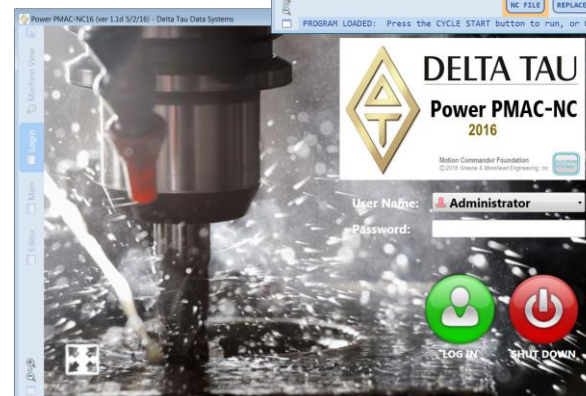
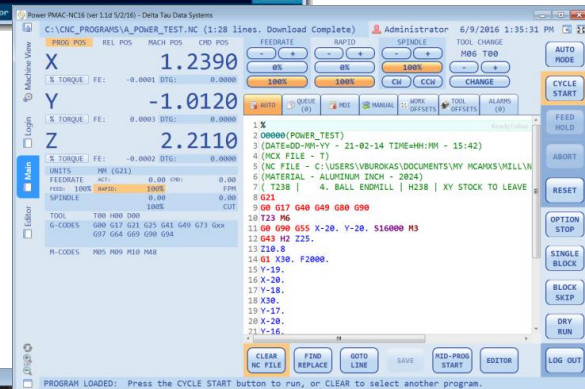
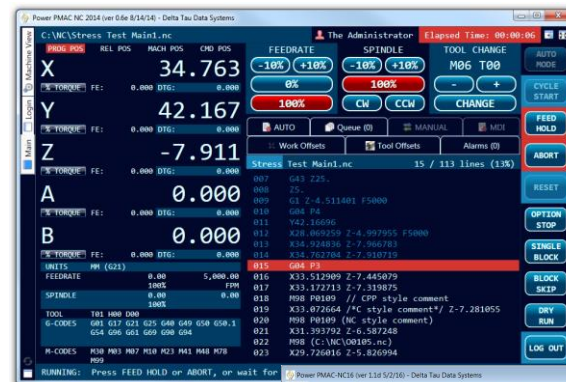


# ODT – Power Pmac NC 16 Software Training Presentation

March 2018  
Vincent Burokas





**Presented by:**

**Vincent Burokas  
CNC Product Manager  
vburokas@deltatau.com**



The PPNC16 program is compatible with Windows 7 or newer (64-bit or 32-bit). The application requires .NET 4.6.1 and the Visual C++ 2010 runtime libraries. The application will install these components automatically if not present. The following links can be used for manual installation of the same libraries.

***Microsoft .NET Framework 4 (Web Installer)***

<http://www.microsoft.com/en-us/download/details.aspx?id=17851>

***Microsoft Visual C++ 2010 Redistributable Package (x64)***

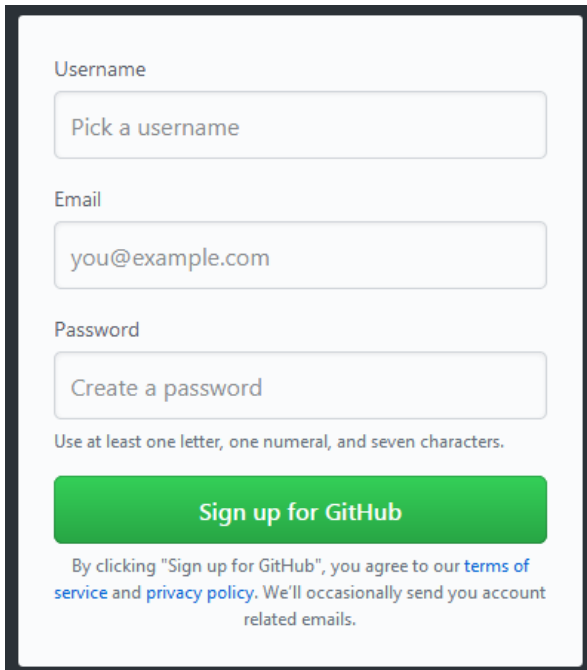
<http://www.microsoft.com/en-us/download/details.aspx?id=14632>

***Microsoft Visual C++ 2010 Redistributable Package (x86)***

<http://www.microsoft.com/en-us/download/details.aspx?id=5555>



- Power PMAC-NC is distributed via a private GitHub repository (Cloud Based)
- Sign-Up for a [free GitHub account](#) and send your username (password not required) to your **Omron Delta Tau** support representative.
- You will be invited and given read-only access to the repository.
- Install [GitHub for Windows](#) on your development PC, log in, and "Clone" the repository. The GitHub application will give you access to the latest versions.



Username

Email

Password

Use at least one letter, one numeral, and seven characters.

[Sign up for GitHub](#)

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.

<https://github.com>

[Download for Windows \(64bit\)](#)

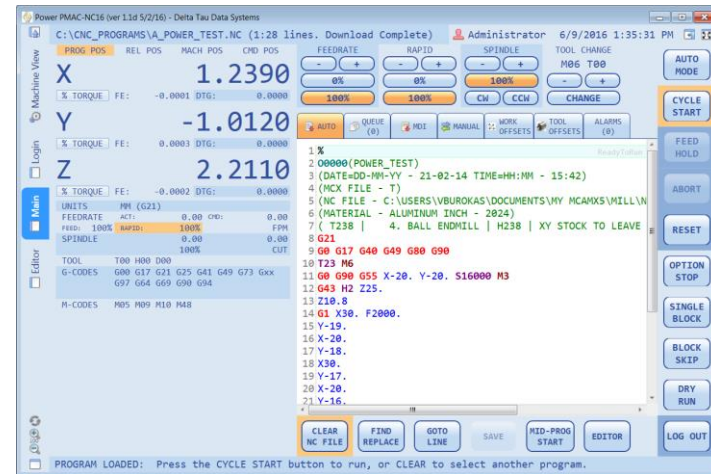
<https://desktop.github.com/>



- The Power PMAC-NC is a fully portable software application.
- It does not require the typical “Installation” other software packages require.
- There is no data maintained in the Windows registry.
- You can copy or move the application folders and the program will work.
- This makes cloning the project easy for OEM machine builders.

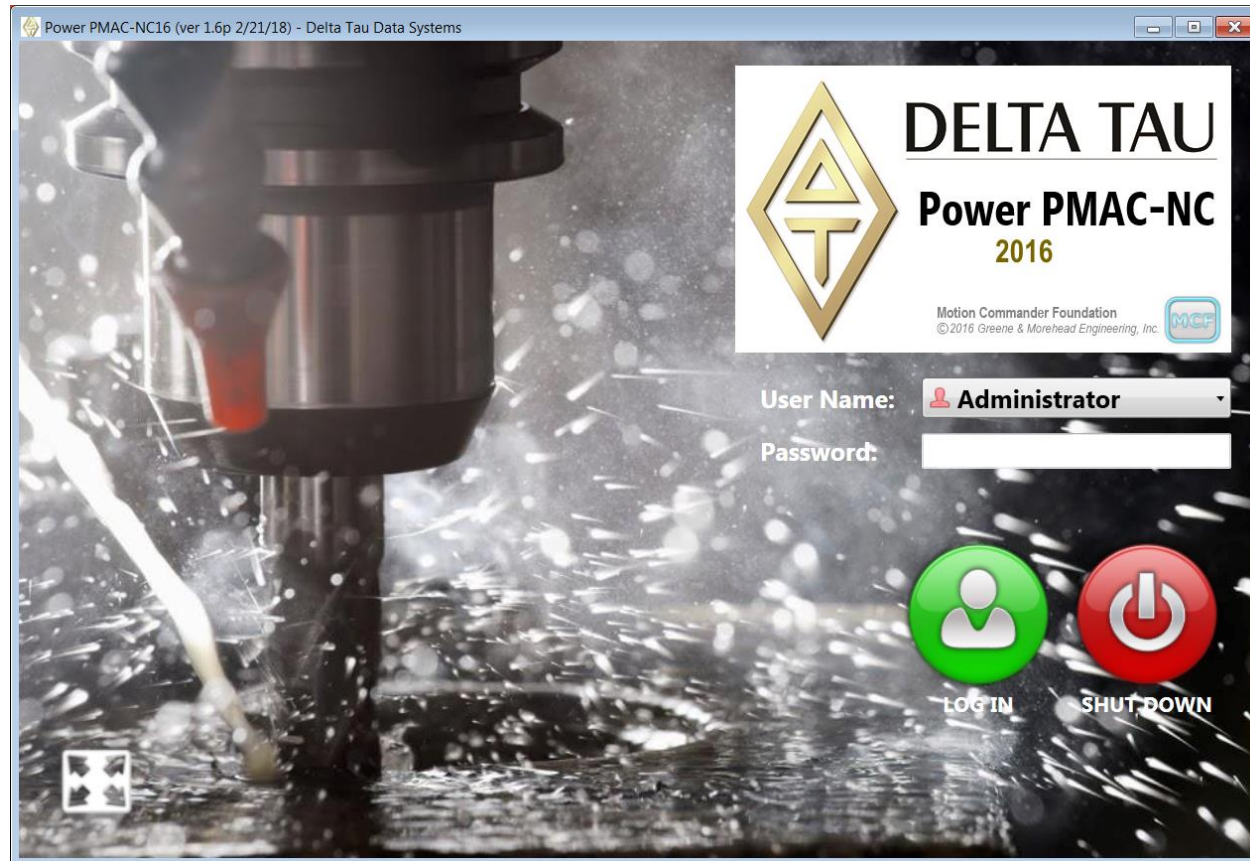


**The Power PMAC-NC Executable code is hardware dongle protected!**

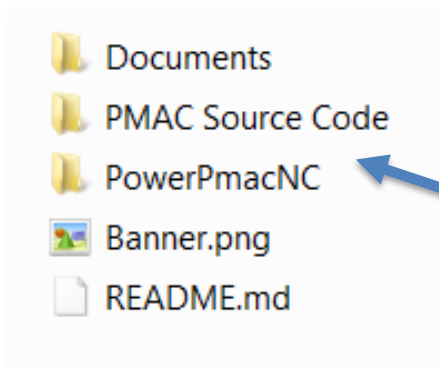
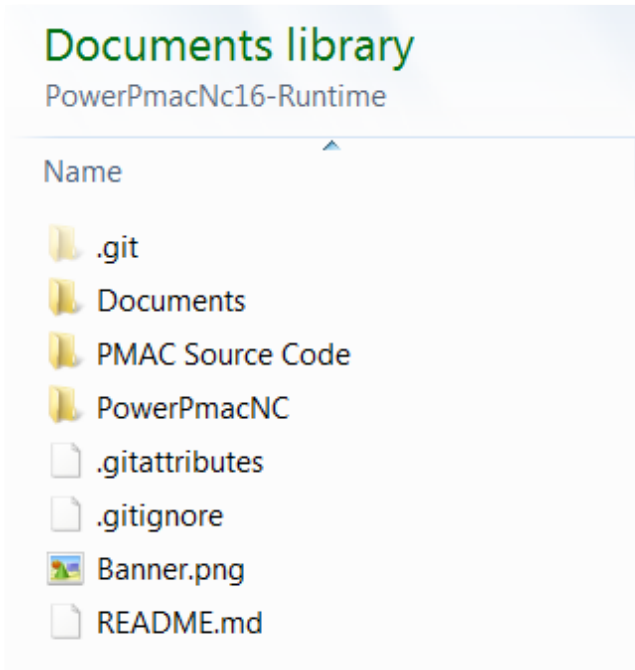




# Power PMAC NC16-Runtime







**Documents** - Contains documents and manuals for the Power PMAC-NC16 application.

**PMAC Source Code** - This is where you will find the Power PMAC-NC16 specific project for the PMAC. The HMI requires these files.

**PowerPmacNC** - The actual HMI application executable will be contained in this folder.

**.git** - Various files and folders which are used by the GitHub application. These files are required in the clone folder, but can be deleted in the working folder.

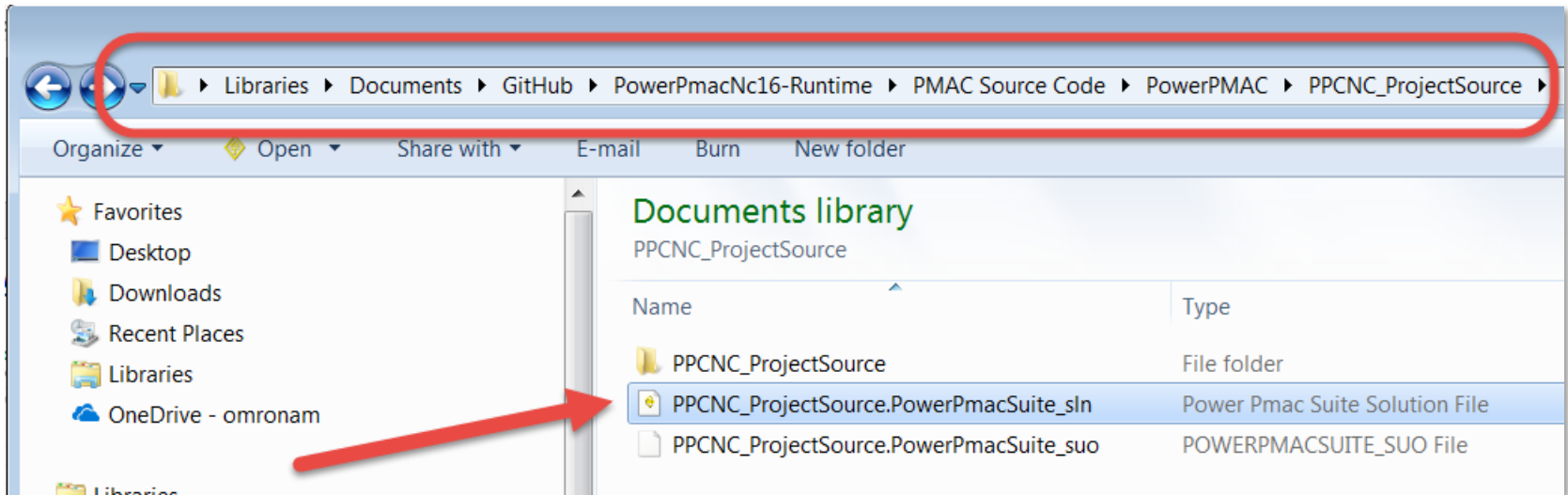
The **.git** files can be deleted in a working copy or destination machine.



## PMAC Source Code

(The PMAC script files...)

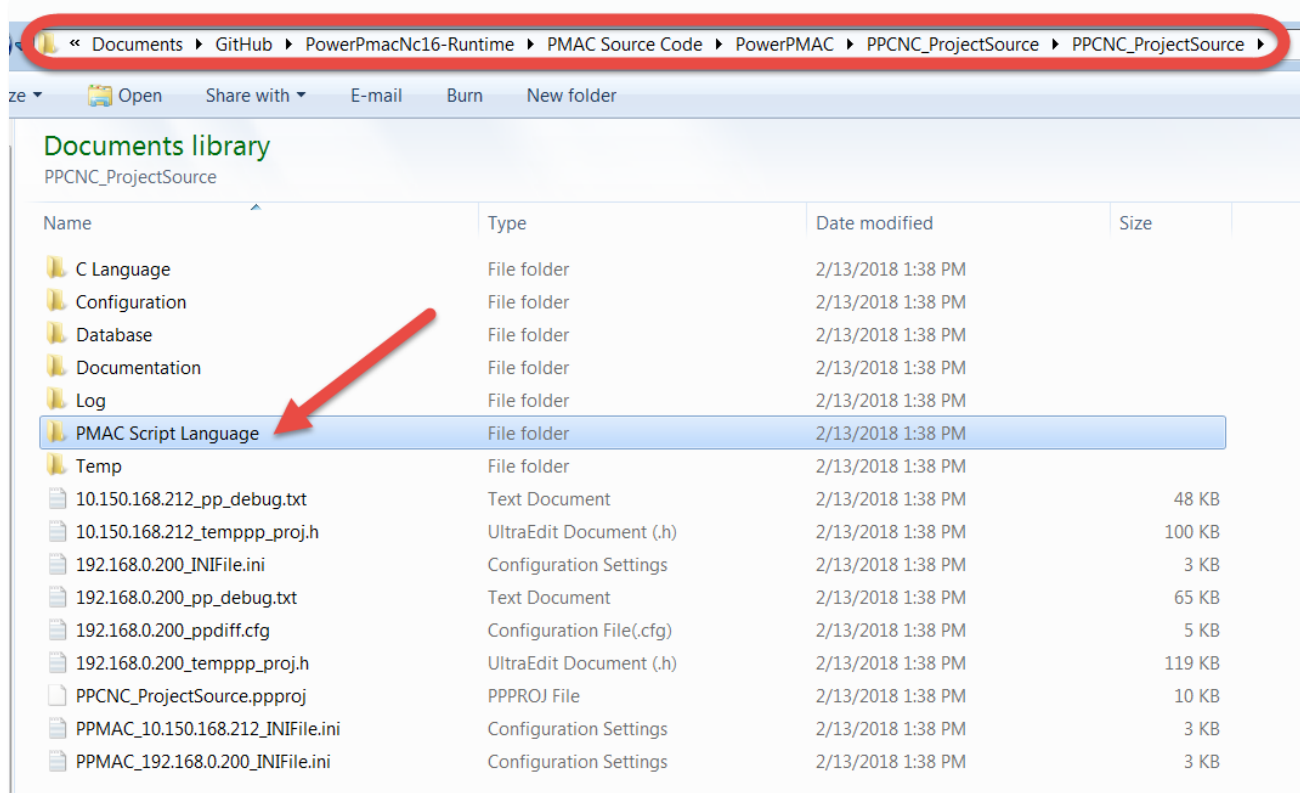




- The default Power PMAC IDE project will be found in the above location.
- This is the project solution you will open in the IDE.







# PMAC Script Language



- All PMAC script files necessary for the Power PMAC-NC16 HMI.
- All files necessary for application customization and modification (...in general).



-  Global Includes
-  Libraries
-  Motion Programs
-  PLC Programs


**Global Includes** – Contains header files in the .pmh format

**Libraries** – Contains subprograms in the .pmc format.

**Motion Programs** – Contains the bootloader programs in the .pmc format.

**PLC Programs** – Contains the PLC's which do all the status monitoring, command/status handshaking, and logic control for the application.





- ppnc\_codestatus.pmh
- ppnc\_csparameters.pmh
- ppnc\_dtpendant.pmh
- ppnc\_laserparameters.pmh
- ppnc\_messages.pmh
- ppnc\_ncinterfacedefinitions.pmh
- ppnc\_spindleparameters.pmh
- ppnc\_virtualmotors.pmh

**ppnc\_codestatus** – G-Code alias names to enhance program readability.

**ppnc\_csparameters** - Variables and parameters related to the coordinate system.

**ppnc\_dtpendant** - Variables and parameters required by the pendant when used (optional).

**ppnc\_laserparameters** – Variables and parameters used when the custom laser example is enabled.







**ppnc\_messages** - Custom messages defined by the integrator.

**ppnc\_ncinterfacedefinitions** - Parameters definitions required by the HMI for setup and handshaking.

**ppnc\_spindleparameters** - Parameters relative to spindle functionality.

**ppnc\_virtualmotors** - Virtual parameters which can be used for initial setup if no actual motors exist.



 `ppnc_dcodes.pmc` `ppnc_gcodes.pmc` `ppnc_mcodes.pmc` `ppnc_tcodes.pmc` `ppnc_timer.pmc` `ppnc_xforms.pmc`

**ppnc\_dcodes** - D-Code script which reads cutter radius and wear for each tool.

**ppnc\_gcodes** - G-Code script which contains the actual G-code instructions followed by PMAC.


**ppnc\_mcodes** - M-Code script which contains the actual M-code instructions followed by PMAC.

**ppnc\_tcodes** - T-Code script which reads tool length and wear for each tool.

**ppnc\_timer** – PMAC subprogram timer used by other script files.

**ppnc\_xforms** – PMAC transformation matrix which handles offsets, rotation, mirroring, scaling, etc.





















ppnc\_auto.pmc  
ppnc\_mdi.pmc  
ppnc\_readme.pmc

**ppnc\_auto** – Contains the bootloader program for AUTO mode programs. This launches the main program when Cycle Start is initiated.

**ppnc\_mdi** – Contains the bootloader program for MDI mode programs. This launches the MDI program when Cycle Start is initiated.

**ppnc\_readme** – Place holder. No function.



-  ppnc\_function.plc
-  ppnc\_handwheel.plc
-  ppnc\_hmimonitor.plc
-  ppnc\_home.plc
-  ppnc\_initialize.plc
-  ppnc\_laserpositionreport.plc
-  ppnc\_lasersoftpanel.plc
-  ppnc\_messages.plc
-  ppnc\_mip\_pendant.plc
-  ppnc\_override.plc
-  ppnc\_positionreport.plc
-  ppnc\_reset.plc
-  ppnc\_restart.plc
-  ppnc\_softpanel.plc
-  ppnc\_spindle.plc
-  ppnc\_worktooloffset.plc

**ppnc\_function** - Monitors the Command/Status states and executes logic accordingly.

**ppnc\_handwheel** – Executes handwheel jogging motion when a hardware MPG is present (optional).

**ppnc\_hmimonitor** - Monitors and performs specific tasks when the HMI starts or shuts down.

**ppnc\_home** - Performs the homing functionality.

















**ppnc\_initialize** - Initializes parameters required by the PMAC on boot.

**ppnc\_laserpositionreport** - Monitors and reports position and other real time data for display in the HMI when the laser project is enabled (optional).

**ppnc\_lasersoftpanel** - Monitors and performs different software panel tasks when the laser project is enabled (optional).



# PMAC Script – PLC Programs (cont.)

-  ppnc\_function.plc
-  ppnc\_handwheel.plc
-  ppnc\_hmimonitor.plc
-  ppnc\_home.plc
-  ppnc\_initialize.plc
-  ppnc\_laserpositionreport.plc
-  ppnc\_lasersoftpanel.plc
-  ppnc\_messages.plc
-  ppnc\_mip\_pendant.plc
-  ppnc\_override.plc
-  ppnc\_positionreport.plc
-  ppnc\_reset.plc
-  ppnc\_restart.plc
-  ppnc\_softpanel.plc
-  ppnc\_spindle.plc
-  ppnc\_worktooloffset.plc

**ppnc\_messages** – Contains logic to enable and display custom messages. All types are shown for reference.

**ppnc\_mip\_pendant** - Monitors and performs different tasks related to the MIP pendant (optional).

**ppnc\_override** - Monitors and performs override functionality.

**ppnc\_positionreport** - Monitors and reports position and other real time data for display in the HMI.

**ppnc\_reset** - Monitors and performs the reset state and executes reset logic when required.

**ppnc\_restart** -Contains logic to enable the M99 Lxxx looping functionality.

**ppnc\_softpanel** - Monitors and performs different software panel tasks.

**ppnc\_spindle** - Monitors and performs tasks related to the spindle when present.

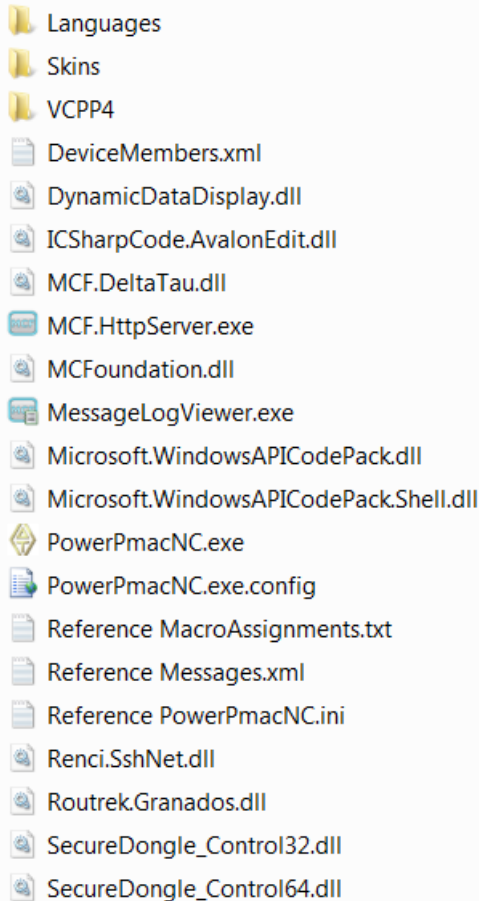
**ppnc\_worktooloffset** – Performs the Work and Tool offset functionality.



## PowerPmacNC

(The HMI files...)





**Main Files** – The main HMI executable files, dll's, xml, and other files.

**Languages** – Adds language files when new users are added to the system. These files can be modified.

**Skins** – Contains various color scheme skin files. These can be modified and or changed to suit the customer.

**VCPP4** – Contains the Visual C++ 2010 runtime libraries. These are used to install these files if the destination PC does not already have them.

Most importantly the executable file, reference files, and .dll files are located in this folder.

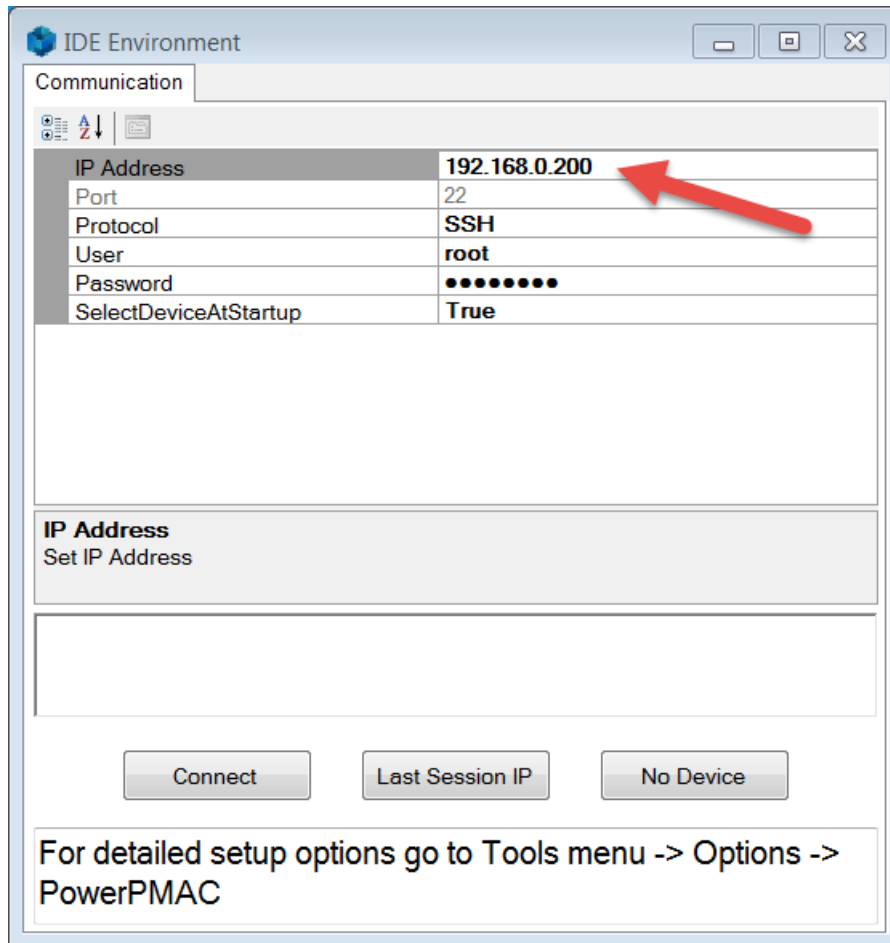


## Setting up the PMAC using the default Power PMAC-NC16 Project files

(Basic Setup using virtual motors...)



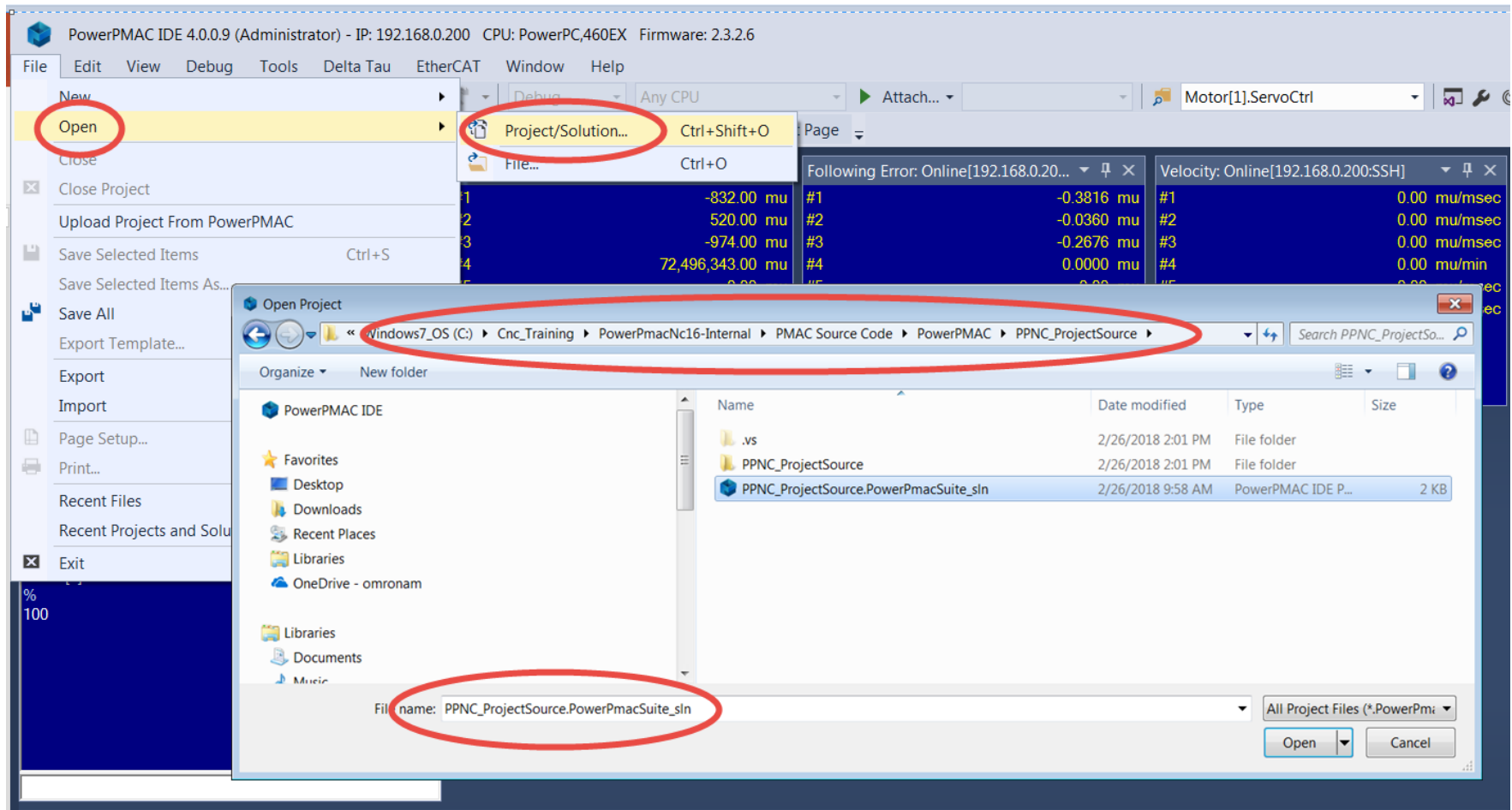
# Starting the Power PMAC IDE



- Start the IDE and establish communication using the default IP address (or other...)

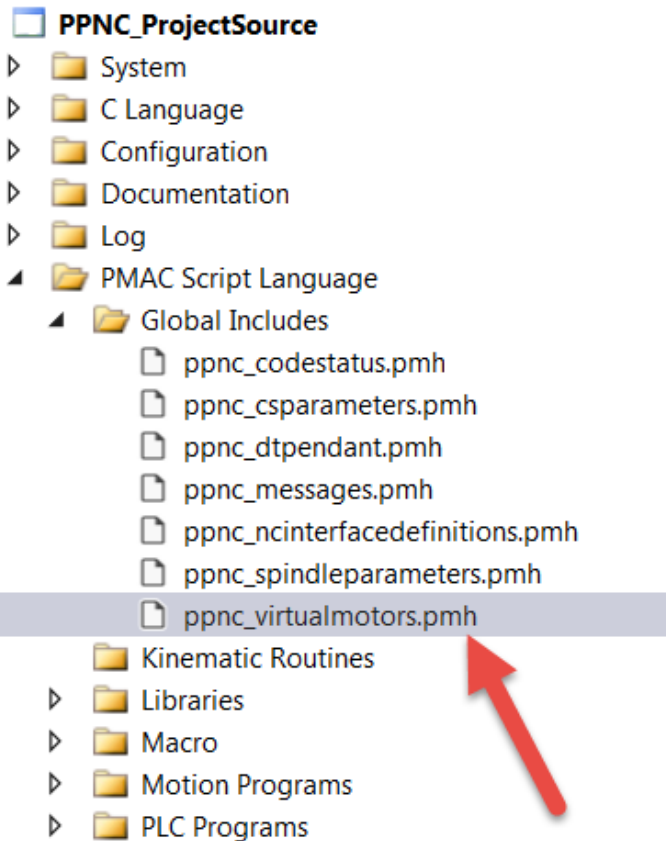


# Loading the Power PMAC-NC Project



- Open the default PPNC project from the IDE.





- In this example we will use a virtual motor setup file.
- We will setup real motors later in the exercise.



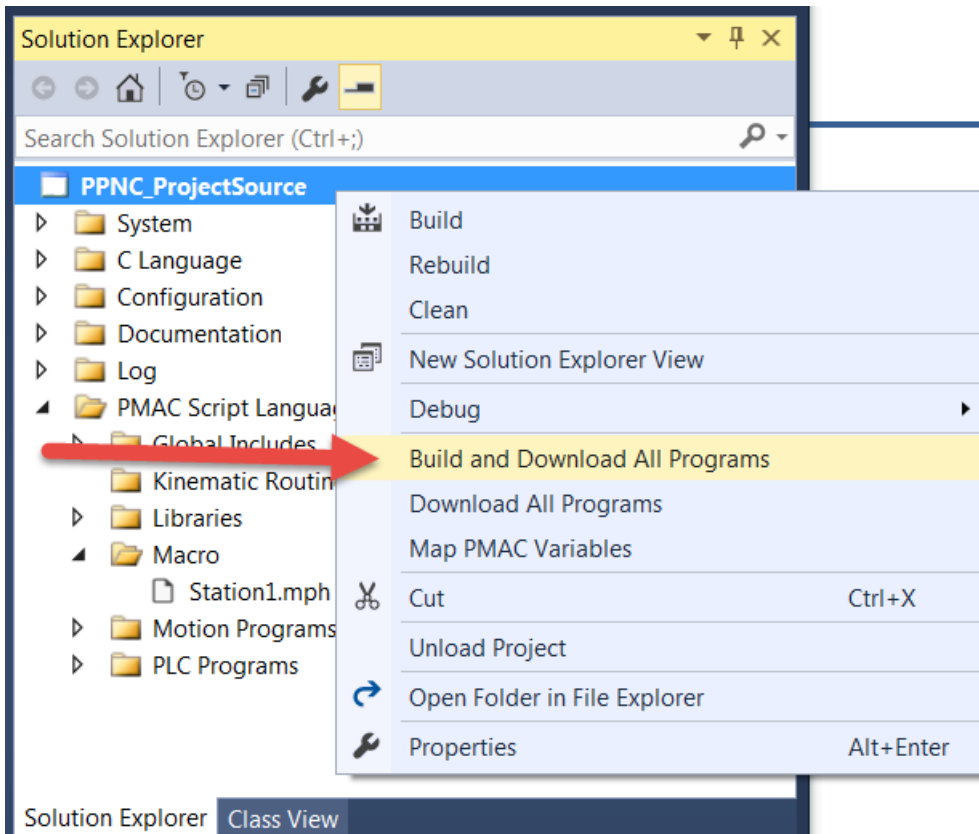
ppnc\_virtualmotors.pmh

```
EncTable[1].type=1
EncTable[1].index1=0
EncTable[1].index2=0
EncTable[1].index3=0
EncTable[1].index4=0
EncTable[1].index5=0
EncTable[1].pEnc1=Sys.udata[11].a
EncTable[1].pEnc=Sys.udata[11].a
EncTable[1].MaxDelta=0
EncTable[1].ScaleFactor=1
EncTable[1].TanHalfPhi=0
EncTable[1].CoverSerror=0
Motor[1].pDac=sys.udata[11].a
Motor[1].Ctrl=sys.PosCtrl
Motor[1].ServoCtrl=1
Motor[1].FatalFeLimit=0
Motor[1].pAmpFault=0
Motor[1].pAmpEnable=0
```

- The ppnc\_virtualmotors.pmh file will setup virtual, non physical motors, so we can test without motors.



# Download to PMAC...



From the IDE Solution Explorer:

Right Click on PPNC\_ProjectSource and Choose “Build and Download All Programs” ...

Look for the “Download Successful” Message...

```
Download successful.
Total Project download time = 8.861 seconds.
Total Project build and download time = 11.4 seconds.
Build Succeeded.
```



A terminal window titled "Terminal: Online [192.168.0.200 : SSH]" with a yellow title bar. The window has a dark blue background with white text. The text shows the execution of a "save" command and its progress: "available\_space = 80404K 1968K", "required\_space = 1968K", "SaveToFlash: cp", "SaveToFlash: sync()", "SaveToFlash: mount", "SaveToFlash: Finish SAVING to Flash.", and "Save Complete". At the bottom, a white input field contains the text "save", with a red arrow pointing to it from the left.

```
Terminal: Online [192.168.0.200 : SSH]
available_space = 80404K 1968K
required_space = 1968K
SaveToFlash: cp
SaveToFlash: sync()
SaveToFlash: mount
SaveToFlash: Finish SAVING to Flash.
Save Complete
save
```

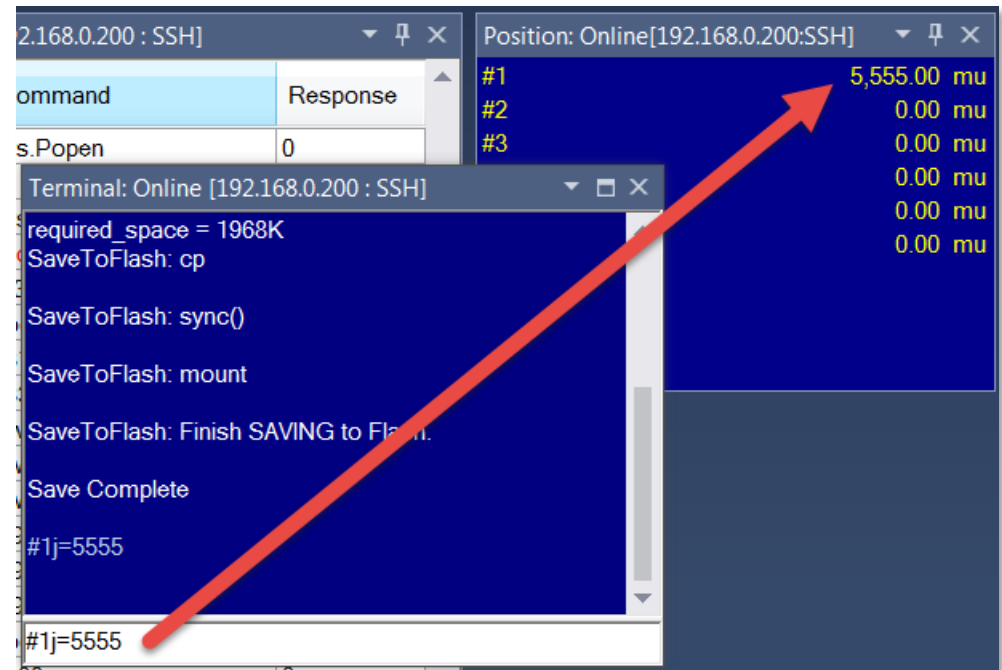
From the IDE Terminal Window:

Issue a “Save” command and look for the “Save Complete” response...



## Jog Test:

From the Terminal Window type #1j=5555 and view position window to insure configuration successful.





## **Setting up the HMI using the default Power PMAC-NC16 Configuration Files**



# The Power PMAC-NC16 Application

PowerPmacNC	
Name	Type
Languages	File folder
Skins	File folder
VCPP4	File folder
DeviceMembers.xml	XML Document
DynamicDataDisplay.dll	Application extension
ICSharpCode.AvalonEdit.dll	Application extension
MCF.DeltaTau.dll	Application extension
MCF.HttpServer.exe	Application
MCFoundation.dll	Application extension
MessageLogViewer.exe	Application
Microsoft.WindowsAPICodePack.dll	Application extension
Microsoft.WindowsAPICodePack.Shell.dll	Application extension
PowerPmacNC.exe	Application
PowerPmacNC.exe.config	XML Configuration File
Reference MacroAssignments.txt	Text Document
Reference Messages.xml	XML Document
Reference PowerPmacNC.ini	Configuration Settings
Renci.SshNet.dll	Application extension
Routrek.Granados.dll	Application extension
SecureDongle_Control32.dll	Application extension
SecureDongle_Control64.dll	Application extension

The Power PMAC-NC 16 HMI can be deployed by simply copying the "PowerPmacNc16-Runtime" folder to any location on your machine. The folder may be renamed if desired.



# Configuring Power PMAC-NC16...

PowerPmacNC	
Name	Type
Languages	File folder
Skins	File folder
VCP4	File folder
DeviceMembers.xml	XML Document
DynamicDataDisplay.dll	Application extension
ICSharpCode.AvalonEdit.dll	Application extension
MCF.DeltaTau.dll	Application extension
MCF.HttpServer.exe	Application
MCFoundation.dll	Application extension
MessageLogViewer.exe	Application
Microsoft.WindowsAPICodePack.dll	Application extension
Microsoft.WindowsAPICodePack.Shell.dll	Application extension
PowerPmacNC.exe	Application
PowerPmacNC.exe.config	XML Configuration File
Reference MacroAssignments.txt	Text Document
Reference Messages.xml	XML Document
Reference PowerPmacNC.ini	Configuration Settings
Renci.SshNet.dll	Application extension
Routrek.Granados.dll	Application extension
SecureDongle_Control32.dll	Application extension
SecureDongle_Control64.dll	Application extension

The PPNC HMI includes three reference files which should be renamed as shown below.

MacroAssignments.txt	Text Document
MCF.DeltaTau.dll	Application extension
MCF.HttpServer.exe	Application
MCFoundation.dll	Application extension
MessageLogViewer.exe	Application
Messages.xml	XML Document
Microsoft.WindowsAPICodePack.dll	Application extension
Microsoft.WindowsAPICodePack.Shell.dll	Application extension
PowerPmacNC.exe	Application
PowerPmacNC.exe.config	XML Configuration File
PowerPmacNC.ini	Configuration Settings
Renci.SshNet.dll	Application extension



## **PowerPmacNC.ini**

- Parameters in this file determine the functionality and look of the HMI.
- Edit this file to customize the Power NC application.

## **Messages.xml**

- Contains the custom messages for Message-Type 1. These are PLC controlled.
- All other messages configured via send 1 “xxxxxx xxxxxx” commands.

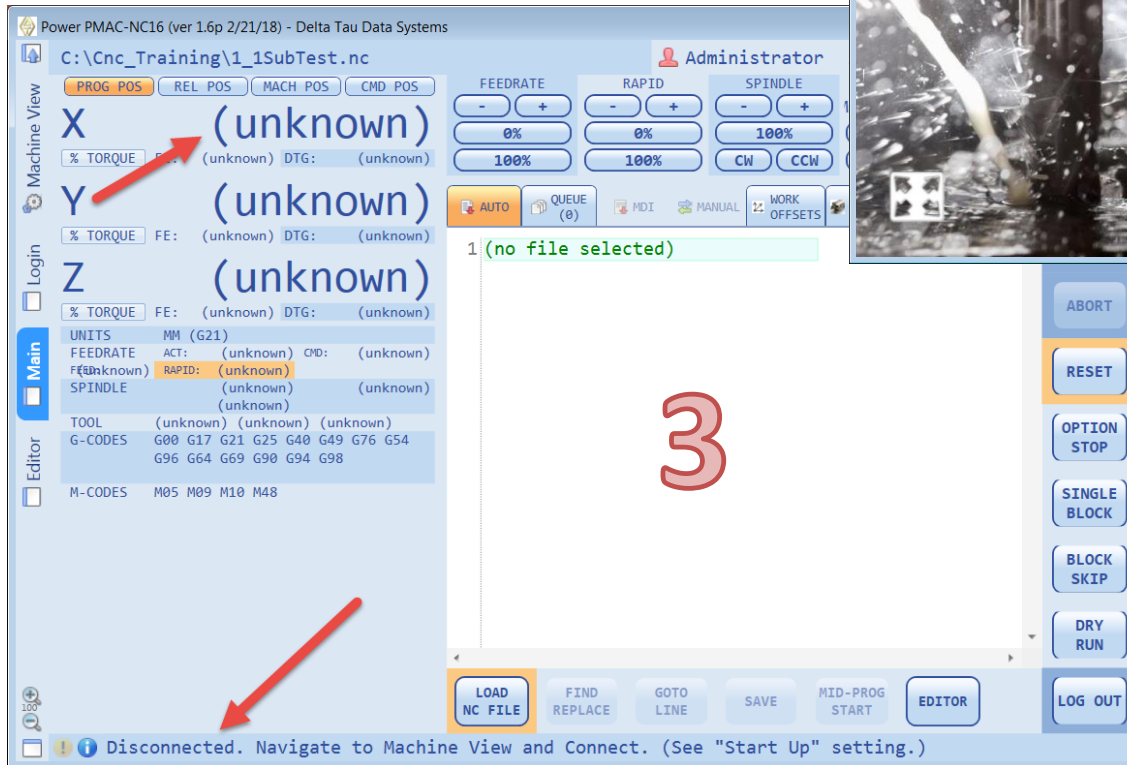
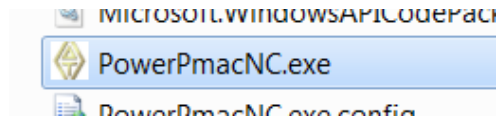
## **MacroAssignments.txt**

- Contains the alias equivalent to setup the Fanuc style #-variables.
- Do not worry about this at now. This is only necessary if you will be using parametric style programming.



# Start the Power PMAC-NC...

1



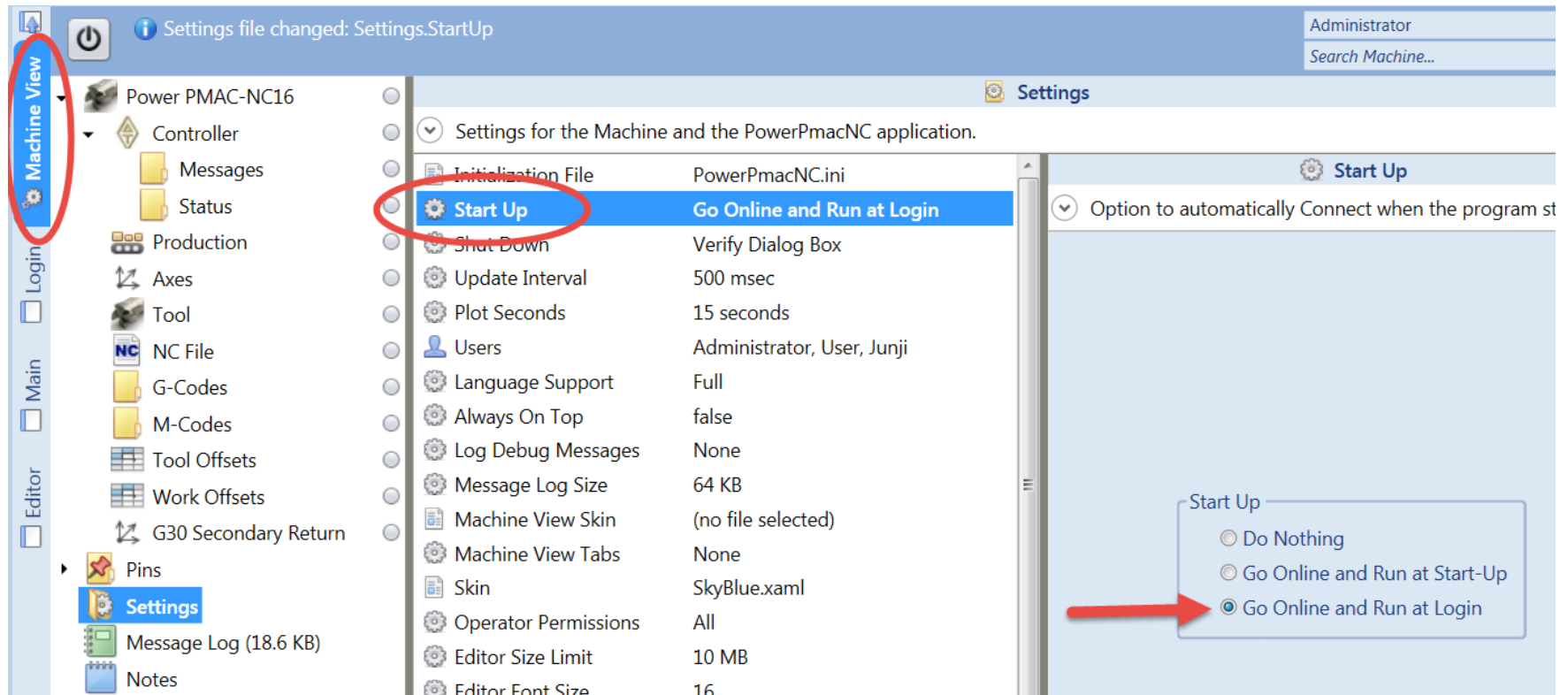
2



3



# Setting up the HMI for the First Time...



Set initial startup action.



# Setting up the HMI for the First Time...

The screenshot displays the OMRON HMI Settings application. On the left, a sidebar contains a tree view with 'Machine View' circled in red. Below it, the 'Settings' icon is also circled in red. The main window is titled 'Settings' and shows a list of settings for the machine and application. A red box highlights the 'Jog Speed1' setting, which is set to '1.000 mm/min'. To the right, a detailed view of the 'Jog Speed1' setting is shown, featuring a dropdown menu set to 'x1 jogging speed', a text input field with '1.000', and an 'Apply' button. The text 'persistent 64-bit float "P700"' is visible below the input field.

Setting	Value
Update Interval	500 msec
Plot Seconds	15 seconds
Users	Administrator, User, Junji
Language Support	Full
Always On Top	false
Log Debug Messages	None
Message Log Size	64 KB
Machine View Skin	(no file selected)
Machine View Tabs	None
Skin	SkyBlue.xaml
Operator Permissions	All
Editor Size Limit	10 MB
Editor Font Size	16
<b>Jog Speed1</b>	<b>1.000 mm/min</b>
Jog Speed2	10.000 mm/min
Jog Speed3	100.000 mm/min
Jog Speed4	1000.000 mm/min
Jog Speed5	10000.000 mm/min
Max Feedrate	5000.000 mm/min

Set jog speeds.



# Homing the Motors Before Running...

Power PMAC-NC16 (ver 1.6p 2/21/18) - Delta Tau Data Systems

C:\Cnc\_Training\CncFiles\WAVE\_SURFACE\_FINISH.NC Administrator 2/28/2018 10:18:07 AM

Machine View Login Editor

PROG POS REL POS **MACH POS** CMD POS

X 0.000  
% TORQUE FE: 0.000 JOG SPEED: x1

Y 0.000  
% TORQUE FE: 0.000

Z 0.000  
% TORQUE FE: 0.000

UNITS MM (G21)  
FEEDRATE ACT: 0.00 CMD: 0.00  
FEED: 100% RAPID: 100% FPM  
SPINDLE 0.00 0.00  
100% CUT

TOOL T00 H00 D00  
G-CODES G00 G17 G21 G25 G40 G49 G80 G97  
G64 G69 G90 G94 G98  
M-CODES M05 M09 M10 M48

FEEDRATE RAPID SPINDLE TOOL CHANGE  
- + - + - + M06 T00  
0% 0% 100% - +  
100% 100% CW CCW CHANGE

AUTO QUEUE (0) MDI **MANUAL** WORK OFFSETS TOOL OFFSETS ALARMS (0)

SELECT JOG SPEED:  
x1 x2 x3 x4 x5

SELECT AXIS TO JOG:  
X Y Z

HOLD BUTTONS TO JOG CONTINUOUS:  
-JOG +JOG STOP **HOME**

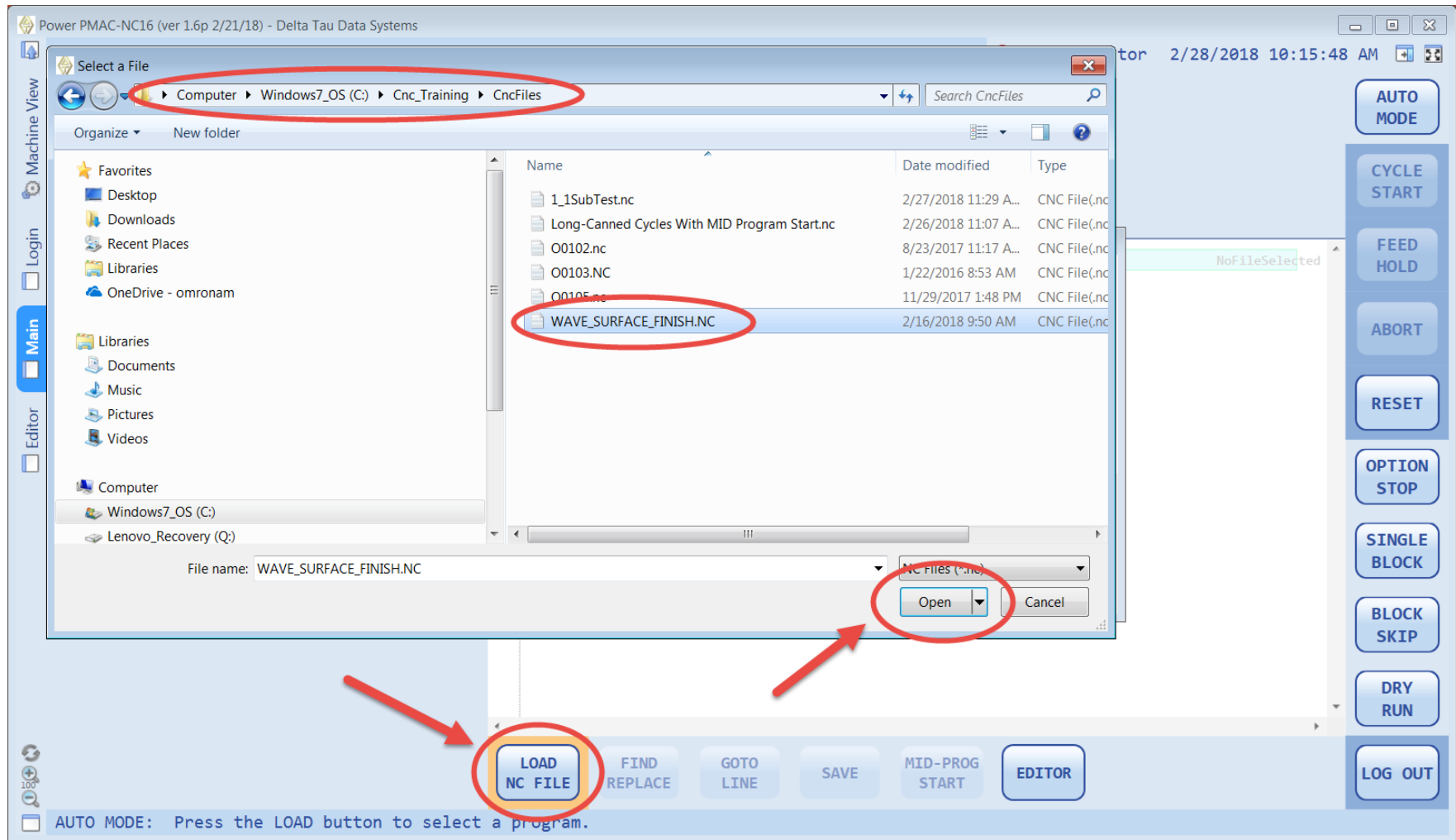
JOG INCREMENTAL:  
-JOG +JOG INCREMENTAL DISTANCE: 5.001

MANUAL MODE  
CYCLE START  
FEED HOLD  
ABORT  
RESET  
OPTION STOP  
SINGLE BLOCK  
BLOCK SKIP  
DRY RUN  
LOG OUT

AUTO MODE: Press the LOAD button to select a program.



# Load a Test File...





# Run the Test Program...

Power PMAC-NC16 (ver 1.6p 2/21/18) - Delta Tau Data Systems

rainning\CncFiles\WAVE\_SURFACE\_FINISH.NC (1:41,939 lines. Download Complete) Administrator 2/28/2018 10:20:00 AM

Machine View

PROG POS REL POS MACH POS CMD POS

X 0.0493

% TORQUE FE: 0.0144 DTG: 0.1089

Y 0.0000

% TORQUE FE: -0.0122 DTG: 0.0000

Z 1.9186

% TORQUE FE: -0.0134 DTG: 0.0000

UNITS INCH (G20)

FEEEDRATE ACT: 0.00 CMD: 500.00

FEED: 100% RAPID: 100% FPM

SPINDLE 0.00 2,000.00

100% CUT

TOOL T01 H04 D00

G-CODES G01 G17 G20 G25 G40 G43 G80 G55

G97 G64 G69 G90 G94 G98

M-CODES M03 M09 M10 M48

FEEEDRATE - + 0% 100%

RAPID - + 0% 100%

SPINDLE - + 100% CW CCW

TOOL CHANGE M06 T00 - + CHANGE

AUTO MODE

CYCLE START

FEED HOLD

ABORT

RESET

OPTION STOP

SINGLE BLOCK

BLOCK SKIP

DRY RUN

LOG OUT

AUTO QUEUE (0) MDI MANUAL WORK OFFSETS TOOL OFFSETS ALARMS (0)

1 (WAVE\_SURFACE\_FINISH)

2 (DATE DD-MM-YY - 22-12-11 TIME=HH:MM - 14:12)

3 (NC FILE - C:\USERS\PUBLIC\DOCUMENTS\PMAC\_COUNTOURING(WAVE\_S

4 (MATERIAL - ALUMINUM INCH - 2024)

5 ( T249 | 1/8 BALL ENDMILL | H249 )

6 N100 G20

7 N102 G0 G17 G40 G49 G80 G90

8 N104 T1

9 N106 G0 G90 G55 X0. Y0 S2852 M3

10 N108 G43 H4 Z2

11 N110 Z2.01855184

12 N112 G1 Z1.91855184 F500

13 N114 X.15822651

14 N116 X.16310516 Z1.91816986

15 N118 Z1.91836114

16 N120 X.1728327 Z1.9175995

17 N122 X.1754129 Z1.91720619

18 N124 X.17768159 Z1.91702856

19 N126 X.17955457 Z1.91657486

20 N128 X.1818616 Z1.9162232

21 N130 X.19254341 Z1.91364137

CLEAR NC FILE FIND REPLACE GOTO LINE SAVE MID-PROG START EDITOR

PROGRAM LOADED: Press the CYCLE START button to run, or CLEAR to select another program.



How does the HMI and Power  
PMAC work together to form one  
application?



```
//PowerPMAC NC State/Command/Status Registers
#define MachineState    M1
#define CommandReg      M2
#define StatusReg       M3
#define JogOptions      M4
#define RunOptions      M5
#define MachineMode     M6
#define HmiCounter      M7
#define DialogResponse  M8
```

- The HMI & PMAC use various parameter to set States, Commands, and Status values.
- These parameters can be found in the ppnc\_ncdefinitions.pmh file.



# State/Command/Status Architecture...

```
#define MachineState M1
```

```
//PowerPMAC NC Machine States
```

```
#define Offline      0
#define NotReady    1
#define Ready       2
#define ProgramLoaded 3
#define Running     4
#define InFeedHold  5
#define Completed   6
#define Failed      7
#define Aborted     8
#define EStopped    9
#define Error       10
#define Downloading 11
```

```
#define MachineMode M6
```

```
//PowerPMAC NC Machine Modes
```

```
#define Auto      0
#define Manual    1
#define MDI       2
```

```
#define CommandReg M2
```

```
//PowerPMAC NC Command Code Structure
```

```
#define UI_None      0
#define UI_Initialize 100
#define UI_Reset     101
#define UI_Home      102
#define UI_Cancel    109
#define UI_JogPlus   110
#define UI_JogMinus  111
#define UI_JogIncPlus 112
#define UI_JogIncMinus 113
#define UI_JogStop    114
```

```
//#define UI_Incremental
```

```
#define UI_CycleStart 120
#define UI_FeedHold   121
#define UI_Abort      122
```

```
#define UI_SetWorkOfs 130
#define UI_SetToolOfs 131
#define UI_ZeroRelPos 132
```



# How does it work...

The HOME button is pressed



The HMI sets the CommandReg code to The UI\_Home value:

Send	On Demand	Command	Response
	<input type="checkbox"/>	&1	
	<input type="checkbox"/>	CommandReg	102
	<input type="checkbox"/>		

The command codes are defined in the ppnc\_ncinterfacedefinitions.pmh file.

```
//PowerPMAC NC State/Command/Status Registers
#define MachineState M1
#define CommandReg M2
#define StatusReg M3
#define JogOptions M4
#define RunOptions M5

//PowerPMAC NC Command Code Structure
#define UI_None 0
#define UI_Initialize 100
#define UI_Reset 101
#define UI_Home 102
#define UI_Cancel 103
#define UI_JogPlus 110
#define UI_JogMinus 111
```

The Homing PLC picks up the command, clears it, does the homing, and once completed, sends the “HomeCompleted” handshake to the HMI.

```
if (CommandReg == UI_Home)
{
    CommandReg = UI_None

    // User Defined
    // Homing Code Here.

    // Acknowledge HMI homing
    // process is complete.
    send1 "HomeCompleted"
}
```



# How are These Used...

These States, Commands, and Status values are used by the PMAC in PLC's to act on the commands sent from the HMI.

```
// ----- Auto and MDI Mode Functions -----  
if (MachineMode == Auto || MachineMode == MDI)  
{  
    if (MachineState==ProgramLoaded || MachineState==InFeedHold || MachineState==Running)  
    {  
        if (CommandReg == UI_CycleStart && Coord[1].ProgActive == 0)  
        {  
            CommandReg = UI_None  
            if (MachineMode == Auto)  
            {  
                if(Coord[1].Program.Number == 100 || Coord[1].Program.Number == 0)  
                {  
                    run  
                    send1 "CycleStarted"  
                }  
            }  
        }  
    }  
}
```



# Send1 Commands...

These commands are sent from PMAC to the HMI via the unsolicited send1"xxxx" format. These commands are used to request modes, actions, HMI features, and to acknowledge initiated command requests.

EstopPressed		
EstopReleased	RequestCycleStart	HideManual
	RequestFeedhold	ShowManual
Initialized	RequestReset	ManualSubmodeNone
Canceled	RequestRewind	ManualSubmodeContinuous
ProgramAborted	RequestAbort	ManualSubmodeHandle
CycleStarted	RequestOptionStop	ManualSubmodeHome
InFeedHold	RequestSingleBlock	
	RequestBlockSkip	PendantConnected
Homecompleted	RequestDryRun	PendantDisconnected
ResetCompleted	RequestSpindleCW	MidProgStart=num
	RequestSpindleCCW	
WorkOffsetsSet	RequestToolChangePlus	Jogging
ToolOffsetsSet	RequestToolChangeMinus	JogStopped
	RequestJogSpeed1..5	
RequestAutoMode	RequestJog1..10	
RequestManualMode	RequestHome	
RequestMdiMode		



# Custom Messages in the HMI...

The ppnc\_messages.plc includes comprehensive examples of how to use each type of message. Use the example variables to trigger these different messages in the HMI.

```
//*****  
//*****Message Type 1 Example*****  
//*****  
if(FatalMsg_Bit_01 == 1)  
{  
    MsgFatal = MsgFatal|MSG_Fatal_01 //Enable first fatal message from Messages.xml file  
}  
else  
{  
    MsgFatal = MsgFatal&~MSG_Fatal_01 //Clear first fatal message from Messages.xml file  
}  
  
if(WarnMsg_Bit_01 == 1)  
{  
    MsgWarning = MsgWarning|MSG_Warning_01 //Enable first warning message from Messages.xml file  
}  
else  
{  
    MsgWarning = MsgWarning&~MSG_Warning_01 //Clear first warning message from Messages.xml file  
}  
  
if(InfoMsg_Bit_01 == 1)  
{  
    MsgInfo = MsgInfo|MSG_Information_01 //Enable first information message from Messages.xml file  
}  
else  
{  
    MsgInfo = MsgInfo&~MSG_Information_01 //Clear first information from Messages.xml file  
}
```



## The SDK



# Power PMAC NC16-SDK

PowerPmacNC - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Design Format Tools Test Analyze Window Help

Debug Any CPU PowerPmacNC Start

PageMain.xaml Main.cs

(no file selected)

Administrator Date and Time

FEEDRATE RAPID SPINDLE TOOL CHANGE

0% 0% 100% M06 T01

100% 100% CW CCW CHANGE

AUTO FEED HOLD ABORT RESET

OPTION STOP SINGLE BLOCK BLOCK SKIP DRY RUN LOG OUT

STICKY COMMENT

66.67% Design XAML

<Page x:Class="PowerPmacNC.PageMain"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Output

Show output from: Source Control - Git

Error List Output

This item does not support previewing

0 99% PowerPmacNc16-SDK Beta master

Solution Explorer

Search Solution Explorer (Ctrl+):

- Skins
- SupportClasses
- VCPP4
- app.config
- DeltaTau.ico
- DeviceMembers.xml
- Enumerations.cs
- GCodes.cs
- Machine1.cs
- Machine2.cs
- Main.cs
- EntryPoint
- MCodes.cs
- PageEditor.xaml
- PageLogin.xaml
- PageMacroVars.xaml
- PageMain.xaml
- PageProduction.xaml
- PowerPmacController.cs
- Reference MacroAssignments.txt
- Reference Messages.xml

Solution Explorer Team Explorer

Properties

PageMain.xaml File Properties

Build Action	Page
Copy to Output Directory	Do not copy
Custom Tool	XamlIntelliSenseFileGenerator
Custom Tool Namespace	
File Name	PageMain.xaml
Full Path	C:\Cnc_Training\PowerPmacNc16-SDK

Build Action

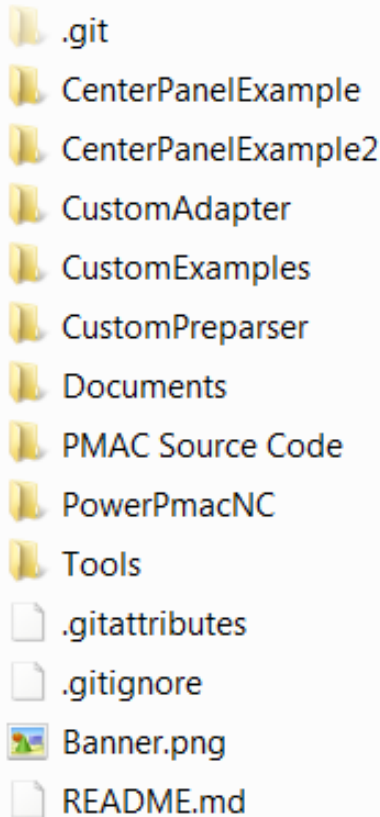
How the file relates to the build and deployment processes.



## Documents library

PowerPmacNc16-SDK

Name



**SDK vs. Runtime** – The SDK package will contain the same files as the Runtime, but will include the source code files and examples for the Visual Studio project.

**CenterPanelExample** – A custom center panel example.

**CustomAdapter** – Source code example of how to create an MTConnect adapter.

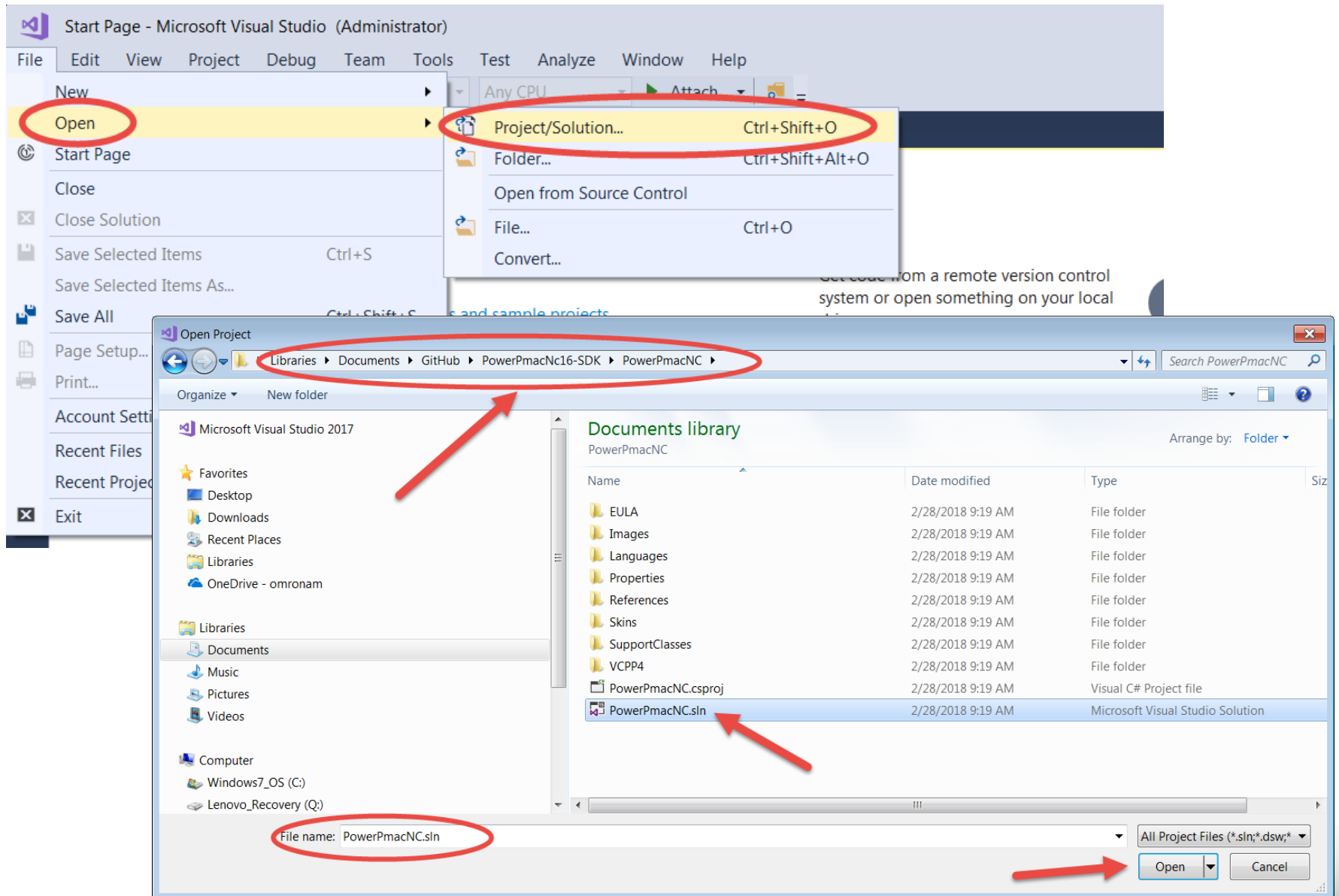
**CustomExamples** – Various custom examples.

**CustomPreparser** – Custom Pre-Parser example.

**Tools** – A comprehensive example of how to comprehensively change the functionality of the HMI.

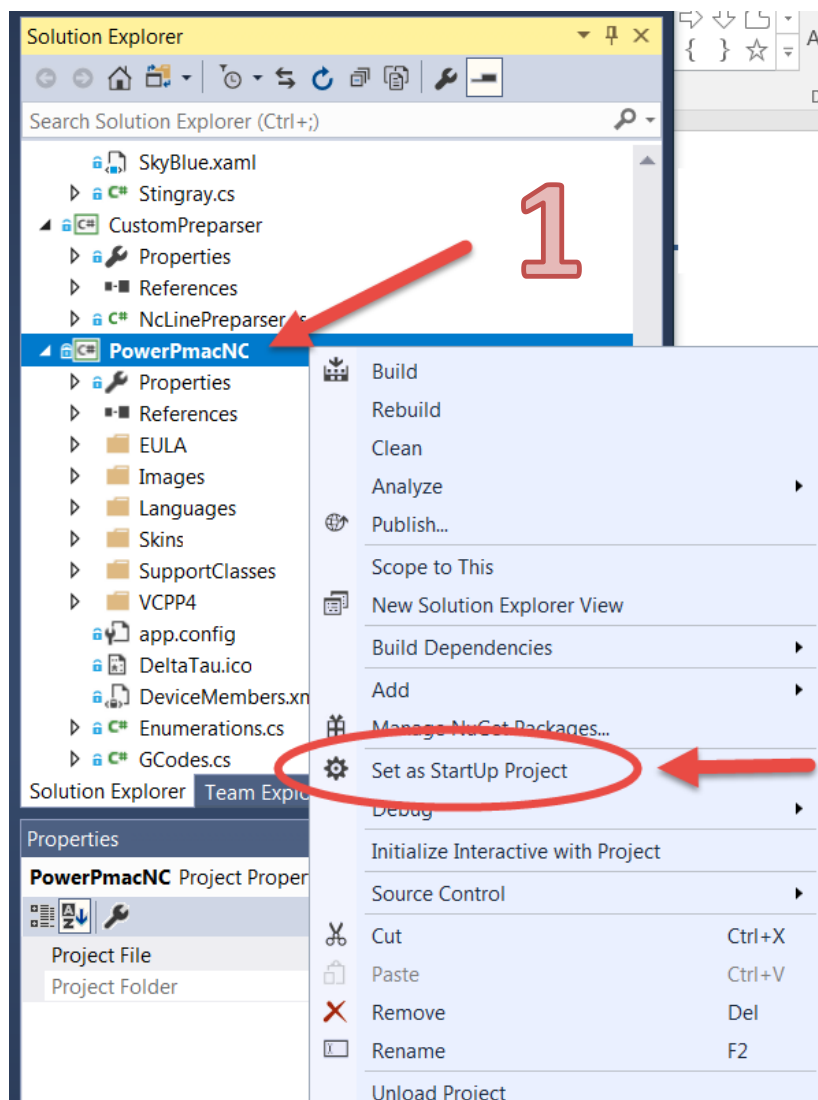


# Loading the SDK Project into VS Studio...





# Building the SDK for the first time...

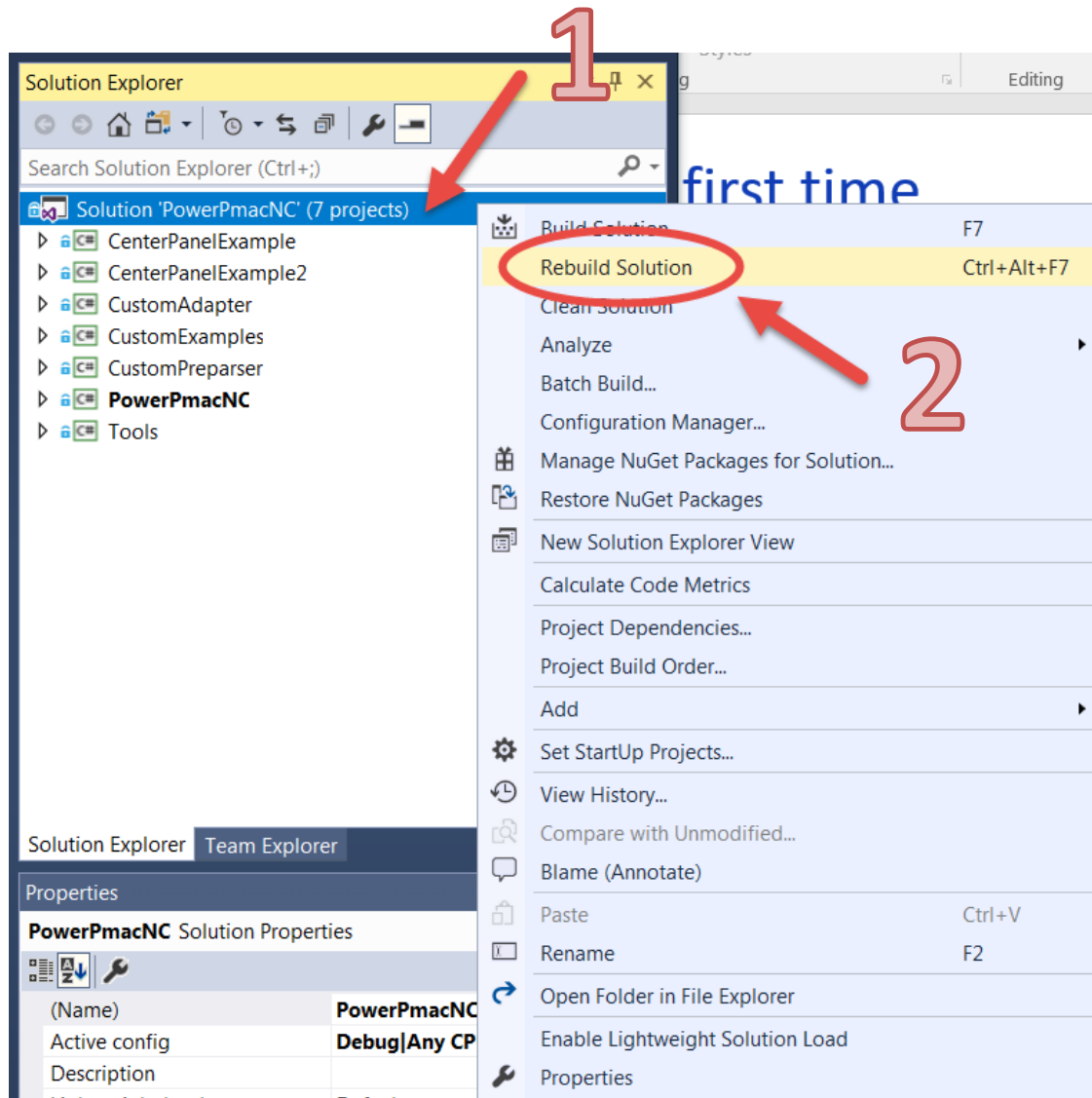


From the VS Studio Solution Explorer:

- Right Click “PowerPmacNC”
- Click “Set as StartUp Project”



# Building the SDK for the first time (cont.)...

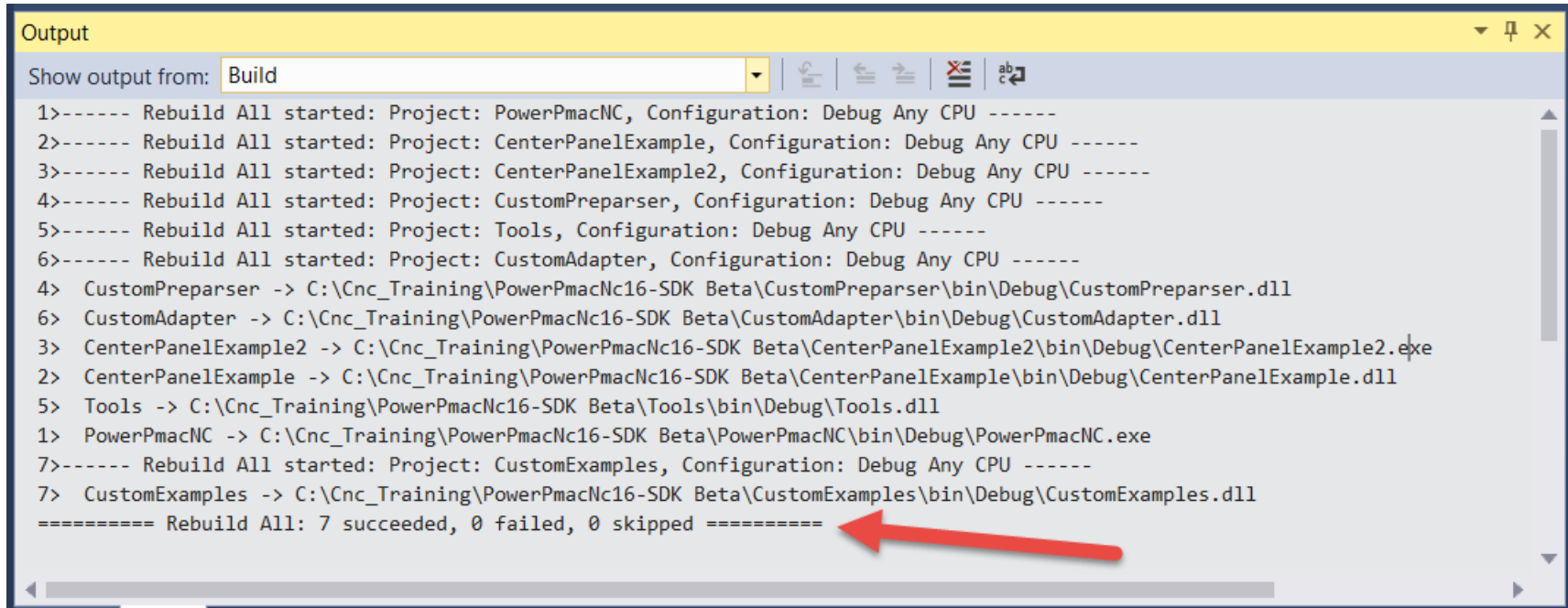


From the VS Studio Solution Explorer:

- Right Click “Solution ‘PowerPmacNC’ ”
- Click “Rebuild Solution”



# Checking Build...

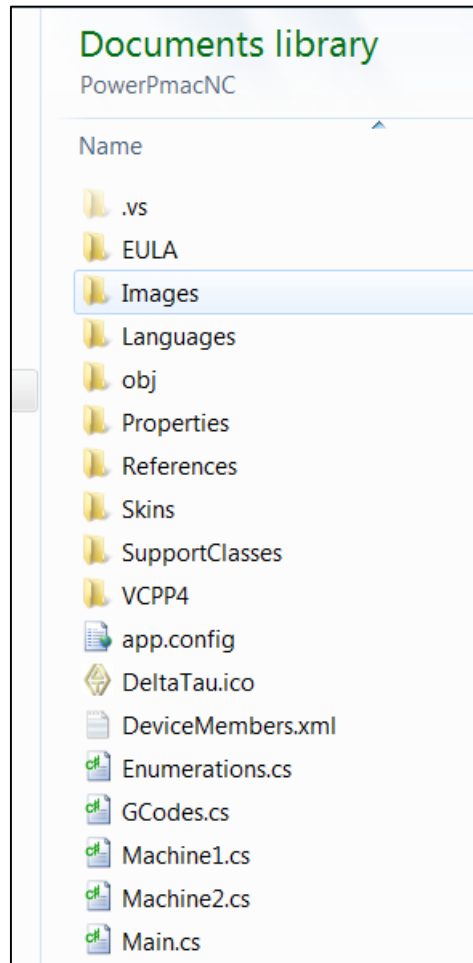


```
Output
Show output from: Build
1>----- Rebuild All started: Project: PowerPmacNC, Configuration: Debug Any CPU -----
2>----- Rebuild All started: Project: CenterPanelExample, Configuration: Debug Any CPU -----
3>----- Rebuild All started: Project: CenterPanelExample2, Configuration: Debug Any CPU -----
4>----- Rebuild All started: Project: CustomPreparser, Configuration: Debug Any CPU -----
5>----- Rebuild All started: Project: Tools, Configuration: Debug Any CPU -----
6>----- Rebuild All started: Project: CustomAdapter, Configuration: Debug Any CPU -----
4> CustomPreparser -> C:\Cnc_Training\PowerPmacNc16-SDK Beta\CustomPreparser\bin\Debug\CustomPreparser.dll
6> CustomAdapter -> C:\Cnc_Training\PowerPmacNc16-SDK Beta\CustomAdapter\bin\Debug\CustomAdapter.dll
3> CenterPanelExample2 -> C:\Cnc_Training\PowerPmacNc16-SDK Beta\CenterPanelExample2\bin\Debug\CenterPanelExample2.exe
2> CenterPanelExample -> C:\Cnc_Training\PowerPmacNc16-SDK Beta\CenterPanelExample\bin\Debug\CenterPanelExample.dll
5> Tools -> C:\Cnc_Training\PowerPmacNc16-SDK Beta\Tools\bin\Debug\Tools.dll
1> PowerPmacNC -> C:\Cnc_Training\PowerPmacNc16-SDK Beta\PowerPmacNC\bin\Debug\PowerPmacNC.exe
7>----- Rebuild All started: Project: CustomExamples, Configuration: Debug Any CPU -----
7> CustomExamples -> C:\Cnc_Training\PowerPmacNc16-SDK Beta\CustomExamples\bin\Debug\CustomExamples.dll
===== Rebuild All: 7 succeeded, 0 failed, 0 skipped =====
```

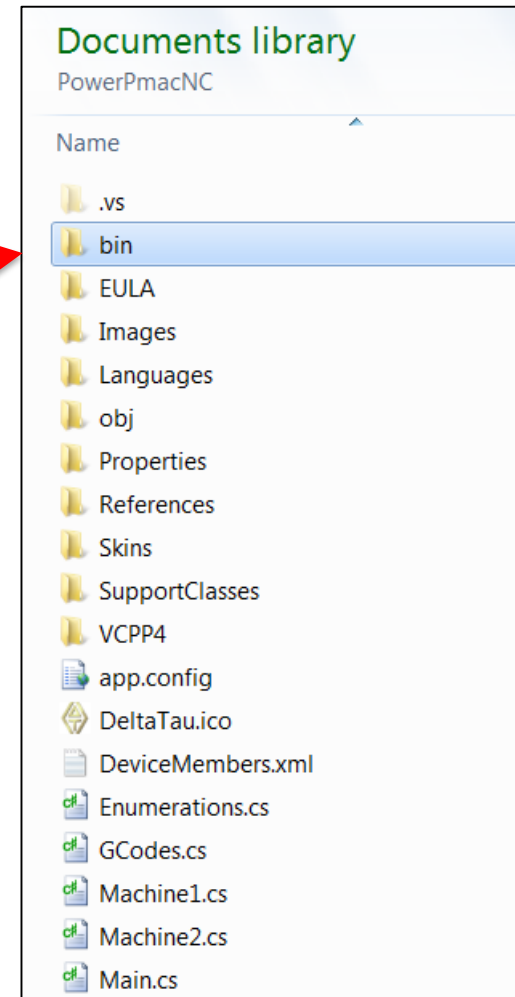
- Verify the build completed in the Output window.
- Insure all 7 projects were successfully built.



# Folder Structure Post Build...

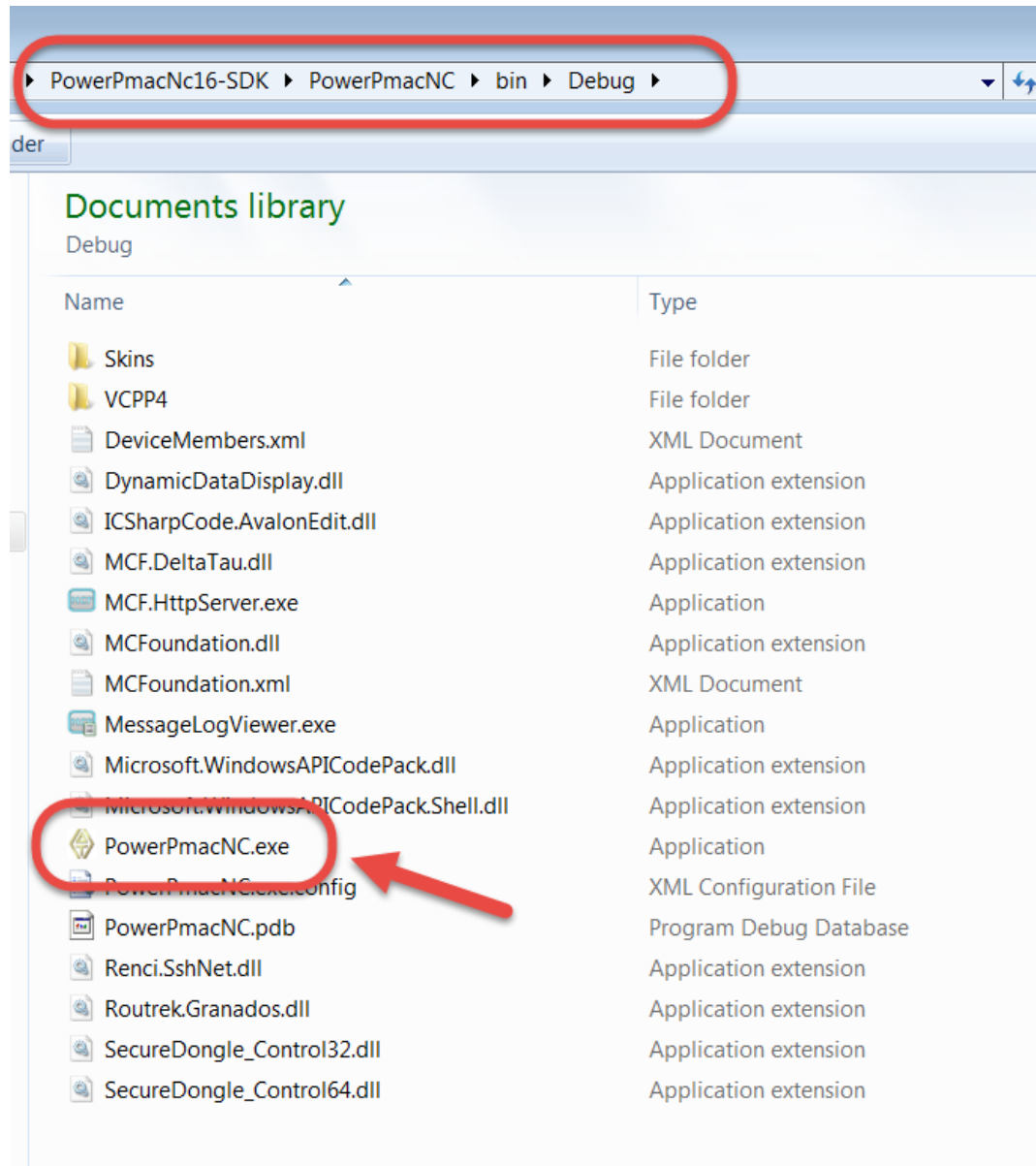


After initial build a new “bin” folder will be created. Within this folder you will find the newly created folder which will include the Runtime files.





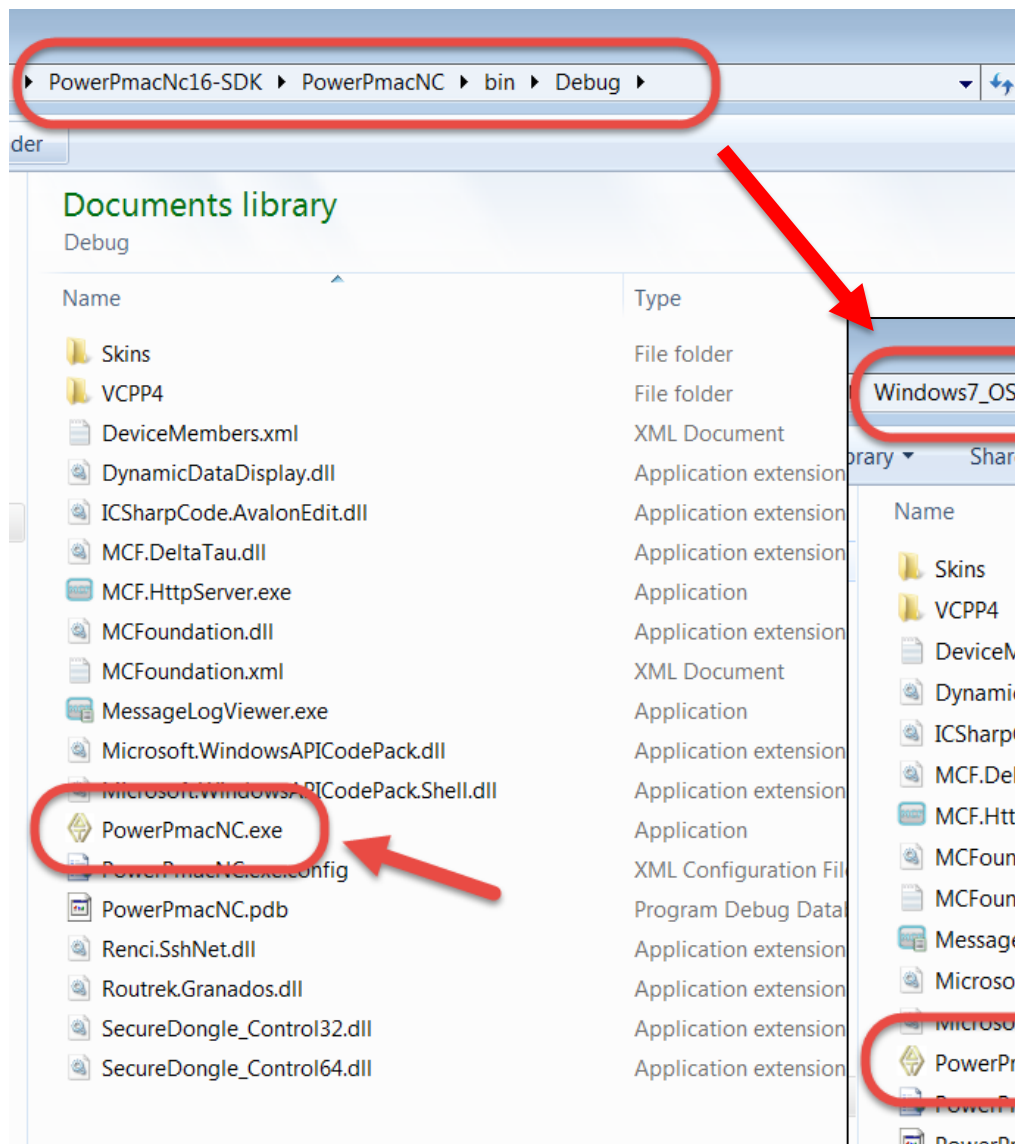
# bin -> Debug -> PowerPmacNC.exe



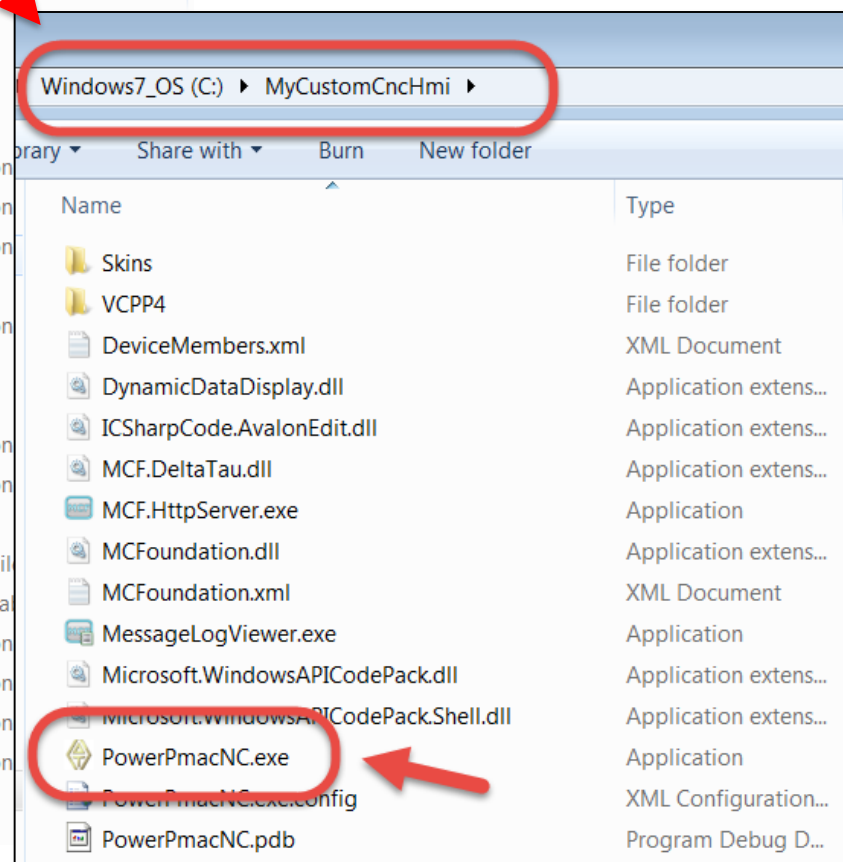
Within the 'bin' folder you will find the 'Debug' folder. The PowerPmacNC executable code will be found in this folder.



# Fully Portable Executable Folder



The 'debug' folder can be copied, moved, and renamed without any issues.





# Adding Reference Files...

PowerPmacNc16-SDK ▶ PowerPmacNC ▶

Documents library  
PowerPmacNC

Name	Type
PageMain.xaml.cs	Visual C# Source File
PowerPmacController.cs	Visual C# Source File
PowerPmacNC.csproj	Visual C# Project File
PowerPmacNC.sln	Microsoft Visual Studio Solution File
Reference MacroAssignments.txt	Text Document
Reference Messages.xml	XML Document
Reference PowerPmacNC.ini	Configuration File
SecureDongle_Control32.dll	Application Extension
SecureDongle_Control64.dll	Application Extension
SkyBlue.xaml	Windows Markup Extension
UserControlEditor.xaml	Windows Markup Extension
UserControlEditor.xaml.cs	Visual C# Source File

PowerPmacNc16-SDK ▶ PowerPmacNC ▶ bin ▶ Debug ▶

E-mail Burn New folder

Documents library  
Debug

Name	Type
Skins	File folder
VCPP4	File folder
DeviceMembers.xml	XML Document
DynamicDataDisplay.dll	Application extension
ICSharpCode.AvalonEdit.dll	Application extension
MacroAssignments.txt	Text Document
MCF.DeltaTau.dll	Application extension
MCF.HttpServer.exe	Application
MCFoundation.dll	Application extension
MCFoundation.xml	XML Document
MessageLogViewer.exe	Application
Messages.xml	XML Document
Microsoft.WindowsAPICodePack.dll	Application extension
Microsoft.WindowsAPICodePack.Shell.dll	Application extension
PowerPmacNC.exe	Application
PowerPmacNC.exe.config	XML Configuration File
PowerPmacNC.ini	Configuration Settings
PowerPmacNC.pdb	Program Debug Database

Copy the reference files from the base folder to the executable folder and rename.



**External Assemblies are separate 'projects' which are initialized via the .ini file during application startup. These assemblies are very useful for customizing the HMI.**



# Custom Center Panel...

Power PMAC-NC16 (ver 1.6p 2/21/18) - Delta Tau Data Systems

C:\Cnc\_Training\CncFiles\WAVE\_SURFACE\_FINISH.NC

Administrator 2/28/2018 3:10:48 PM

**X** 0.9834 **Laser**  
% TORQUE FE: 0.0036 DTG: 0.0028 PEAK POWER

**Y** 0.0000 **0 W**  
% TORQUE FE: -0.0122 DTG: 0.0000

**Z** 1.7819 **FREQUENCY**  
% TORQUE FE: 0.0074 DTG: -0.0007 **0.00 kHz**

UNITS INCH (G20)  
FEEDRATE ACT: 0.00 CMD: 500.00  
FEED: 100% RAPID: 100% FPM  
SPINDLE 0.00 2,000.00  
100% CUT

TOOL T01 H04 D00  
G-CODES G01 G17 G20 G25 G40 G43 G80 G55  
G97 G64 G69 G90 G94 G98

M-CODES M03 M09 M10 M48

**0.00 ms**  
PULSE DURATION

**0.00 W**  
AVERAGE POWER

**0 J**  
PULSE ENERGY

**GAS** **WATER** **LASER**

**CYCLE START** **FEED HOLD**

FEEDRATE RAPID SPINDLE TOOL CHANGE  
- + - + - + M06 T00  
0% 0% 100% - +  
100% 100% CW CCW CHANGE

AUTO QUEUE (0) MDI MANUAL WORK OFFSETS TOOL OFFSETS ALARMS (0)

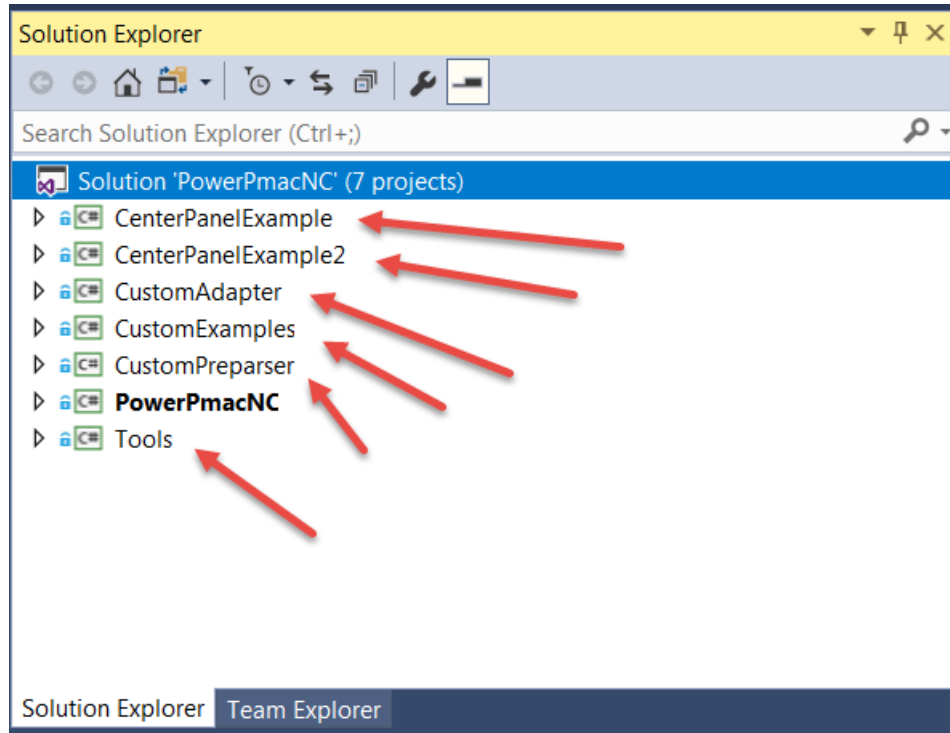
1 %  
2 00000(WAVE\_SURFACE\_FINISH)  
3 (DATE=DD-MM-YY - 22-12-11 TIME=HH:MM - 14:12)  
4 (NC FILE - C:\USERS\PUBLIC\DOCUMENTS\PMAC\_COUNTOURING\WAVE\_SURFACE\_F  
5 (MATERIAL - ALUMINUM INCH - 2024)  
6 ( T249 | 1/8 BALL ENDMILL | H249 )  
7 N100 G20  
8 N102 G0 G17 G40 G49 G80 G90  
9 N104 T1  
10 N106 G0 G90 G55 X0. Y0 S2852 M3  
11 N108 G43 H4 Z2  
12 N110 Z2.01855184  
13 N112 G1 Z1.91855184 F500  
14 N114 X.15822651  
15 N116 X.16310516 Z1.91816986  
16 N118 Z1.91836114  
17 N120 X.1728327 Z1.9175995  
18 N122 X.1754129 Z1.91720619  
19 N124 X.17768159 Z1.91702856  
20 N126 X.17955457 Z1.91657486  
21 N128 X.1818616 Z1.9162232  
22 N130 X.19254341 Z1.91364137  
23 N132 X.19492479 Z1.91285171  
24 N134 X.19743243 Z1.91224428  
25 N136 X.19980617 Z1.91123305  
26 N138 X.20232477 Z1.91039789  
27 N140 X.21193799 Z1.9063026  
28 N142 X.21456041 Z1.90511332  
29 N144 X.21580213 Z1.90441867  
30 N146 X.21665886 Z1.9040537

CLEAR NC FILE FIND REPLACE GOTO LINE SAVE MID-PROG START EDITOR

PROGRAM LOADED: Press the CYCLE START button to run, or CLEAR to select another program.



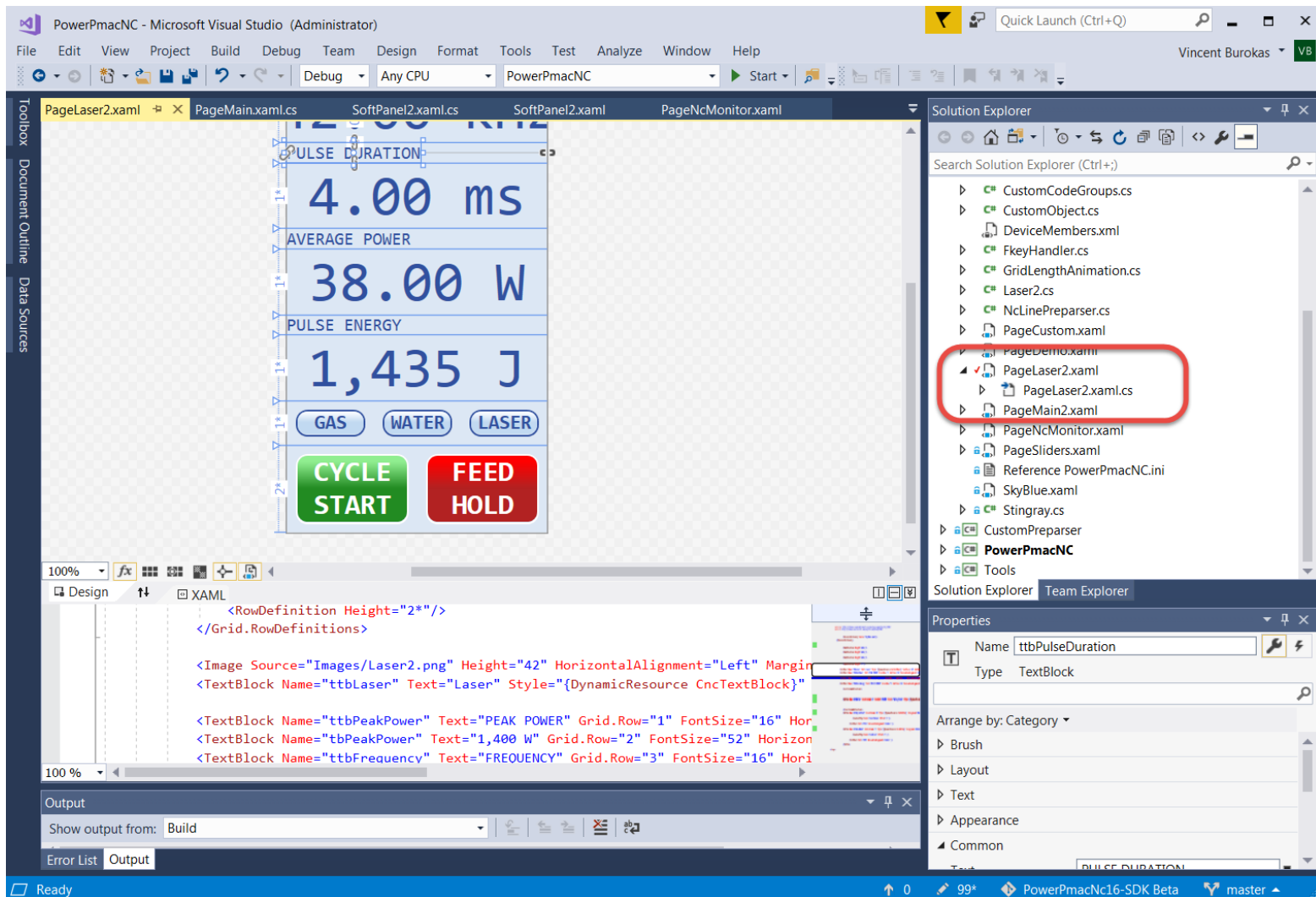
# External Assembly Examples



The SDK package includes various external assembly examples. These can be used to customize your application visually, as well as provide sophisticated Parser modification ability.

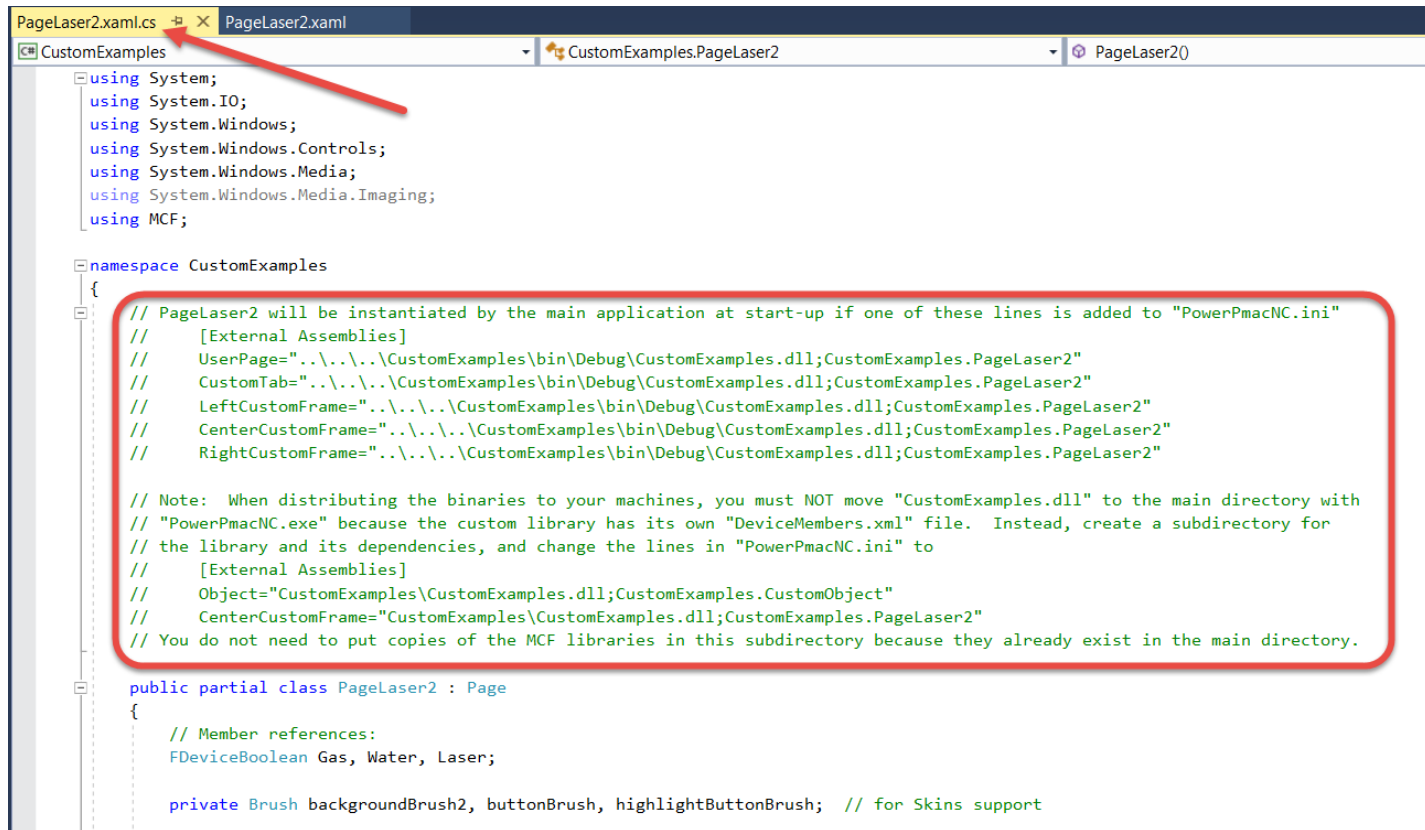


# PageLaser2 External Assembly Example...





# .ini Header Instructions...



```
PageLaser2.xaml.cs PageLaser2.xaml
CustomExamples CustomExamples.PageLaser2 PageLaser2()

using System;
using System.IO;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using MCF;

namespace CustomExamples
{
    // PageLaser2 will be instantiated by the main application at start-up if one of these lines is added to "PowerPmacNC.ini"
    // [External Assemblies]
    // UserPage="..\..\..\CustomExamples\bin\Debug\CustomExamples.dll;CustomExamples.PageLaser2"
    // CustomTab="..\..\..\CustomExamples\bin\Debug\CustomExamples.dll;CustomExamples.PageLaser2"
    // LeftCustomFrame="..\..\..\CustomExamples\bin\Debug\CustomExamples.dll;CustomExamples.PageLaser2"
    // CenterCustomFrame="..\..\..\CustomExamples\bin\Debug\CustomExamples.dll;CustomExamples.PageLaser2"
    // RightCustomFrame="..\..\..\CustomExamples\bin\Debug\CustomExamples.dll;CustomExamples.PageLaser2"

    // Note: When distributing the binaries to your machines, you must NOT move "CustomExamples.dll" to the main directory with
    // "PowerPmacNC.exe" because the custom library has its own "DeviceMembers.xml" file. Instead, create a subdirectory for
    // the library and its dependencies, and change the lines in "PowerPmacNC.ini" to
    // [External Assemblies]
    // Object="CustomExamples\CustomExamples.dll;CustomExamples.CustomObject"
    // CenterCustomFrame="CustomExamples\CustomExamples.dll;CustomExamples.PageLaser2"
    // You do not need to put copies of the MCF libraries in this subdirectory because they already exist in the main directory.

    public partial class PageLaser2 : Page
    {
        // Member references:
        FDeviceBoolean Gas, Water, Laser;

        private Brush backgroundBrush2, buttonBrush, highlightButtonBrush; // for Skins support
    }
}
```

- Each example will include a .cs file.
- At the top of each .cs file will be instructions on how to include the header information in the .ini file.
- Copy and paste this header information into the .ini file.



- The .ini External Assembly section will look like the following:

```
[External Assemblies]

;CenterCustomFrame="..\..\..\CustomExamples\bin\Debug\CustomExamples.dll;CustomExamples.PageLaser2"
;Object="..\..\..\CustomExamples\bin\Debug\CustomExamples.dll;CustomExamples.Laser2"
```



# Comprehensive Customized HMI Example...

Power PMAC-NC16 (ver 1.2e 8/25/16) - Delta Tau Data Systems

C:\Cnc\_Progs\FUNKY\_SURFACE\_FINISH.NC

PROG POS REL POS **MACH POS** CMD POS

X 25.0000  
% TORQUE FE: 0.0000 JOG SPEED: x1

Y 24.9920  
% TORQUE FE: -0.0003

Z 17.0000  
% TORQUE FE: 0.0000

UNITS MM (G21)  
FEEDRATE ACT: 0.00 CMD: 99,903.00  
FEED: 100% RAPID: 100% EDM

**POWER** 0.00 0.00  
100% CUT

T-CODE T00 T01 T02 T03 T04 T05 T06 T07 T08 T09 T10 T11 T12 T13 T14 T15 T16 T17 T18 T19 T20 T21 T22 T23 T24 T25

G-CODES G09 G00 G17 G21 G25 G41 G49 G72 Gxx G97 G64 G69 G90 G94

M-CODES M30 M19 M09 M10 M48

**Laser**

Power (watts)

Frequency (kHz)

ENABLE

SHUTTER

EMISSION

SPEED MODULATION

SPEED OVERRIDE RAPID OVERRIDE **POWER OVERRIDE** LASER PARAMETER

- + - + - + Change to T00

0% 0% 0% - +

100% 100% 100% CHANGE

AUTO QUEUE (0) MDI MANUAL WORK OFFSETS **LASER PARAMS** ALARMS (0)

T-Code	Speed	Power	Frequency	Height	Offset	Path	Offset
Parameter T01	0	0	0	0	0.0000	0.0000	
Parameter T02	0	0	0	0	0.0000	1.5875	
Parameter T03	0	0	0	0	0.0000	0.0000	
Parameter T04	0	0	0	0	0.0000	0.0000	
Parameter T05	0	0	0	0	0.0000	0.0000	
Parameter T06	0	0	0	0	0.0000	0.0000	
Parameter T07	0	0	0	0	0.0000	0.0000	
Parameter T08	0	0	0	0	0.0000	0.0000	
Parameter T09	0	0	0	0	0.0000	0.0000	
Parameter T10	0	0	0	0	0.0000	0.0000	
Parameter T11	0	0	0	0	0.0000	0.0000	
Parameter T12	0	0	0	0	0.0000	0.0000	
Parameter T13	0	0	0	0	0.0000	2.7940	
Parameter T14	0	0	0	0	0.0000	0.0000	
Parameter T15	0	0	0	0	0.0000	0.0000	
Parameter T16	0	0	0	0	0.0000	0.0000	
Parameter T17	0	0	0	0	0.0000	0.0000	
Parameter T18	0	0	0	0	0.0000	0.0000	
Parameter T19	0	0	0	0	0.0000	0.0000	
Parameter T20	0	0	0	0	0.0000	0.0000	
Parameter T21	0	0	0	0	0.0000	0.0000	
Parameter T22	0	0	0	0	0.0000	0.0000	
Parameter T23	0	0	0	0	0.0000	0.0000	
Parameter T24	0	0	0	0	0.0000	0.0000	
Parameter T25	0	0	0	0	0.0000	0.0000	

SET HEIGHT OFFSET: (Manual Mode Only)

SET HEIGHT OFFSET

MANUAL MODE

CYCLE START

FEED HOLD

ABORT

RESET

OPTION STOP

SINGLE BLOCK

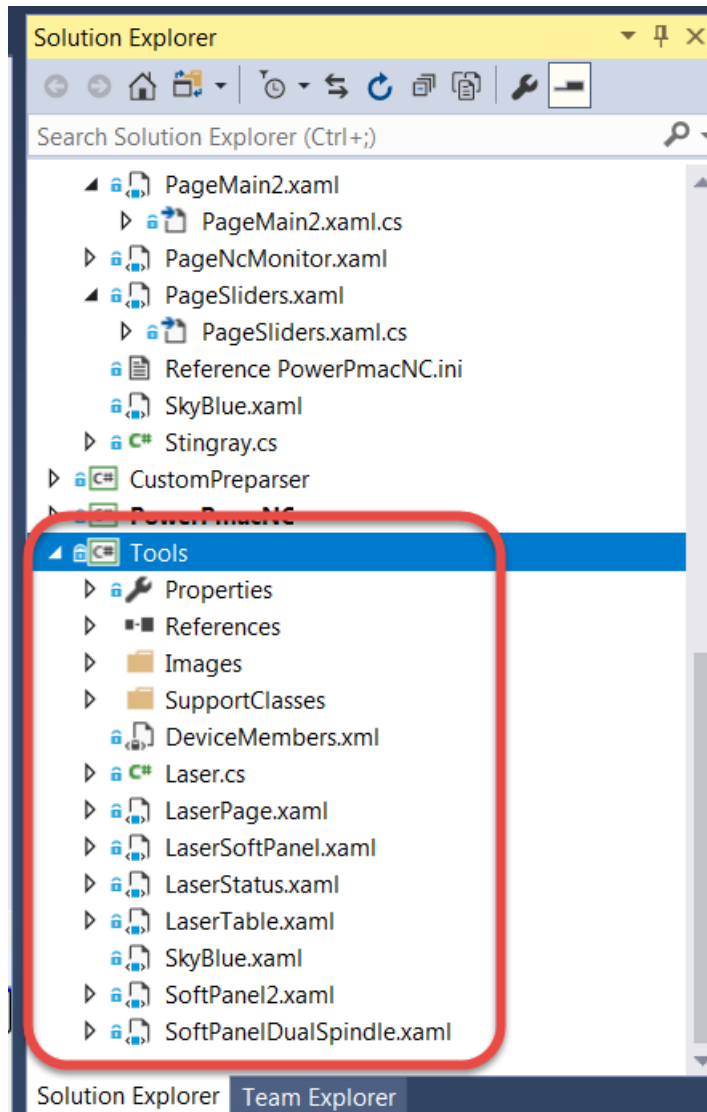
BLOCK SKIP

DRY RUN

LOG OUT

AUTO MODE: Press the LOAD button to select a program.





The Tools Example Provides sophisticated modification of the entire Main screen.

- All references to the spindle are removed.
- A custom center panel is added.
- Laser Power controls are added to the software control panel.
- The Tool Offsets data grid is completely replaced by a laser specific xml data grid.



```
[Machine Constructor]
;MachineType=Standard
MachineType=Custom

[External Assemblies]
Tool="..\..\..\Tools\bin\Debug\Tools.dll;Tools.Laser"
ToolTable="..\..\..\Tools\bin\Debug\Tools.dll;Tools.LaserParameters"
SoftPanel="..\..\..\Tools\bin\Debug\Tools.dll;Tools.LaserSoftPanel"
StatusPanel="..\..\..\Tools\bin\Debug\Tools.dll;Tools.LaserStatus"
ToolTab="..\..\..\Tools\bin\Debug\Tools.dll;Tools.LaserTable"
CenterCustomFrame="..\..\..\Tools\bin\Debug\Tools.dll;Tools.LaserPage"
CustomTab="..\..\..\Tools\bin\Debug\Tools.dll;Tools.LaserPage"
```