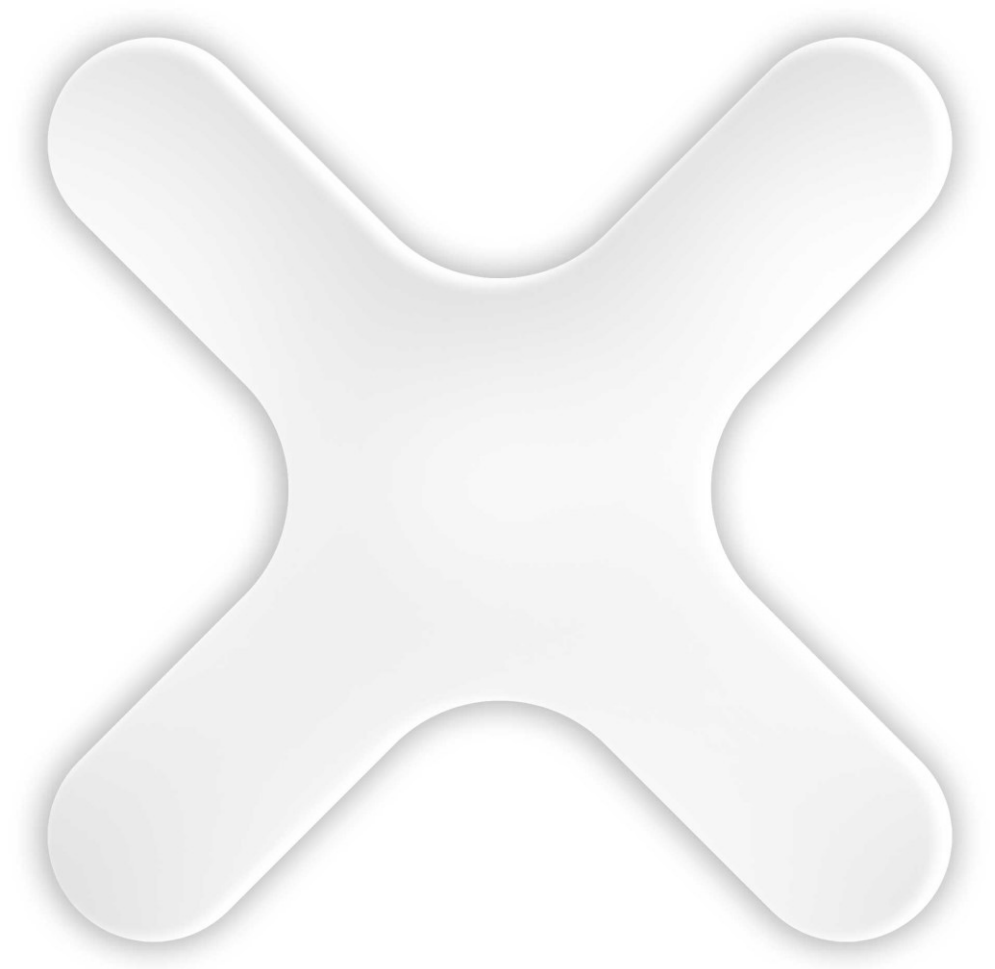


JMobile Developer Day

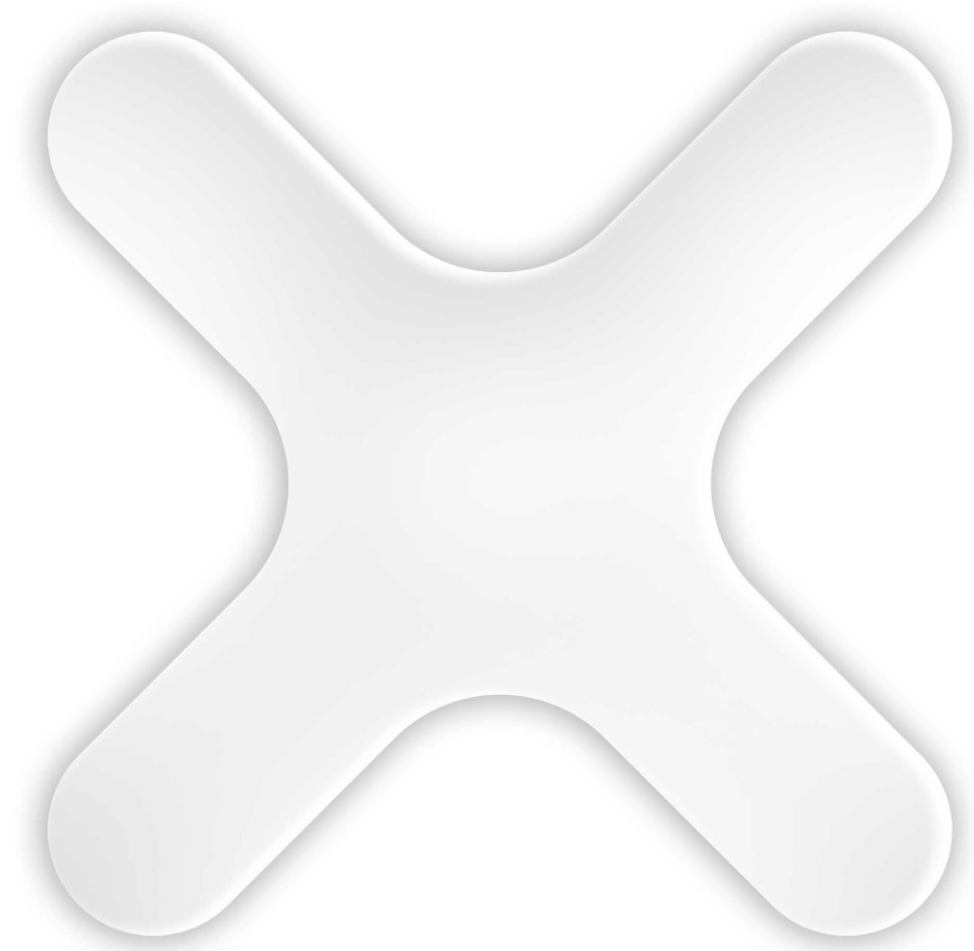


JMobile 4.5 SP1

Preliminary improvements overview

JMobile[®]

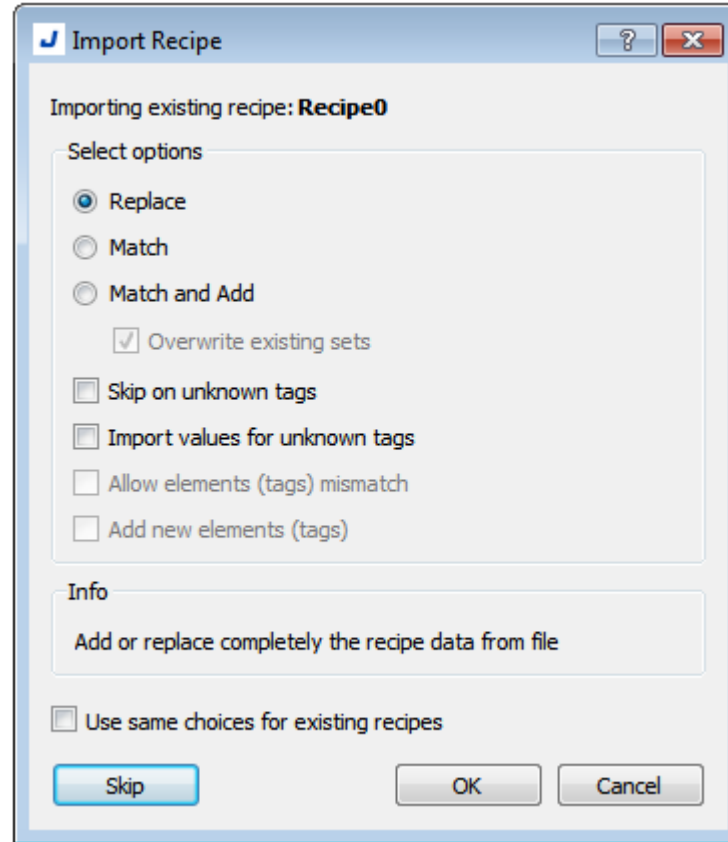
EXOS



Recipes



Improved importing management



Different element numbers
between HMI and file

Full Unicode Support for Recipe data

index	Element Name	Tag	@à123	18893'@#	2134_^?=#
0	Element0	Tag3	2	0	0
1	Element1	Tag4	0	0	1
2	Element2	Tag2	0	0	0

Recipe : _RecipeMgr	
Recipe Name	Recipe0
Number of set	3
Set 0	@à123
Set 1	18893'@#
Set 2	2134_^?=#

Multilanguage



Improved translations handling

Fast copy and paste translations

Quick search and filters

View only user texts

Export texts one time only

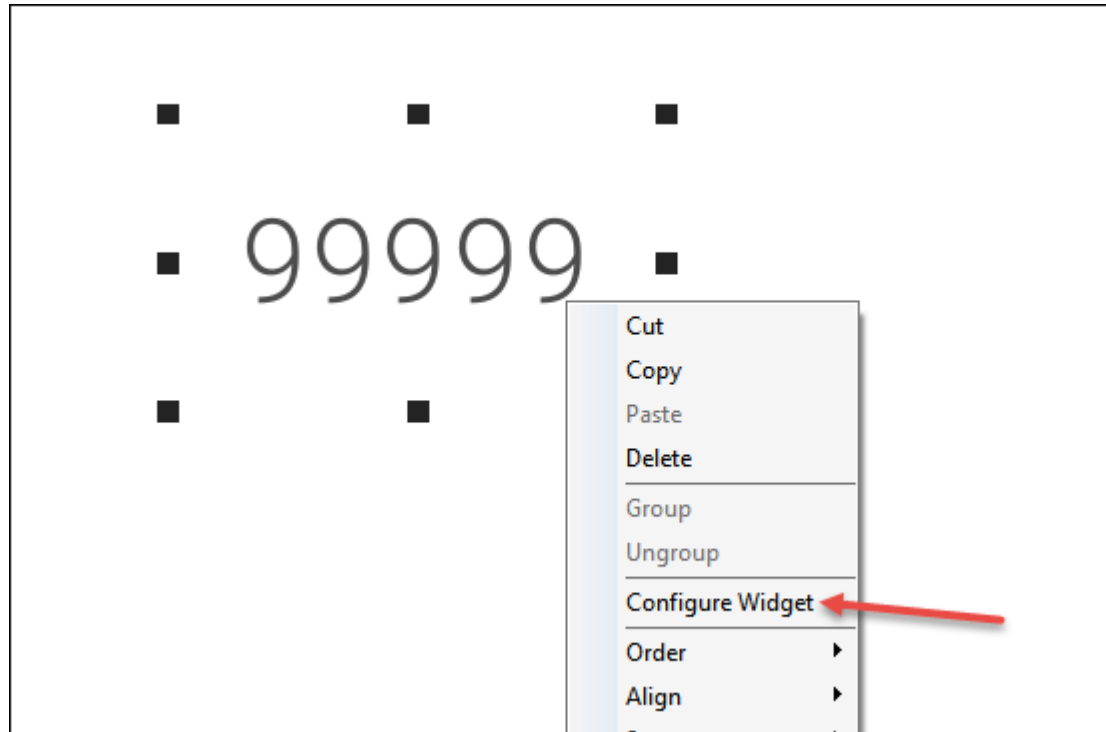
Multiselection to export only relevant texts

Page	Widgetid	ita	
LoginPage_combo.jmx	error_msg:messages.14	Warning: Your password will expire soon	Warning: Your password
LoginPage_combo.jmx	error_msg:messages.15	Error : User is set Inactive. Please contact administrator	Error : User is set Inacti
LoginPage_combo.jmx	error_msg:messages.27	Error : Username or password is wrong. Login locked for a while	Error : Username or pas
LoginPage_combo.jmx	error_msg:messages.0		
Page2.jmx	label6:text	a	b
Page2.jmx	label7:text	a	b
Page2.jmx	label8:text	ciao	hello
Project102.jpr	_TrendMgr:Timestamp	Timestamp	Timestamp
Project102.jpr	_TrendMgr:Time	Time	Time
Project102.jpr	_TrendMgr:Quality	Quality	Quality
Project102.jpr	_TrendMgr:Date	Date	Date
Project102.jpr	_AlarmsMgr:Not Triggered	Not Triggered	Not Triggered
Project102.jpr	_AlarmsMgr:Disabled	Disabled	Disabled
Project102.jpr	_AlarmsMgr:Triggered	Triggered	Triggered

Minor improvements

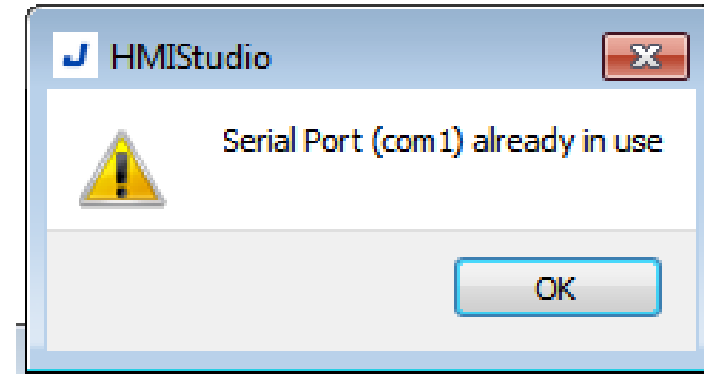
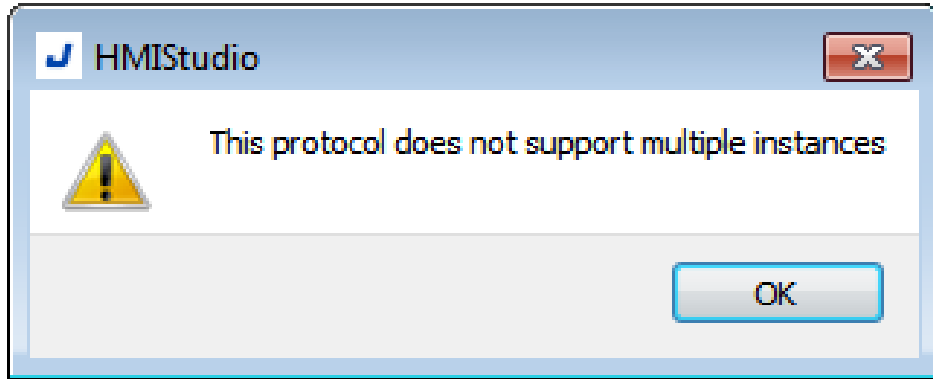


Create a custom widget from a single one



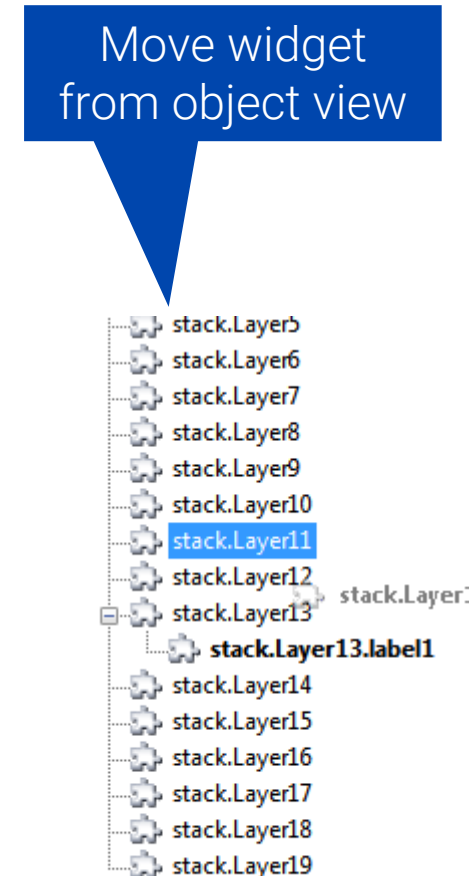
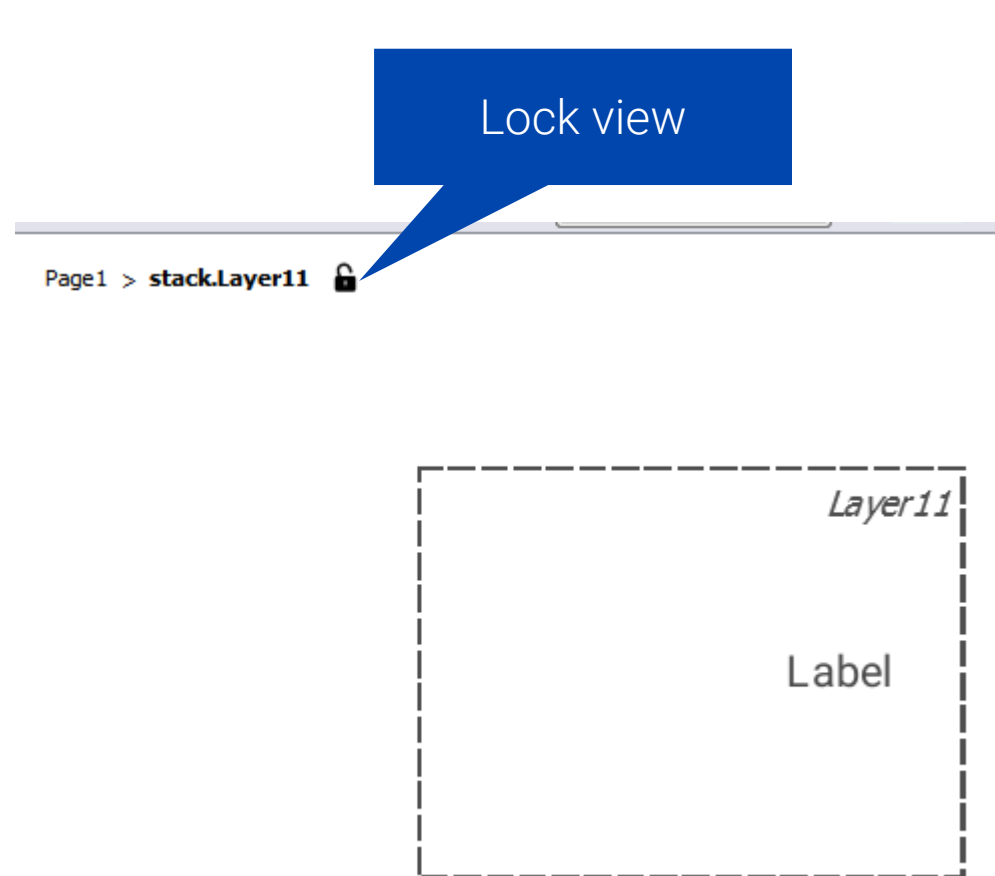
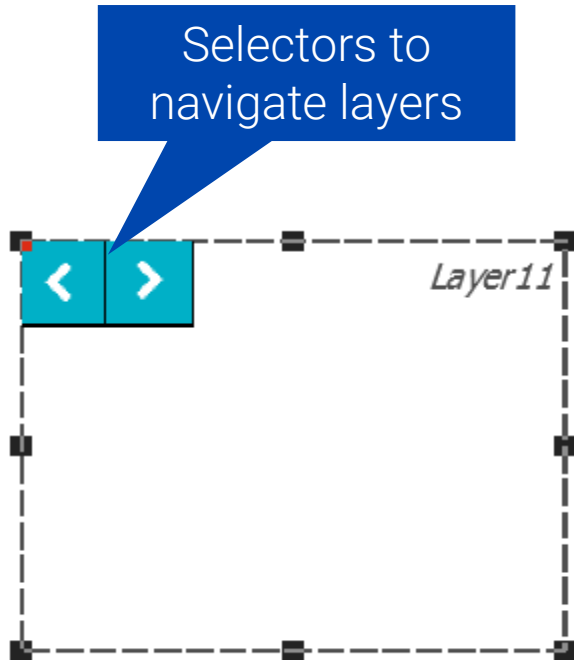
Create smart widgets with parameters

Studio control for already configured COM ports and protocols instances



Better catching of human errors

Improvements on stack widget usability

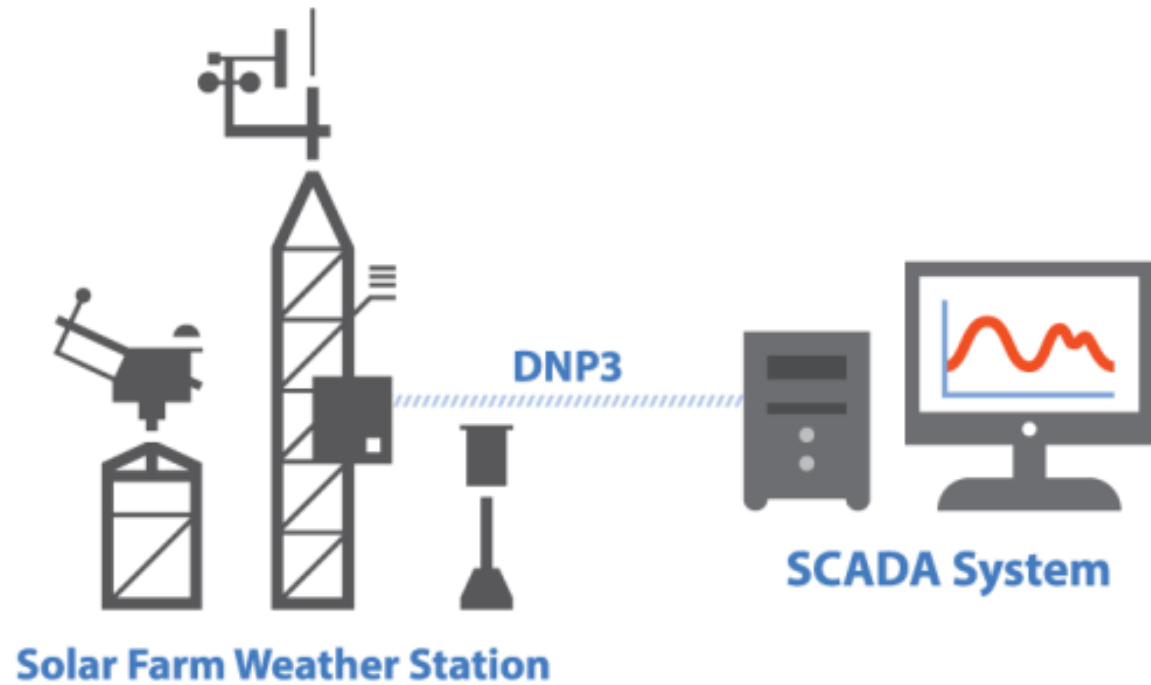


New protocols

EXOR



DNP3 Outstation now available for Linux HMIs



Used between components
in process automation
systems

CODESYS V3 HANDLER



CODESYS

Supports security and encryption

Fully compatible with CODESYS V3 ETH

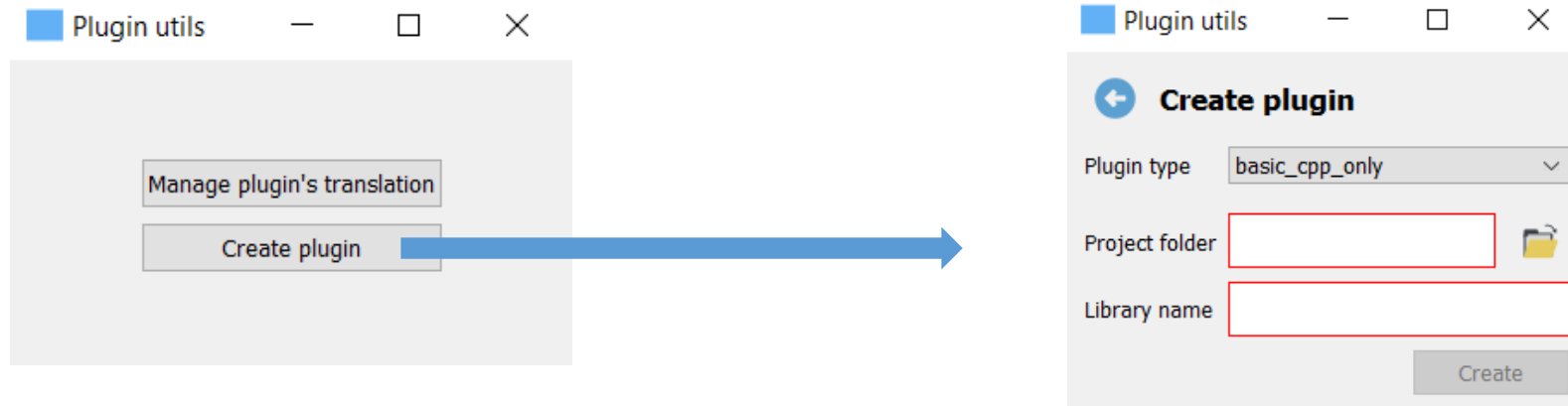
Open Framework:
CUSTOM BUILT WIDGET

Beta phase opened



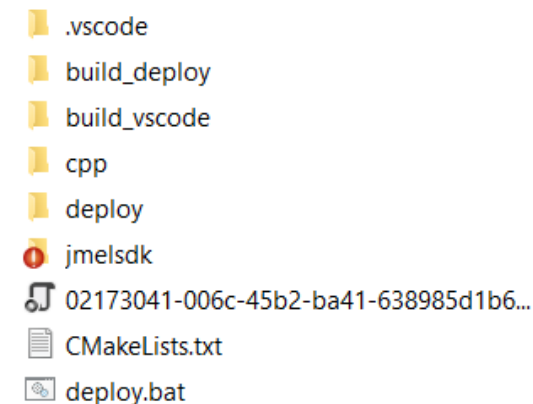
Custom Built Widgets (CBW) – Creation of new plugin

Using a [simply wizard](#) it's possible to create a new project of a custom built widget, giving the destination folder of the plugin and the name of the library. Indeed, a plugin represents a [library of widgets](#).

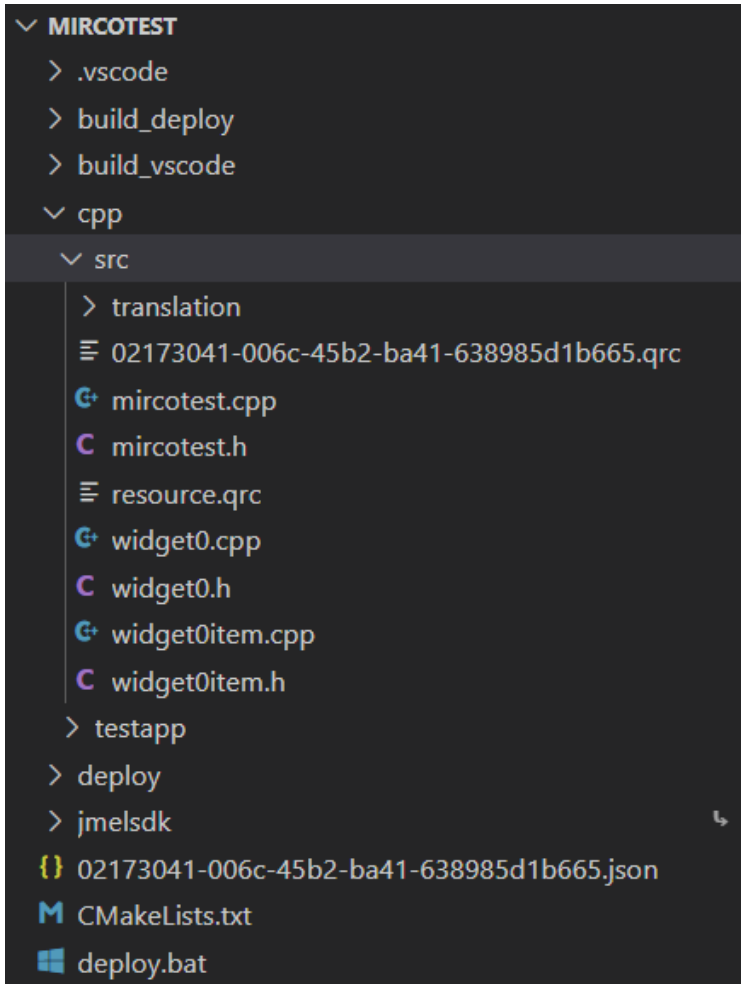


The output is directly a Visual Studio Code project with:

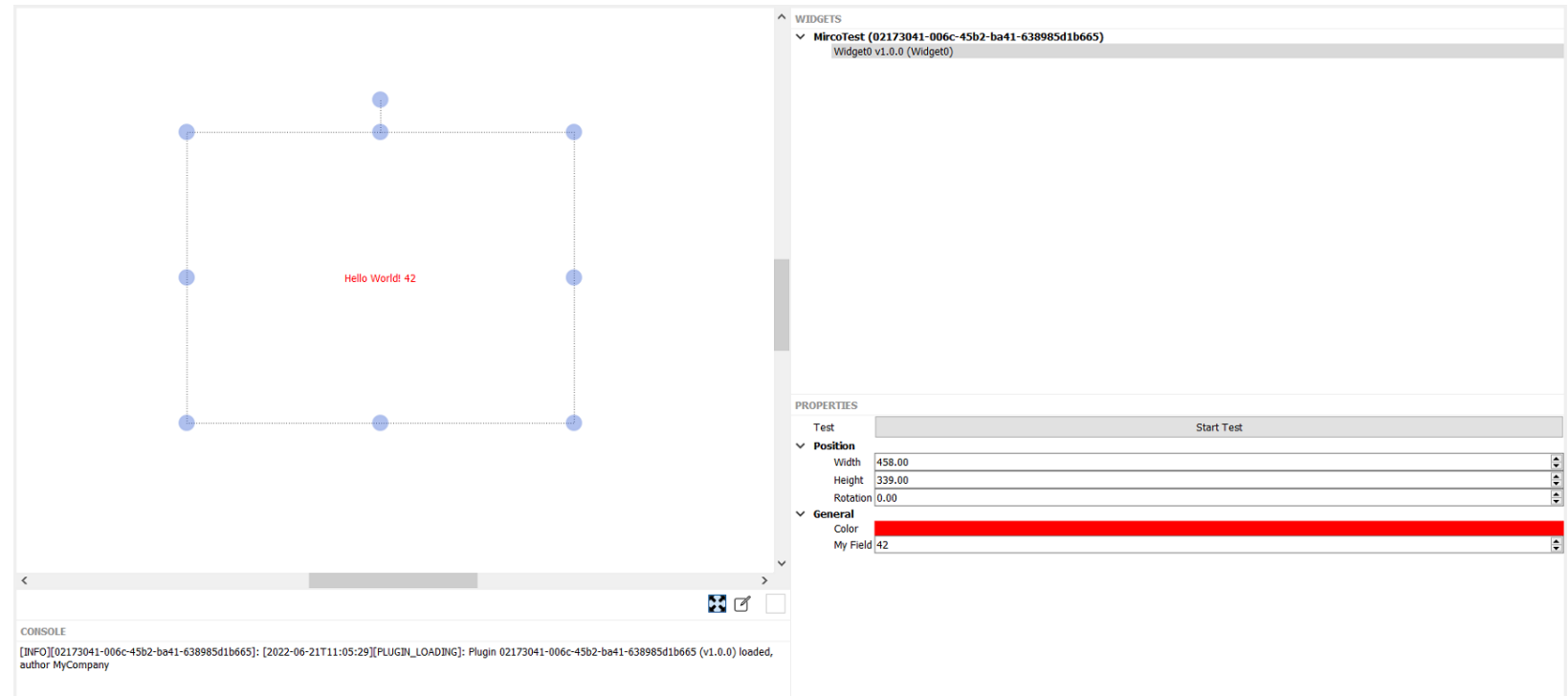
- a demo **Hello World** project as starting point for user
- all dependencies needed to directly build the CBW
- an integrated debug and test environment
- configuration for Visual Studio Code



Custom Built Widgets (CBW) – Build and test the demo plugin



Opening VS Code, we can build (based on [CMake](#)) the project and see it running in the debug environment (still independent from JM Suite)



Custom Built Widgets (CBW) – Add new properties and access them

Properties configuration is done by intuitive [JSON file](#), allowing easily add properties (like color property on the right) and change some general parameters like plugin name, version, translation languages, etc.

Online helper is provided for each field, helping developers a lot

```

"fields" : [
  {
    The default value of the field. If not specified the default value is black #000000.
    The color may be in one of these formats:
    #RGB
  },
  {
    #RRGGBB
    #AARRGGBB
    A name from the list of colors defined in the list
    https://www.w3.org/TR/SVG11/types.html#ColorKeywords
    "defaultValue": "#ff0000"
  }
]

```

Properties access is done by intuitive C++ API

```

{
  "name": "MircoTest",
  "id": "02173041-006c-45b2-ba41-638985d1b665",
  "author": "MyCompany",
  "version": "1.0.0",
  "translation": [ "de-DE", "fr-FR", "es-ES", "it-IT" ],
  "widgets": [
    {
      "id": "Widget0",
      "version": "1.0.0",
      "defaultSize": "100x100",
      "display": "Widget0",
      "description": "Widget0",
      "fields": [
        {
          "id": "myfield",
          "display": "My Field",
          "type": "integer",
          "defaultValue": 42
        },
        {
          "id": "color",
          "display": "Color",
          "type": "color",
          "defaultValue": "#ff0000"
        }
      ]
    }
  ]
}

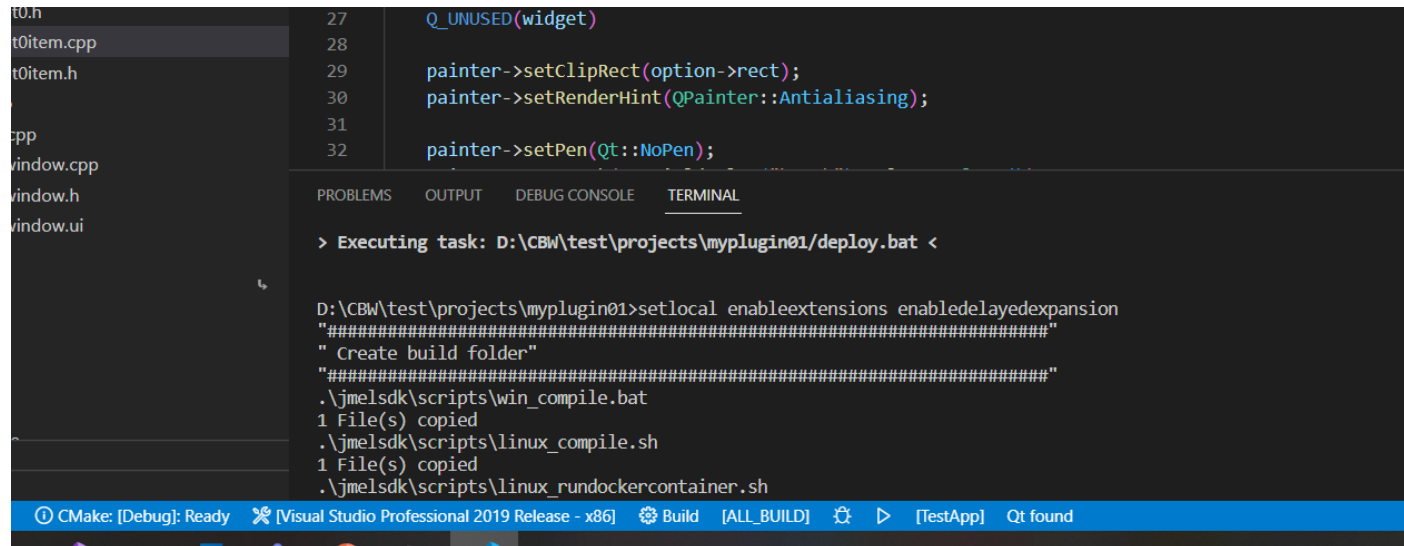
```

Custom Built Widgets (CBW) – Deploy into JMobile Suite

Deploy script allows to build plugin for x86/x64 (Studio and Runtime PC) and ARM32/64 Linux targets (our panels). This is achieved by [CMake](#) and [Docker containers](#), which embed all the toolchains and libraries (g++ compiler, Qt5, etc)

In future, could be potentially done by [Corvina](#), without needed to configure environment on local PC.

Deploy a plugin on JMobile suite means just copy the dynamic libraries created. The goal is to provide an [automatic installer](#) (like we do for protocols)



```

t0.h          27      Q_UNUSED(widget)
t0item.cpp    28
t0item.h      29      painter->setClipRect(option->rect);
               30      painter->setRenderHint(QPainter::Antialiasing);
               31
               32      painter->setPen(Qt::NoPen);
window.cpp
window.h
window.ui

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
> Executing task: D:\CBW\test\projects\myplugin01\deploy.bat <

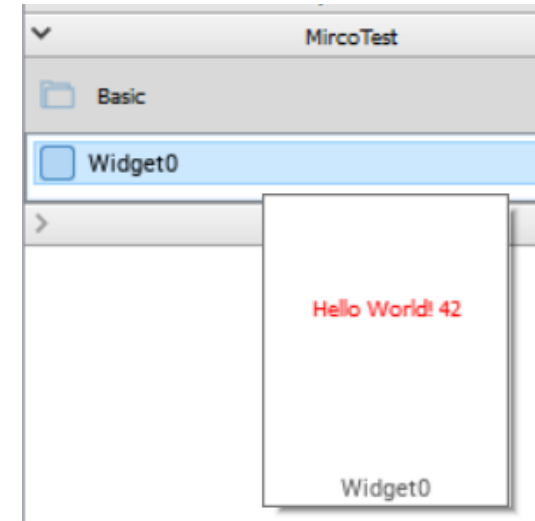
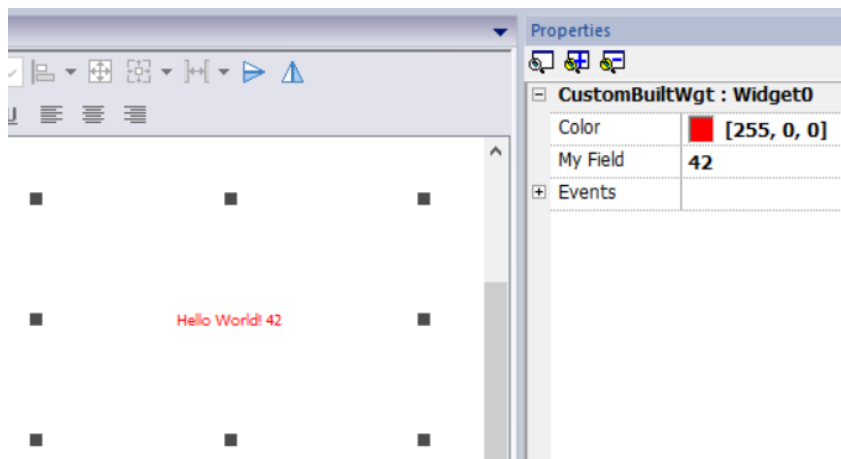
D:\CBW\test\projects\myplugin01>setlocal enableextensions enabledelayedexpansion
"#####"
" Create build folder"
"#####"
.\jmlsdk\scripts\win_compile.bat
1 File(s) copied
.\jmlsdk\scripts\linux_compile.sh
1 File(s) copied
.\jmlsdk\scripts\linux_rundockercontainer.sh

CMake: [Debug]: Ready  Visual Studio Professional 2019 Release - x86  Build [ALL_BUILD]  [TestApp] Qt found
```

Custom Build Widgets (CBW) – Deploy into JM Suite plugin

Deployed plugin is now a part of the current JM Studio gallery, with its category, name, icon and preview

CBW in page is like any other native JM Suite widget



Manager to enable/disable and see logs of plugins in Studio

