">
  <FRAME src="http://192.168.1.20/Visu.html">
  <FRAMESET rows="50%,50%">
    <FRAME src="rtsp://192.168.1.5:8554/live1">
    <FRAME src="rtsp://192.168.1.6:8554/live1">
  </FRAMESET>
</FRAMESET>
```

File: frameset6.html

```
<FRAMESET rows="50%,50%" cols="33%,33%,33%">
  <FRAME src="rtsp://192.168.1.101/live1">
  <FRAME src="rtsp://192.168.1.102/live1">
  <FRAME src="rtsp://192.168.1.103/live1">
  <FRAME src="rtsp://192.168.1.104/live1">
  <FRAME src="rtsp://192.168.1.105/live1">
  <FRAME src="rtsp://192.168.1.106/live1">
</FRAMESET>
```

Note: frame size in px or percent are not supported yet

Home screen

After having installed the Automation Browser Home Screen edition, press the Android Home button. So, you will be asked to choose for the default launcher (*). Press the Home button again to see the ALWAYS option.

Once you have chosen the Automation Browser (ALWAYS), it will automatically be launched on the next start and you will no longer see the default Android Home screen (Desktop)!

To find this option again, you will have to clear the "Open defaults" setting of the Automation Browser app

Open Android Settings (app menu in the top right corner) and depending on Android version you will have to go either

- * Apps > Automation Browser > Open by default > CLEAR DEFAULTS

- * Storage & USB > Apps > Automation Browser > (i) icon > Open by default > CLEAR DEFAULTS

Then, press on the Android Home button again.

(*) If you don't see the popup to select the default launcher, it is probably because another launcher is already selected as default.

In that case you will have to clear the "Open defaults" settings in the current launcher app. The name of the app is often something like "Launcher" or "Google Now Launcher", ...

CODESYS MicroBrowser FAQ

Following questions are being asked often concerning the MicroBrowser CODESYS.

I can't connect! It says "File Not Found!"

This most often has 2 possible causes:

1. Older versions of the MicroBrowser CODESYS (prior to 1.5.15.116) don't support the option "compressed webvisu".

2. Some CODESYS PLC's are case sensitive. Older versions of MicroBrowser CODESYS get the entry page with the name "PLC_VISU.xml" written in big letters. But the file on the webserver is written "plc_visu.xml". Possible workaround: Modify the "webvisu.htm" File on your PLC and change this line:

```
<param name="STARTVISU" value="PLC_VISU">  
to  
<param name="STARTVISU" value="plc_visu">
```

Why are arrays not displayed correctly?

Normally arrays are not working with the applet from CODESYS itself. But with our MicroBrowser CODESYS it is possible, BUT the indexed variants of the needed array elements must also exist in the view, because otherwise the variable address is unknown.

Example:

".g_afb_GF[g_index].i_bo_configured" is an indexed variant which can be read and written with MicroBrowser CODESYS, but you must already know during build-time of your project, which indexes are used, and you also have to include the resolved variants of the variable in the view: ".g_afb_GF[6].i_bo_Configured" (for instance in a hidden text-field).