

# Application note

## Exporting AC500 tags for use with Panel Builder

AN00238

Rev A (EN)

Save significant time developing CP600 HMI applications by exporting tags from your AC500 PLC project and importing these directly into Panel Builder



### Introduction

The CP600 range of intelligent HMI panels is able to communicate with other peripherals (e.g. AC500 PLCs, ABB motion products) via a selection of communication protocols. If using the CP600 to communicate with an AC500 PLC via the ABB CoDeSys protocol it becomes possible to export tags (symbols) from the Automation Builder project and import these directly into the Panel Builder HMI application.

This process save significant time and reduces errors as there is no need to manually enter tag data into the CP600 project.

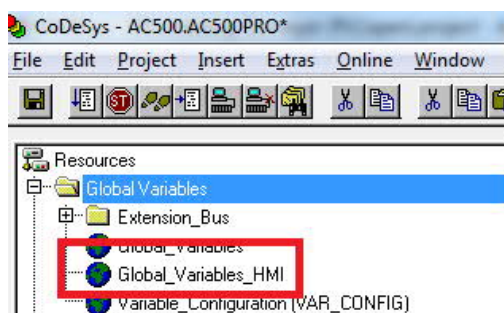
This application note details the export process, showing how to configure Automation Builder and the IEC programming environment as well as the import process, showing how to include the tags in a CP600 application.

Screenshots from Automation Builder version 1.1.2 and Panel Builder version 1.91.0 (432) are used throughout this document but the process is the same for any previous versions too.

It is assumed the reader is already familiar with the general operation of both the Automation Builder and Panel Builder software applications. If additional help is required please consult the relevant user manuals, online help systems or contact your local ABB automation support centre.

### Exporting tags from Automation Builder

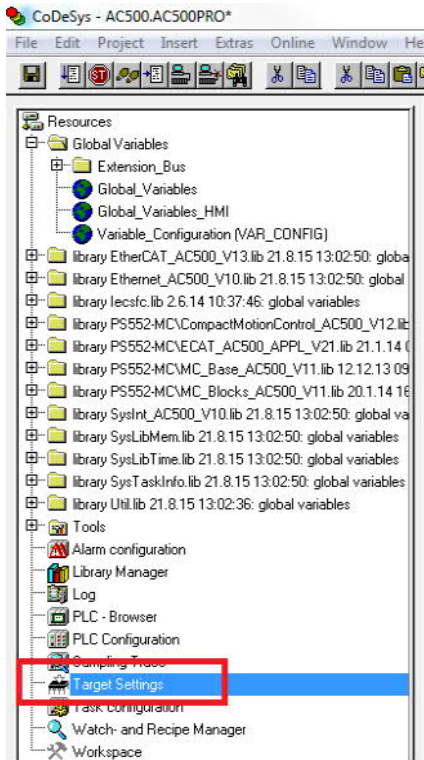
It is possible to export as many or as few of the PLC project tags as are needed. A good “tip” when developing your AC500 PLC projects is to declare all variables (tags) that are to be used in combination with a CP600 HMI as Global Variables and declare these in a dedicated global variable module within the ‘Resources’ tab of the CoDeSys programming environment as shown below...



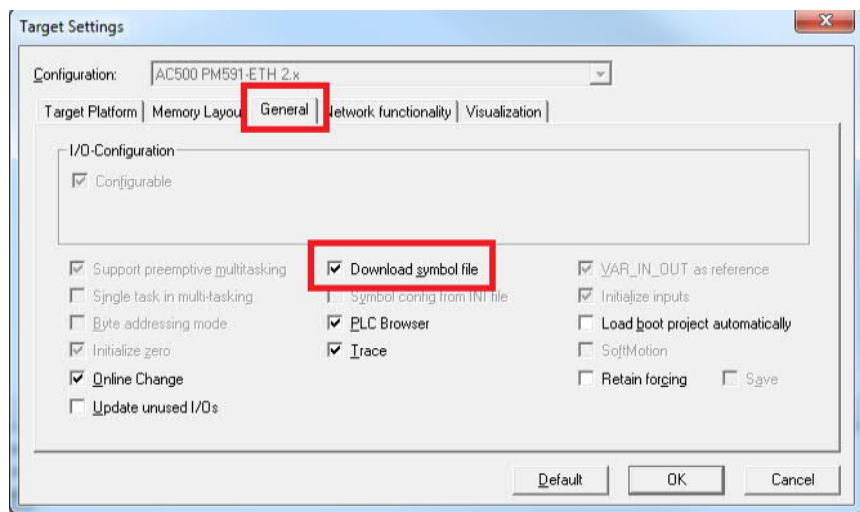
This allows you to easily identify the HMI related data and makes it very simple to select the symbols (tags) that need to be exported for use in the Panel Builder HMI software. This isn't mandatory though and if you like you can scatter your HMI related variables throughout the project.

For the example we'll use in this application note all of the HMI related variables are in the standard "Global\_Variables" module, along with some other general purpose variables and we will show how to specifically select those we are interested in.

First of all, having launched the CoDeSys application from your Automation Builder project and in an offline state (i.e. Log out from the AC500 PLC if you are already logged in), select 'Target Settings' from the 'Resources' tab in CoDeSys...

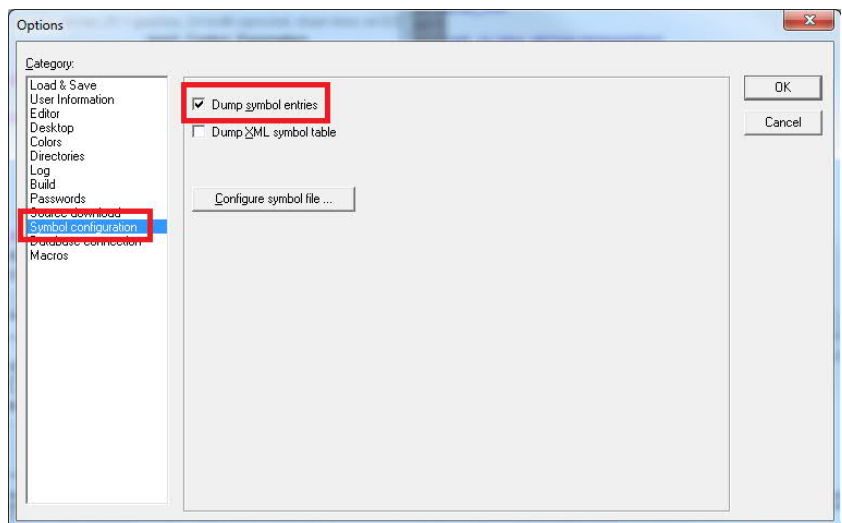
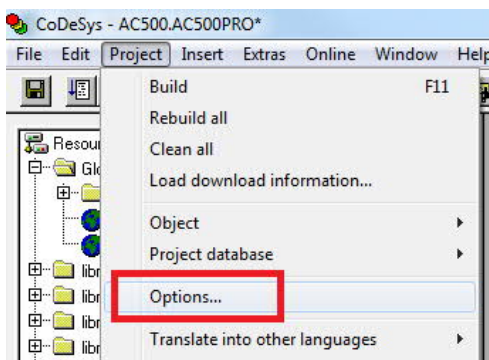


Select the 'General' tab from the resulting dialog and then ensure the 'Download symbol file' option is selected as shown below...



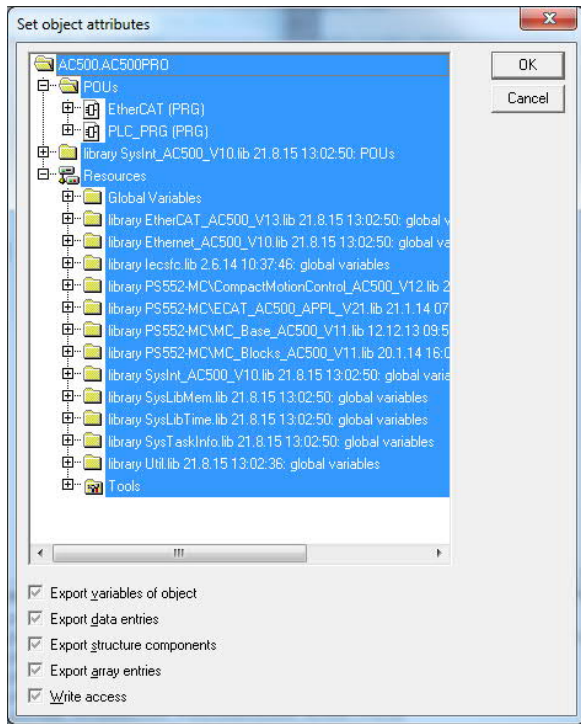
Click OK to accept this setting.

We now need to configure which symbols (tags) from the project we need to export, so from the 'Project' menu at the top of the CoDeSys window select the 'Options...' entry...



From the Options dialog select the 'Symbol configuration' category and ensure 'Dump symbol entries' is selected as shown above.

Now click the ‘Configure symbol file...’ button on the Options dialog. CoDeSys will present a dialog showing the structure of the complete PLC project with everything initially selected as shown below..



If we accept this configuration then, when we eventually export the project symbols (tags), we will end up with a file that contains every single variable in our PLC project. Whilst this is good enough to use (i.e. if the symbol file includes every single variable we won't be missing any!) it does make the resulting file unnecessarily large and also difficult to navigate later when trying to find the specific tags we need to use in our CP600 HMI project.

It is recommended therefore to only select the parts of the project that contain HMI related data (hence why we provided the “tip” earlier that declaring HMI related variables in their own global variables module was a good idea).

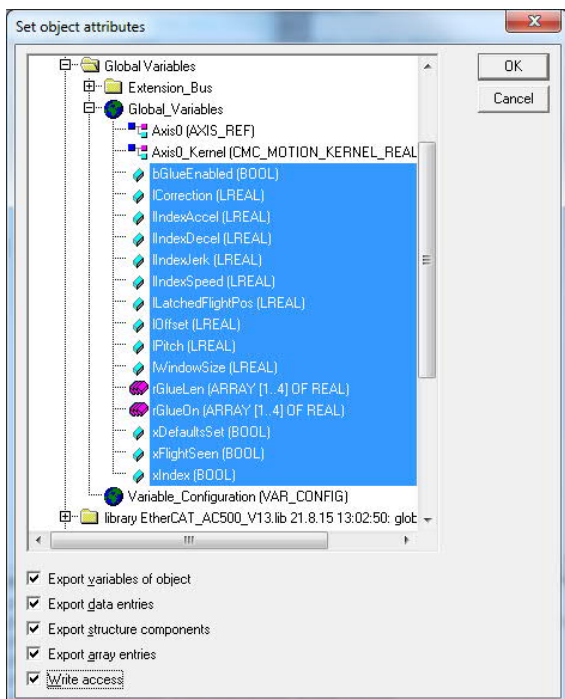
HOWEVER (this part is very important), before attempting to select the relevant part(s) of the project we must first deselect everything. Although the checkboxes underneath the project tree look greyed-out it is still possible to deselect them....so do this, starting with the bottom option (Write access) and working upwards.

Once you have deselected all of the checkboxes click on the ‘OK’ button.

Now click the ‘Configure symbol file...’ button again, the dialog will re-appear, the entire project structure will be selected still, but now all of the checkboxes will be clear. At this point you can now expand the relevant section(s) of the tree where your HMI data is located and start selecting variables/symbols/tags that you would like to export.

Our data for this example is located in the Global Variables section so we will expand this and select the variables we're particularly interested in (using the Shift and/or Ctrl buttons on our keyboard to make multi-selections as required)...

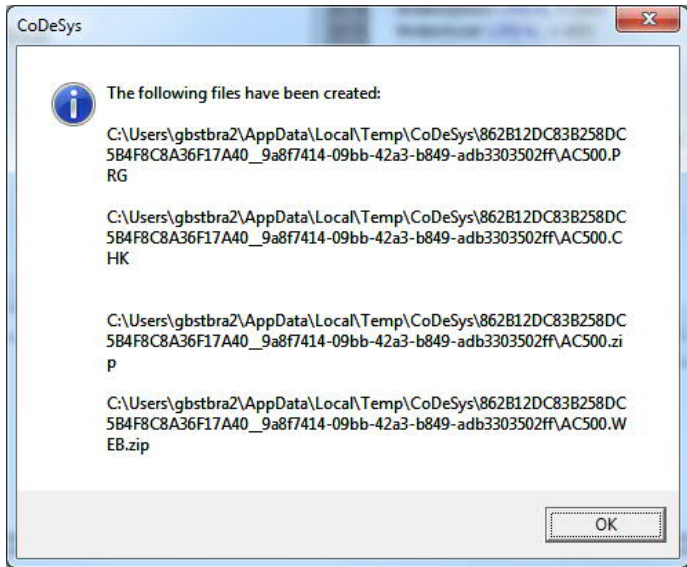
Once we've selected the necessary variables we then re-select all of the options below the tree view (as shown below) and click the ‘OK’ button once more (and then click ‘OK’ again on the Options dialog)...



We are now ready to create the export file, but to do this it is first necessary to rebuild the project and create a boot project (so be sure that your PLC project is in a state that allows it to build without any errors). It is recommended to first select 'Project > Clean all' (say 'Yes' when asked if you wish to continue) followed by 'Project > Rebuild all'.

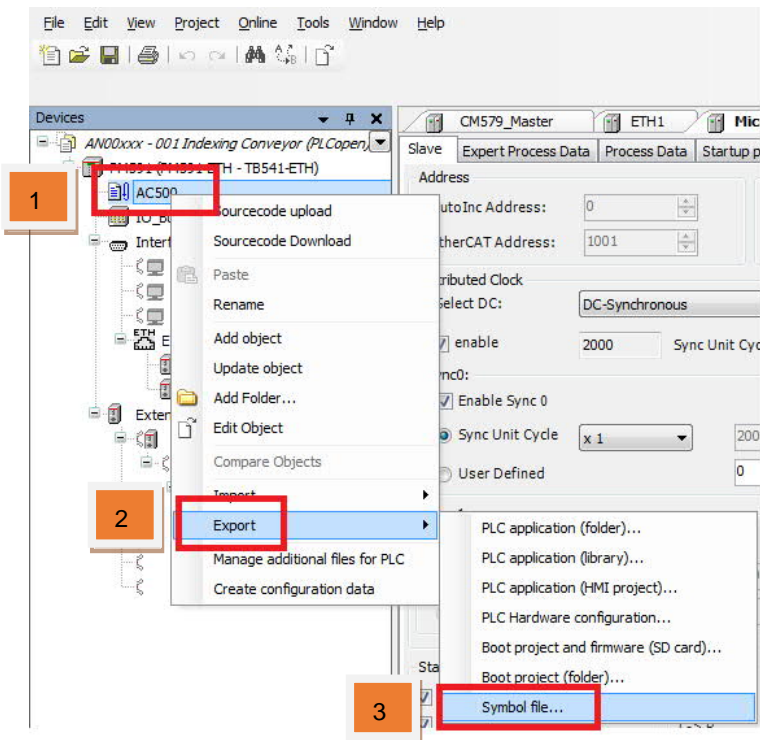
Once the PLC project has compiled successfully select 'Online > Create boot project' (there is no need to Login to a PLC to create the necessary files, just select Create boot project, but of course at some point during the application development you will need to Login first and then, once connected to the PLC, create the actual boot project on the PLC itself).

If this works correctly you should see a dialog something like this...



...showing that AC500.prg, AC500.chk, AC500.zip and AC500.web.zip files have been created.

We now need to export the Symbol file using Automation Builder, so save the changes and close CoDeSys. Once CoDeSys is closed, right click the PLC application icon in the Automation Builder Device Tree and select 'Export > Symbol file'...



Automation Builder will ask you where the Symbol file should be stored (it is a good idea to place it in the same directory as the PLC project) so navigate to the required path and click 'OK'. Automation Builder will show a dialog confirming the AC500.sym file has been copied successfully. At this point you may wish to check that the symbols have indeed been exported as required,

so using a text editor (e.g. Notepad) open the AC500.sym file that has just been saved. If the process was successful you should see the variables you selected listed in the file. Here is an example based on the variables we selected in our earlier example screenshots...

```

AC500.SYM - Notepad
File Edit Format View Help
|:Version=2
|:ProjectId=261369
|:Checksum=0
|:FileSize=0
|.bGlueEnabled:BOOL:3:88:1:b:16#02040048
|.lCorrection:LREAL:5:11232:8:b:16#02000040
|.lIndexAccel:LREAL:3:16:8:b:16#02040048
|.lIndexDecel:LREAL:3:24:8:b:16#02040048
|.lIndexJerk:LREAL:3:32:8:b:16#02040048
|.lIndexSpeed:LREAL:3:8:8:b:16#02040048
|.lLatchedFlightPos:LREAL:5:11224:8:b:16#02000040
|.lOffset:LREAL:3:40:8:b:16#02040048
|.lPitch:LREAL:3:0:8:b:16#02040048
|.lWindowSize:LREAL:3:48:8:b:16#02040048
|.rGlueLen:ARRAY [1..4] OF REAL:3:72:16:b:16#02040048
|.rGlueLen[1]:REAL:3:72:4:b:16#02040048
|.rGlueLen[2]:REAL:3:76:4:b:16#02040048
|.rGlueLen[3]:REAL:3:80:4:b:16#02040048
|.rGlueLen[4]:REAL:3:84:4:b:16#02040048
|.rGlueOn:ARRAY [1..4] OF REAL:3:56:16:b:16#02040048
|.rGlueOn[1]:REAL:3:56:4:b:16#02040048
|.rGlueOn[2]:REAL:3:60:4:b:16#02040048
|.rGlueOn[3]:REAL:3:64:4:b:16#02040048
|.rGlueOn[4]:REAL:3:68:4:b:16#02040048
|.xDefaultsSet:BOOL:3:89:1:b:16#02040048
|.xFlightSeen:BOOL:5:3906:1:b:16#02000040
|.xIndex:BOOL:5:3905:1:b:16#02000040

```

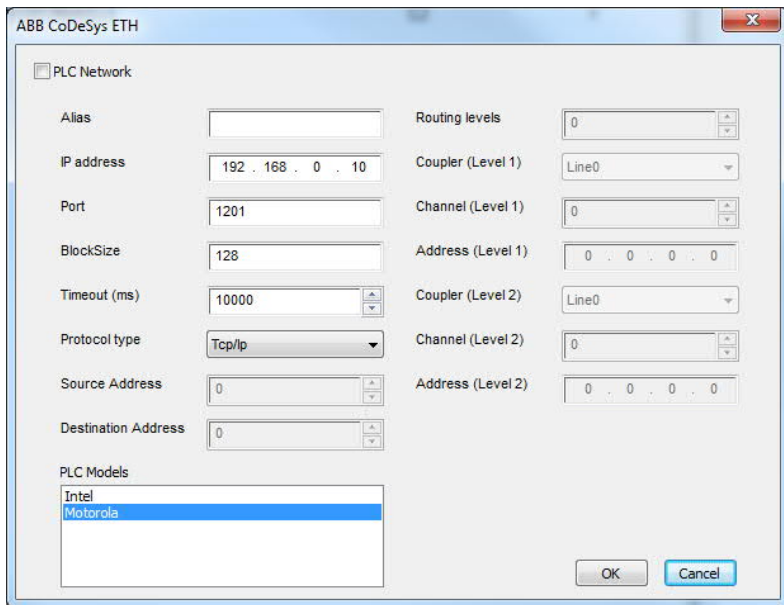
If your file is empty or it contains all of the variables from the complete PLC project then you have not followed one of the steps correctly (e.g. be sure to deselect all of the object attribute options and accept this change before finally selecting the required variables).

If the AC500.sym file looks correct we are now ready to import this data into our Panel Builder CP600 project.

### Importing tags into Panel Builder

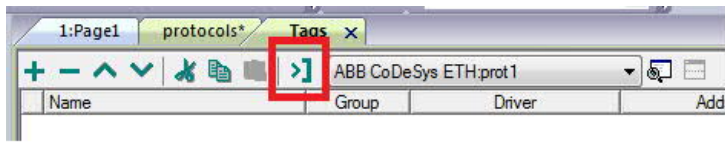
Open your Panel Builder CP600 HMI project (or start a new project) and if necessary configure an ABB CoDeSys ETH protocol (i.e. the protocol that will be used for communication between the HMI and the AC500 PLC's Ethernet port) or an ABB CoDeSys serial protocol (if the HMI is to be connected to the PLC via a serial connection).

Just for reference, here is an example of what the configuration for the ABB CoDeSys ETH protocol would look like if the PLC was using the default IP address of 192.168.0.10...

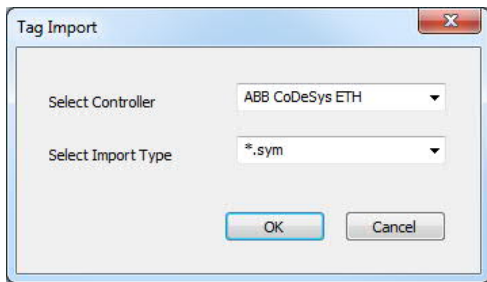


Once the necessary CoDeSys protocol has been added to the project double-click on the 'Tags' entry in the Panel Builder (PB) project view. The right hand pane will show all the tags that are currently defined for the HMI project (this will be an empty list for a new project of course).

Making sure the CoDeSys protocol has been selected in the list of available protocols, click on the 'Import Tags' button just to the left of the protocol selection...



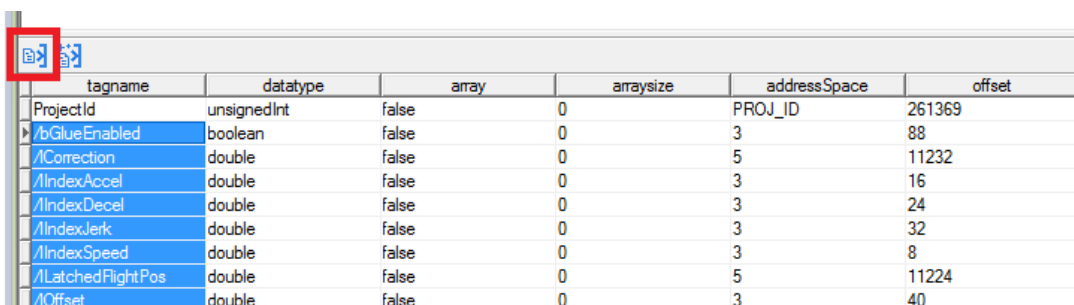
Panel Builder will now ask you to confirm the type of file to be imported – select \*.sym (as this is what we exported from AB) and click OK...



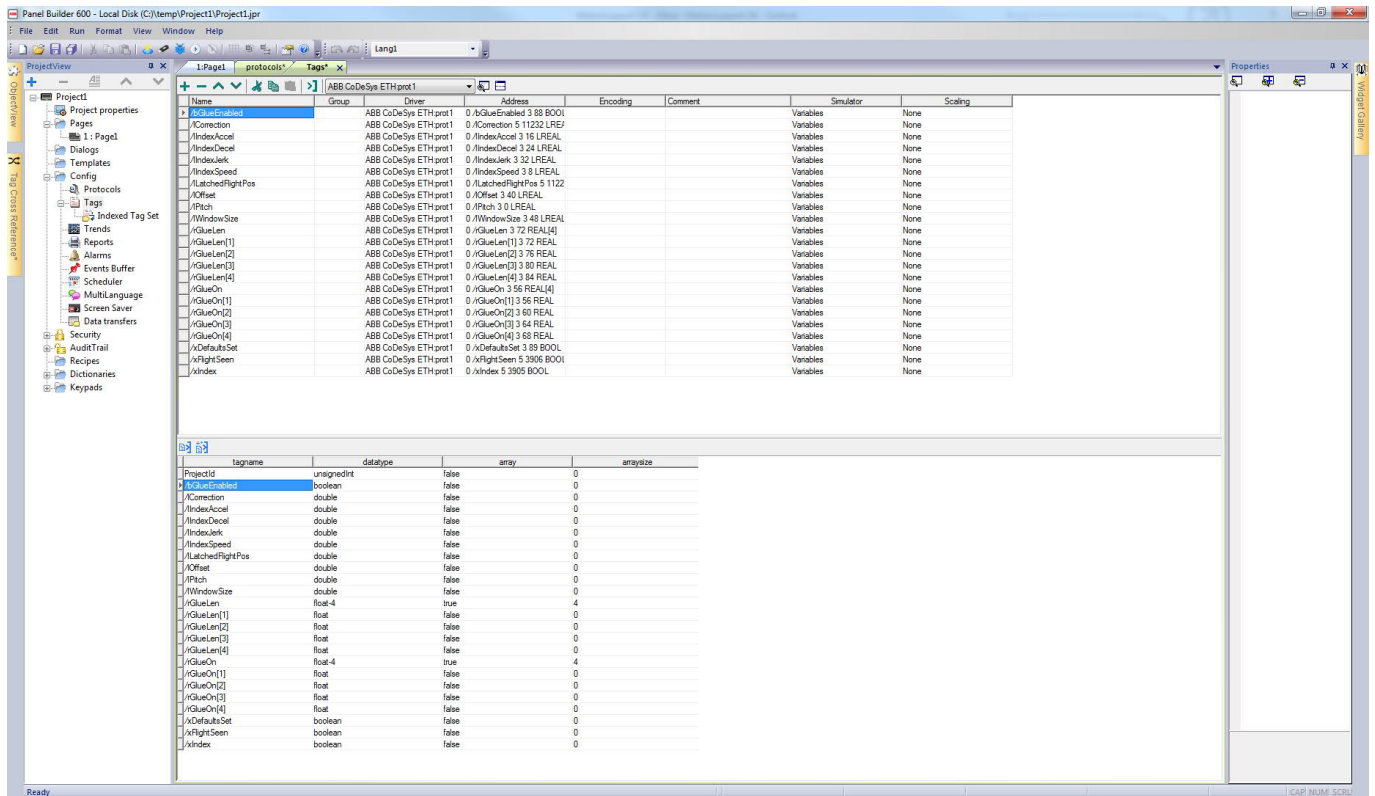
Navigate to the directory where the .sym file was saved (e.g. the PLC project's directory) and open this file. If this has worked the bottom half of the tag pane will now show a list of all of the tags available from the .sym file...

tagname	datatype	array	arraysize	addressSpace	offset
ProjectId	unsignedInt	false	0	PROJ_ID	261369
/bGlueEnabled	boolean	false	0	3	88
/lCorrection	double	false	0	5	11232
/lIndexAccel	double	false	0	3	16
/lIndexDecel	double	false	0	3	24
/lIndexJerk	double	false	0	3	32
/lIndexSpeed	double	false	0	3	8
/lLatchedFlightPos	double	false	0	5	11224
/lOffset	double	false	0	3	40
/lPitch	double	false	0	3	0
/lWindowSize	double	false	0	3	48
/rGlueLen	float-4	true	4	3	72
/rGlueLen[1]	float	false	0	3	72
/rGlueLen[2]	float	false	0	3	76
/rGlueLen[3]	float	false	0	3	80
/rGlueLen[4]	float	false	0	3	84
/rGlueOn	float-4	true	4	3	56
/rGlueOn[1]	float	false	0	3	56
/rGlueOn[2]	float	false	0	3	60
/rGlueOn[3]	float	false	0	3	64
/rGlueOn[4]	float	false	0	3	68
/xDefaultsSet	boolean	false	0	3	89
/xFlightSeen	boolean	false	0	5	3906
/xIndex	boolean	false	0	5	3905

Select one or more tags to be included in the HMI project (or all of them – with the possible exception of the ProjectID tag which is very rarely used) and then click the 'Import Tag(s)' button (the left hand button above the list of imported tags)...

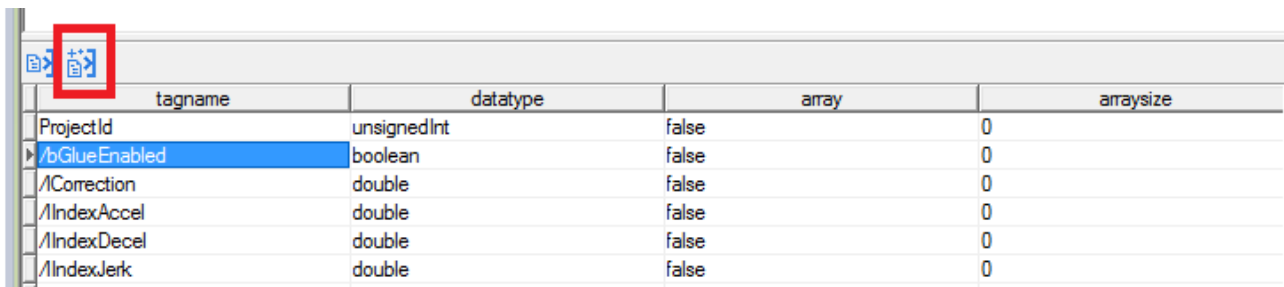


Panel Builder will now copy all of the selected tags to the top half of the window, making them available for selection in the HMI application....



At this point it is probably worth saving your HMI project.

Should you ever re-import a new version of the .sym file (e.g. maybe you decide to change one of your variables from a Byte to a Word) then after importing the .sym file as before you can simply click the 'Update Tags' button (the right hand button above the list of imported tags) and Panel Builder will update the definitions of any tags with matching names.



Contact us

For more information please contact your local ABB representative or one of the following:  
new.abb.com/motion  
new.abb.com/drives  
new.abb.com/drivespartners  
new.abb.com/PLC

© Copyright 2015 ABB. All rights reserved.  
Specifications subject to change without notice.