



# DELTA TAU

## Power PMAC NC 2014

Motion Commander Foundation  
©2014 Greene & Morehead Engineering, Inc.



# Developer's Guide

## Power PMAC NC 2014



Delta Tau Data Systems, Inc.  
September, 2014

## **COPYRIGHT INFORMATION**

Software: © 2014 Delta Tau Data Systems, Inc. All rights reserved.

Developer's Guide: © 2014 Delta Tau Data Systems, Inc. All rights reserved.

Motion Commander Foundation: © 2012-2014 Greene & Morehead Engineering, Inc. All rights reserved.

This document contains proprietary information of Delta Tau Data Systems, Inc. The information contained herein is not to be used by or disclosed to third parties without the express written permission of an officer of Delta Tau Data Systems, Inc.

## **TRADEMARK ACKNOWLEDGMENT**

Windows, Visual Studio and .NET Framework are registered trademarks of Microsoft Corporation. MTConnect is a registered trademark of the MTConnect Institute. Other brands, product names, company names, trademarks and service marks are the properties of their respective holders.

## **REVISION HISTORY**

<b>Version</b>	<b>Date</b>	<b>Description</b>
1.0	9/1/2014	Initial release

# Power PMAC NC™ - Developer's Guide

---

## Contents

Introduction .....	5
Motion Commander Foundation .....	6
Requirements.....	6
Installation .....	6
Deployment.....	7
The Power PMAC Project .....	7
Runtime Operation .....	8
Hardware Key.....	8
Configuration File.....	8
Machine View .....	9
Users .....	10
Foreign Language Support .....	10
Skins .....	11
Ctrl and Shift Keys .....	11
NC Files.....	12
The NC File Parser .....	12
NC File Configuration .....	12
Subprograms .....	13
Native PMAC Commands and Expressions .....	13
G and M-Code Groups .....	14
Mid-Program Start .....	15
Fixed Cycles.....	15
Using M99 to Repeat the Main Program .....	16
The NC Program Queue .....	16
NC File Comments.....	16
NC File Size Limitations .....	16
Customizing the Application .....	17
Private Labeling.....	17
Messages.xml.....	18
External Assemblies .....	18
The Visual Studio Project Template.....	19

Appendix A. The Source Files ..... 20

Appendix B. The Configuration File..... 22

Appendix B. Turbo PMAC Support ..... 23

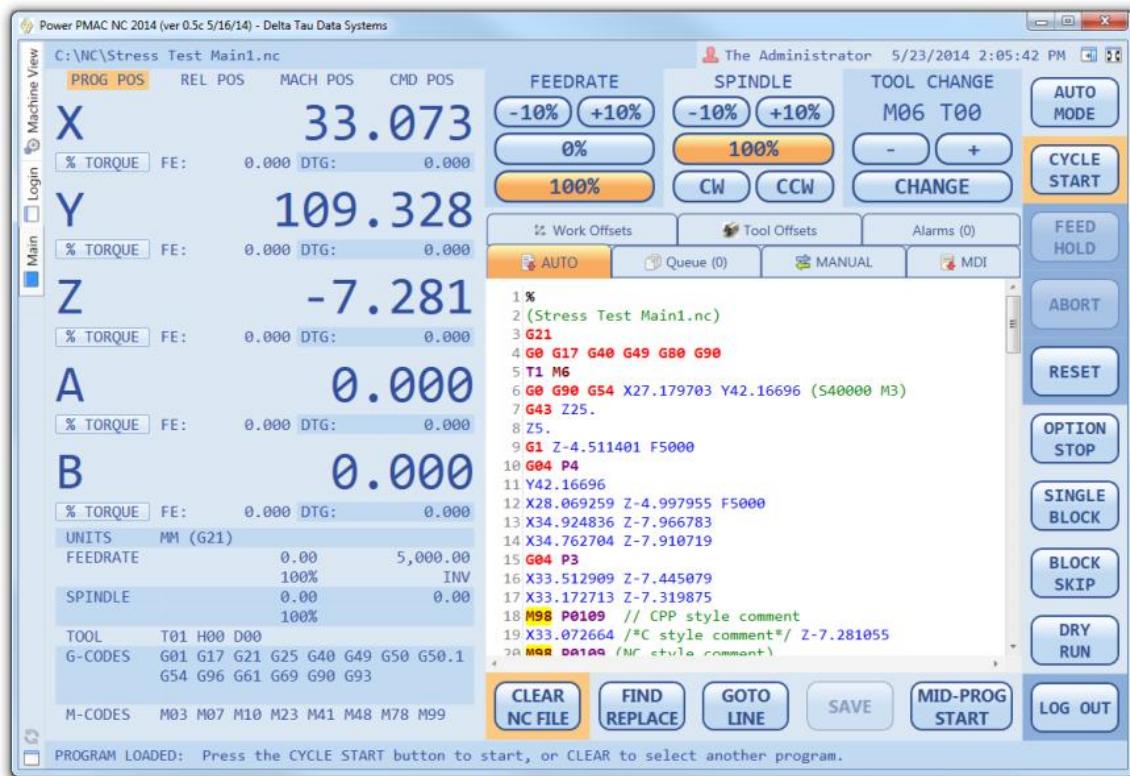
    The Turbo PMAC Project..... 23

Appendix C. Source Code Exclusions..... 24

# Power PMAC NC™ - Developer's Guide

## Introduction

The *Power PMAC NC* program is a host PC application for Delta Tau Power PMAC controlled CNC machines. This document is the Developer's Guide for the *Power PMAC NC Software Development Kit* (SDK). It contains information about how to install, configure, operate, and customize the Power PMAC NC program and the companion Power PMAC project.



- Configurable for 1-10 axes, choice of Mill or Laser, and INCH or MM
- Secure SSH/SFTP communications with Power PMAC
- Built-in Power PMAC command terminal and Linux terminal
- Colorized NC file editor optimized for large files
- NC file Execution Queue with drag-and-drop support
- Real-time Execution Monitoring including Subprograms
- Mid-Program Start
- Manual Mode for Jogging and Homing, and MDI with Execution Monitoring
- User Login system with Foreign Language Translation
- MTConnect 1.1 Agent for supervisory data collection
- Simply copy the files to install for easy deployment (5MB total file size!)
- Fully customizable with the *Software Development Kit* (SDK)
- External assembly (plugin) system for custom Screens, Code Groups and other data.

## Motion Commander Foundation

The Power PMAC NC program is based on the *Motion Commander Foundation* (MCF) .NET framework for machine control applications. The *Motion Commander Foundation Developer's Guide* is also included in the SDK for reference. <http://www.MotionCommander.com>



## Requirements

The Power PMAC NC program is compatible with Windows 7 or newer (64-bit or 32-bit).

Visual Studio 2013 or newer (Express or Professional) is required to build the application from the SDK source code. The application requires .NET 4.0 and the Visual C++ 2010 runtime libraries. Visual Studio will install these components automatically. The following links will be helpful if the application fails to run on a PC without Visual Studio installed.

### *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>

 Some versions of the Power PMAC NC program also support the Turbo PMAC and are 32-bit applications, requiring the x86 version of the Visual C++ Redistributable - even on 64-bit Windows systems.

## Installation

The *Power PMAC NC Software Development Kit* (SDK) is distributed via a private GitHub repository. Sign up for a [free GitHub account](#) and give your account name to **Delta Tau Data Systems** when you purchase the SDK. You will be given read-only access to the repository. Install [GitHub for Windows](#) on your development PC, log in, and "Clone" the repository. You will want to "Sync" occasionally to update your copy of the SDK to the latest release version.



 It is highly recommended that you make working copies of both the *Power PMAC NC* source code and the Power PMAC project in order to avoid losing your edits when you Sync. If a Sync fails for any reason, simply delete the entire "GitHub\PowerPmacNcRelease" folder and Clone again.

## Deployment

The *Power PMAC NC* application's "PowerPmacNC\bin\Debug" folder may simply be copied and renamed for deployment to your machines. Your distribution must include the files shown below. (Files not shown in this list may be deleted without affecting the application.)


Name	Type	Size
Languages	File folder	
MessageLogViewer.exe	Application	52 KB
PowerPmacNC.exe	Application	974 KB
PowerPmacNC.ini	Configuration settings	3 KB
DynamicDataDisplay.dll	DLL File	350 KB
ICSharpCode.AvalonEdit.dll	DLL File	612 KB
MCF.CustomControls.dll	DLL File	12 KB
MCF.DeltaTau.dll	DLL File	51 KB
MCFoundation.dll	DLL File	1,084 KB
Microsoft.WindowsAPICodePack.dll	DLL File	104 KB
Microsoft.WindowsAPICodePack.Shell.dll	DLL File	530 KB
Renci.SshNet.dll	DLL File	450 KB
Routrek.Granados.dll	DLL File	136 KB
SecureDongle_Control32.dll	DLL File	111 KB
SecureDongle_Control64.dll	DLL File	146 KB
PowerPmacNC.pdb	Program Debug Database	194 KB
DeviceMembers.xml	XML Document	107 KB
Messages.xml	XML Document	14 KB
PowerPmacNC_Settings.xml	XML Document	21 KB

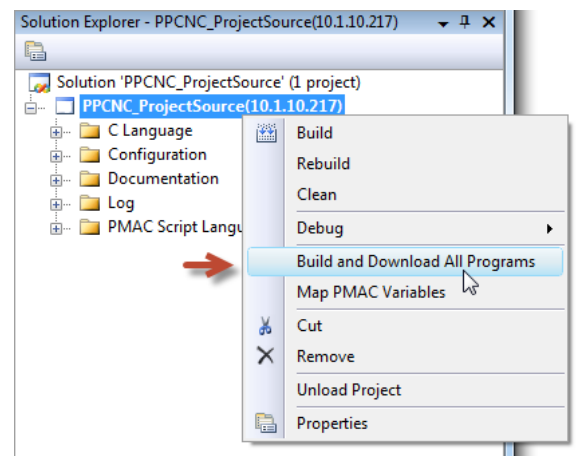
## The Power PMAC Project

The Power PMAC project will be located in the "GitHub\PowerPmacNcRelease\PMAC Source Code\PowerPMAC" folder. Make a working copy of this directory before you download the project to the controller.

Open the "PPCNC\_ProjectSource.PowerPmacSuite\_sln" solution file in the *Power PMAC IDE*, right-click and select "Build and Download" as shown. Look for the "Download Successful" message in the Output window.

```
Download Successful.  
Total Project Download time = 13.057 sec  
Total Project Build and Download time = 24.212 sec
```

 After downloading the project, use the Terminal window to issue a "save" command to copy the project to nonvolatile flash memory, then issue a "\$\$\$" command to reset the controller.



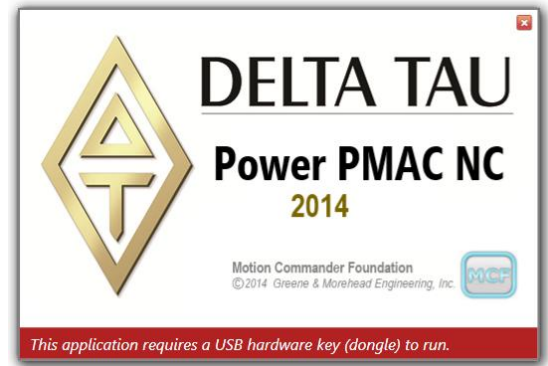
The Power PMAC controller is now ready to work with the Power PMAC NC program!

## Runtime Operation

### Hardware Key

This application requires a USB hardware key (dongle) to run. You will receive hardware keys when you purchase copies of the Power PMAC NC program from Delta Tau Data Systems.

The hardware key is compatible with all versions of Windows and does not require a driver to be pre-installed.



### Configuration File

The application reads the "PowerPmacNC.ini" configuration file in its exe directory at start-up to obtain its configuration data. A *Reference* copy of this file is included in the project for convenience. Copy "Reference PowerPmacNC.ini" to the exe directory (usually "bin\Debug"), rename it to "PowerPmacNC.ini", and edit it to specify your machine type, axis definitions, units, and other important parameters. The configuration file is well commented for convenience.

```
[Machine Constructor]
; TODO: Specify from one to ten axis labels separated by commas.
; Axis labels can be more than one character but must be short. Suggest two characters max.
AxisLabels=X,Y,Z,A,B

; TODO: Specify motor numbers separated by commas (for status monitoring).
; The first motor number will be used to monitor the status of the first axis, etc.
MotorNumbers=1,2,3,4,5

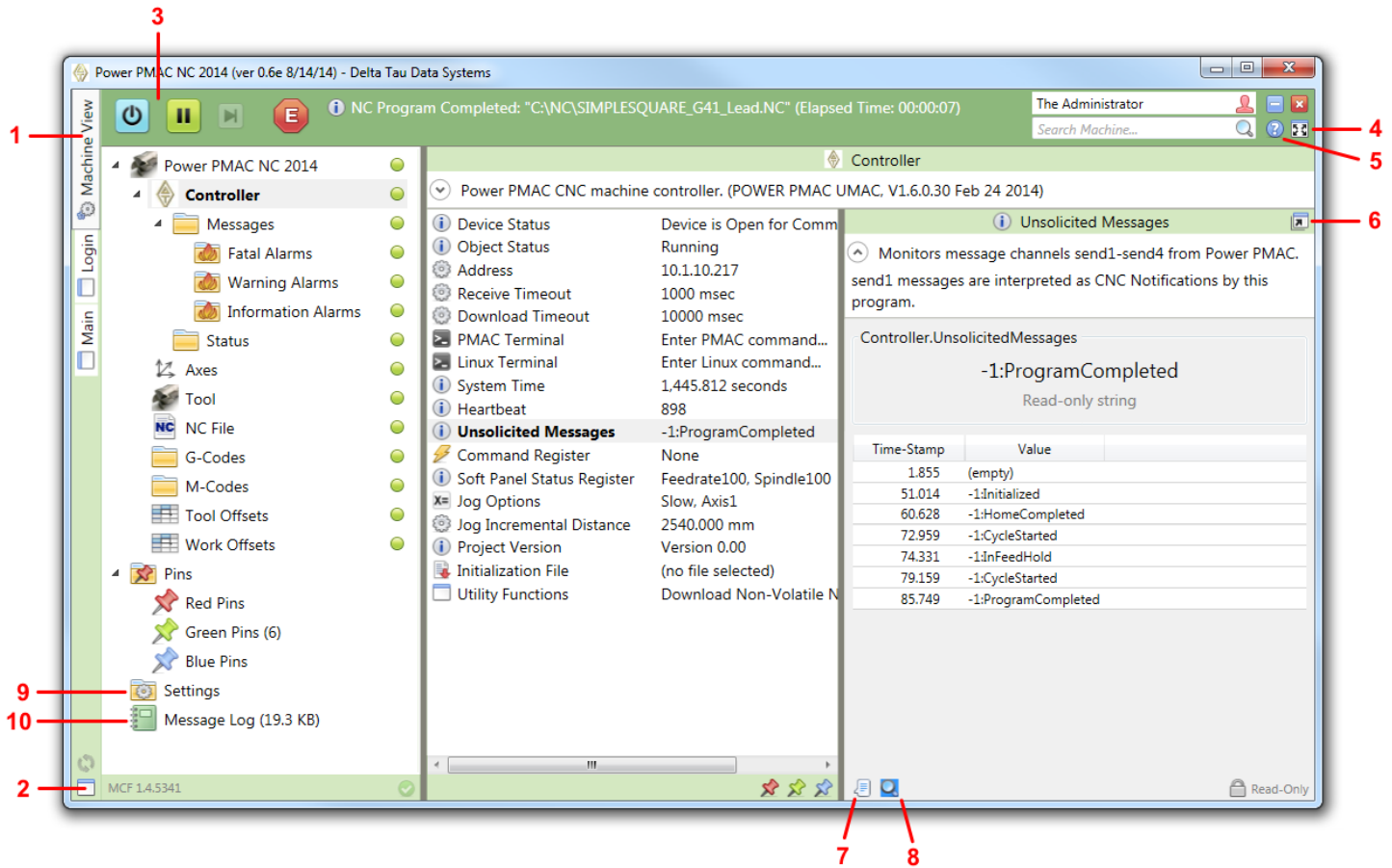
; TODO: Specify native length units (INCH or MM) and decimal places of precision (0-6).
NativeLengthUnits=MM
NativeLengthDecimalPlaces=3

; TODO: Select the controller (PowerPmacController, TurboPmacController or MockController)
Controller=PowerPmacController
```










## Machine View

*Motion Commander Foundation* applications feature *Machine View*, an Explorer-style view of the data shared between the host PC program and the motion controller. Machine View also includes powerful tools for logging and plotting this data, controlling the runtime state of the application, editing settings, viewing the message log, etc.



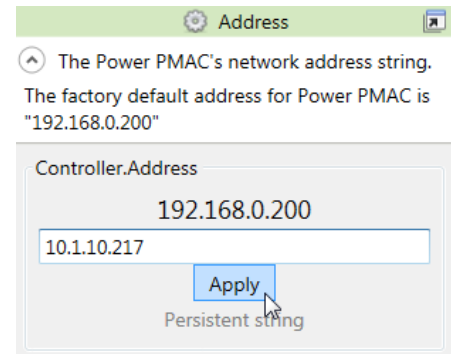
1. Navigate between Machine View and the User Interface.
2. Move the User Interface to a Separate Window.
3. Open/Close Communications, control the update cycle.
4. Toggle Full Screen mode.
5. Open the About Box for version information.
6. Open the member page in a Separate Window.
7. Log changes in value to the Message Log.
8. Open the time-stamped list of changes in value.
9. Edit the program Settings.
10. View the Message Log.


 Logging changes in value (7) of multiple variables allows the technician to view the interplay of these variables as the machine is operated. This is a very powerful diagnostic tool.

Message Log			
	8/16/2014 12:13:26 PM	205.060	Controller.CommandRegister=CycleStart (120)
	8/16/2014 12:13:26 PM	205.064	Controller.UnsolicitedMessages=-1:CycleStarted
	8/16/2014 12:13:26 PM		NC Program Started: "C:\NC\SIMPLESQUARE_G41_Lead.NC"
	8/16/2014 12:13:26 PM	205.090	Controller.CommandRegister=None (0)
	8/16/2014 12:13:34 PM	213.070	Controller.UnsolicitedMessages=-1:ProgramCompleted
	8/16/2014 12:13:34 PM		NC Program Completed: "C:\NC\SIMPLESQUARE_G41_Lead.NC" (Elapsed Time: 00:00:08)

Start the Power PMAC NC program, log in as the Administrator, and navigate to Machine View via the tab along the left edge of the window.

Select the **Controller** object to specify the IP address of your Power PMAC, then click the "Go Online" and "Run" buttons to open communication.



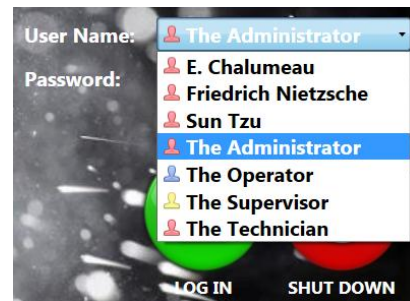
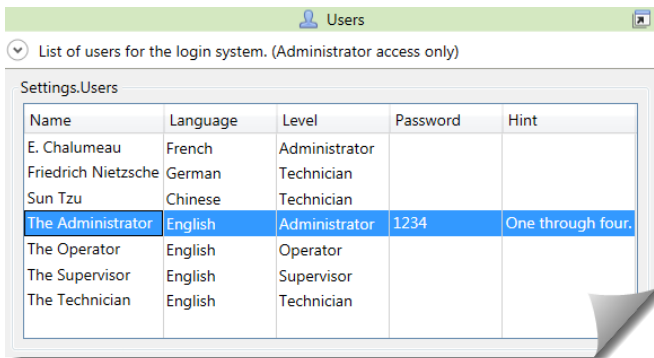
 Navigate to Machine View to change application **Settings** and manage the list of Users. In particular, you will most likely want to change the "Start Up" setting to "Go Online and Run at Login".

## Users

The user login system supports four access levels.

- Operator**      Access to the Operator screens but no access to Machine View
- Supervisor**    Access to the Operator screens and read-only access to Machine View
- Technician**    Unrestricted access to all UI pages and Machine View (except the list of Users)
- Administrator**    Unrestricted access including the list of Users

Log in as the Administrator and navigate to Machine View Settings to manage the list of Users.

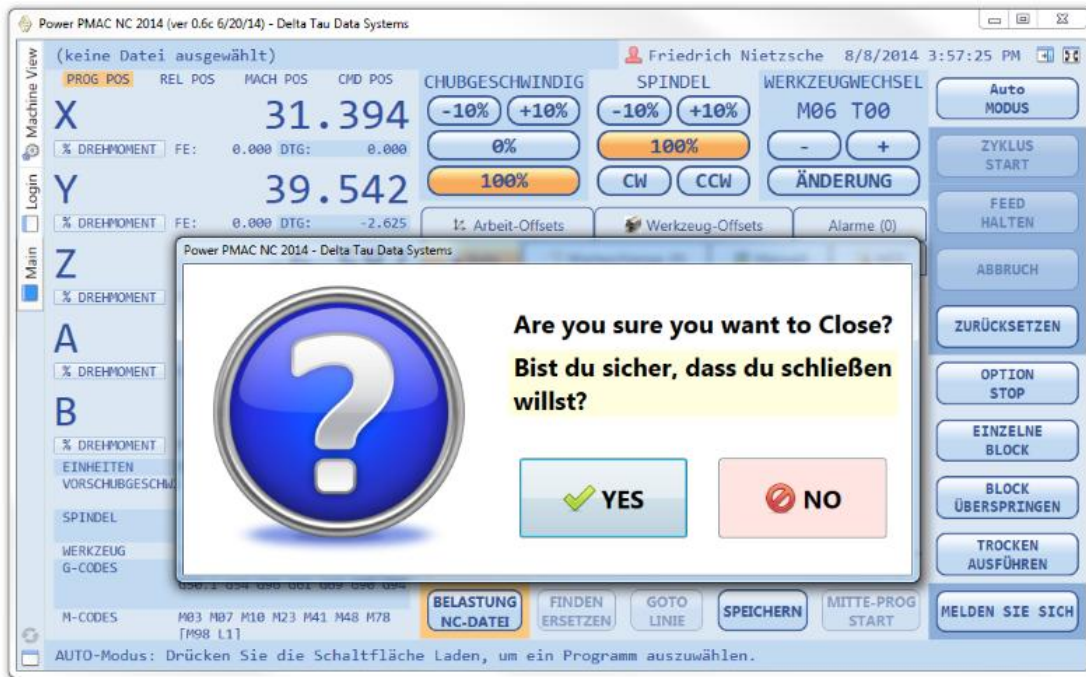


## Foreign Language Support

The Power PMAC NC program includes a sophisticated foreign language translation system. Each user's language is specified in his or her login profile. When a foreign language user logs in, Machine View text and selected UI Page text is displayed in the user's language. The "Language Support" setting provides runtime control of the language translation system.

Initial translations are obtained from the *Microsoft Translator* web service. Therefore, an internet connection will be required the first time that the foreign language user logs in. The translated text will be saved to a file named "*Languages\PowerPmacNC\_Language\_xx.txt*" where xx is one of 38 language codes. These files store native language strings and their foreign language translations, enabling the foreign language text to be edited by a human translator.

English:      Are you sure you want to Close?  
 German:      Bist du sicher, dass du schließen willst?




## Skins

The "Skin" setting may be used to change the color scheme of the main operator screen.

The "SkyBlue" and "MidnightBlue" skins are shown.



## Ctrl and Shift Keys

 If the **Ctrl** key is pressed while the **LOG IN** button is clicked then the user will be logged in but communications will not be opened. This feature is useful for working off line without a controller connected.

If the **Shift** key is pressed while the program is closed then the program will close immediately change the "Are you sure" dialog box or the splash screen. This feature saves time during development.

## NC Files

The Power PMAC NC program parses the selected NC file and generates a temporary file named "NcProgram.pmc" for downloading to the controller. The main NC program is enclosed in "open prog n/close" statements for loading into the PMAC program buffer. Any existing block numbers are stripped and a block number is prepended to each line for execution monitoring as shown in this example.

The NC file	The "NcProgram.pmc" download
1 <b>G21</b>	open prog 100
2 <b>G0 G17 G40 G49 G80 G90</b>	N1G21
3 <b>T1</b>	N2G00G17G40G49G80G90
4 <b>G0 G90 G54 X0 Y0</b>	N3T1
5 <b>G1 Z0. F1500</b>	N4G00G90G54X0Y0
6 <b>G41 X25 Y25</b>	N5G01Z0.F1500
7 <b>G1 X50</b>	N6G41X25Y25
8 <b>Y50</b>	N7G01X50
9 <b>X25</b>	N8Y50
10 <b>Y25</b>	N9X25
11 <b>G40 X0 Y0</b>	N10Y25
12 <b>G00 X-5 Y-5</b>	N11G40X0Y0
13 <b>M30</b>	N12G00X-5Y-5
	M30
	close

## The NC File Parser

- Wraps the NC in "open prog n/close" for downloading
- Prepends block numbers for execution tracking
- Strips comments (NC-style and C-style) and normalizes line endings
- Detects volatile subprogram calls and includes them in the download
- Tracks volatile subprogram file time-stamps and re-downloads if modified
- Detects nonvolatile subprogram calls and uploads them from the controller
- Throws an error if any called subprogram is not found
- Detects native PMAC commands and expressions and passes them unmodified
- Tracks modal G and M-codes and D/F/H/S/T values for Mid-Program Starts
- Supports the Block Skip option
- Supports Fixed Cycles
- Supports Line and File Prepend and Append features
- Warns if closing M30 or M99 is absent

## NC File Configuration

The application's "PowerPmacNC.ini" configuration file includes a section for NC files. These are the default values:

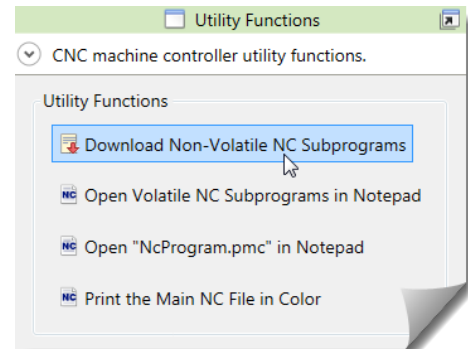
```
[NC Files]
; Main and MDI program numbers are absolute.
; Subprogram numbers start at the specified base address.
MainProgramNumber=100
MdiProgramNumber=99
CoordinateSystem=1
SubprogramFolder="C:\NC"
; To disallow subprograms, set range to (0,-1)
NonvolatileSubprogramMin=0
NonvolatileSubprogramMax=99
VolatileSubprogramMin=100
VolatileSubprogramMax=199
SubprogramBaseAddress=5000
```

## Subprograms

Subprograms can be called by 'O' program number or by path as shown in this example. Subprograms may call other subprograms with the one restriction that *nonvolatile* subprograms may only call other *nonvolatile* subprograms. "Volatile" subprograms exist in NC files and are downloaded along with the main program. "Nonvolatile" subprograms exist on the controller and must be downloaded via a utility included in the Power PMAC NC program and saved.

(Main Program) ... M98 P0101 M98 P0002 L5 // repeat count M98 (C:\myfolder\myname.nc) ... M30  O0101 (subprogram #101) ... M99  O0102 (subprogram #102) ... M99	(C:\NC\O0103.nc) O0103 (subprogram #103) ... M99  (C:\myfolder\myname.nc) O0104 (subprogram #104) ... M99	(Nonvolatile Subprogram File) O0001 (subprogram #001) ... M99  O0002 (subprogram #002) ... M99
---	---	---

"Volatile" subprograms are always downloaded together with the main program via a temporary file named "NcPrograms.pmc". "Nonvolatile" subprograms are downloaded via a utility included in the Power PMAC NC program. An NC file may contain more than one nonvolatile subprogram. The *Power PMAC IDE* must be used to SAVE after nonvolatile subprograms are downloaded. Volatile and nonvolatile subprogram 'O' numbers must be within the ranges specified in the configuration file. These ranges may be set to (0, -1) to disallow subprograms of either type.



Volatile subprograms called by path may reside in any folder under any filename, and their 'O' program numbers must be specified as the first line of the NC program. Volatile subprograms called by 'O' program number may reside in the main program's NC file (after the closing M30) or in the Subprogram Folder specified in the configuration file. Volatile subprogram NC files in the Subprogram Folder must be named "Onnnn.nc" where "nnnn" is the 'O' program number.

All subprograms will be loaded into PMAC buffers at the specified Subprogram Base Address plus 'O' program number.

## Native PMAC Commands and Expressions

The NC file parser detects native PMAC *command lines* and passes them unmodified. Tokens that identify a line as a native PMAC command are "if, else, while, do, goto" or the presence of an equals sign '=' or curly bracket '{}'. In addition, lines with a pipe '|' as the first character will be treated as native PMAC commands. The pipe will be stripped.

```
N100:
P103 = P24 + cos(P102)*P18
do
{
  G01 Y0.01 F2500
  | P779++
}
while (P779 <= P780)
if (P101 > P11) goto 100
```

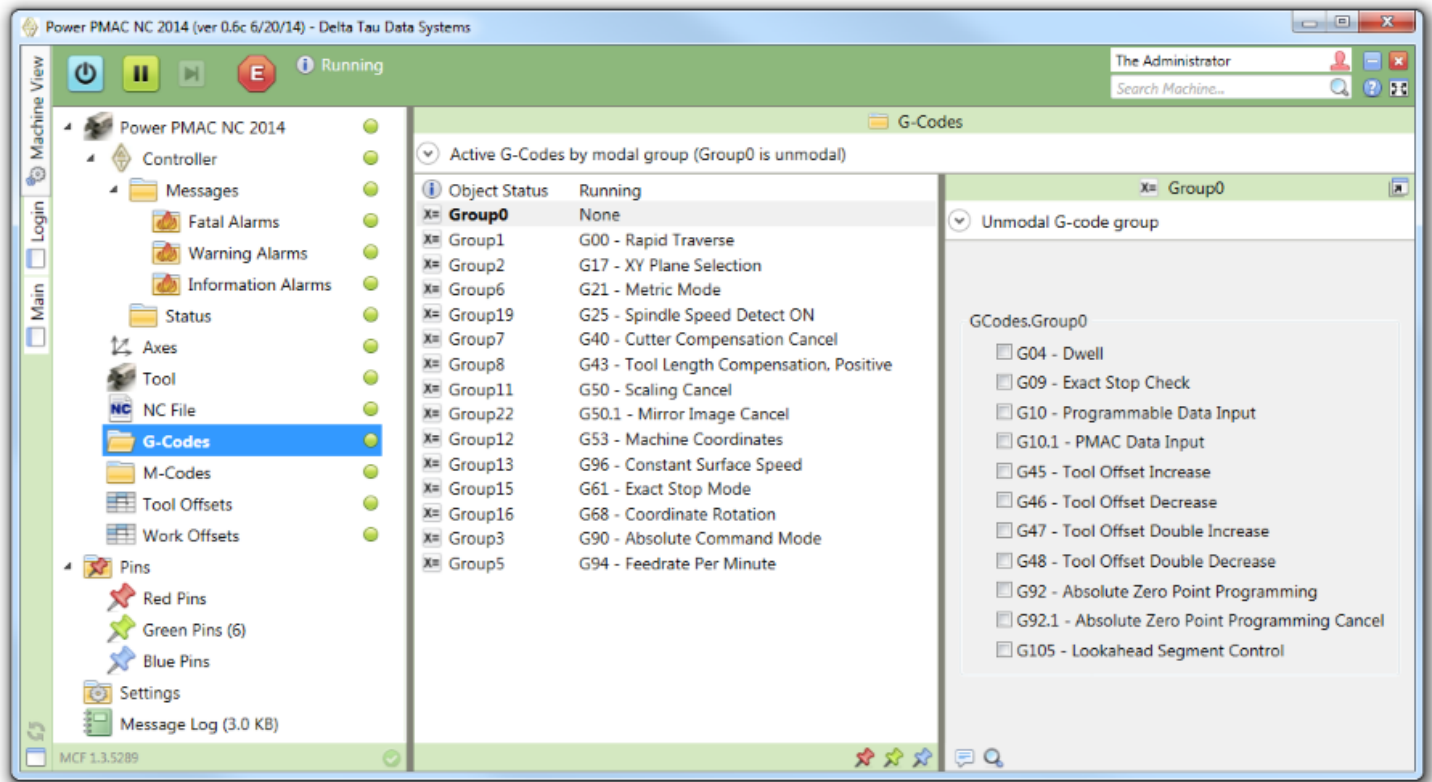
PMAC jump labels such as "N100:" will pass, while NC-style block numbers such as "N100" will be stripped.

Native PMAC *expressions* can be enclosed in square brackets within a line of NC as shown in this example:

```
G54.1 P[-1+Q511*2] (1ST PART OFFSET)
X[cos(P30)] Y[P32]
```

## G and M-Code Groups

The Power PMAC NC program includes support for the common G and M-code groups as shown in Machine View. Group 0 is "unmodal" and the rest are "modal" (the codes in the group are mutually exclusive). Group 6 switches the application between English (INCH) and Metric (MM) modes.



Each G and M-code group is linked to a PMAC variable specified in "DeviceMembers.xml".

```
<Member Name="GCodes.Group0" Getter="P200" TimeBetweenUpdates="500" />
<Member Name="GCodes.Group1" Getter="P201" TimeBetweenUpdates="500" />
<Member Name="GCodes.Group2" Getter="P202" TimeBetweenUpdates="500" />
<Member Name="GCodes.Group6" Getter="P206" TimeBetweenUpdates="500" /> ...
<Member Name="MCodes.SubprogramGroup" Getter="P308" TimeBetweenUpdates="500" />
```

Refer to "GCodes.cs" and "MCodes.cs" in the *PowerPmacNC* project for the numerical values of each code. For example, if P206=1 then G21 of Group 6 is active.

```
/// <summary>G-code Group 6 enumeration.</summary>
public enum EGroup6
{
    /// <summary>Inch Mode</summary>
    [Description("G20 - Inch Mode")]
    G20,
    /// <summary>Metric Mode</summary>
}
```

```

    [Description("G21 - Metric Mode")]
    G21,
};

```

Refer to "CustomCodeGroups.cs" in the *CustomExamples* project to learn how to add custom G and M-code groups to the Power PMAC NC program. Conversely, G and M-code groups that are not required by the application may be removed by specifying them in the application's "PowerPmacNC.ini" configuration file. (Group0, Group6, ProgramGroup and SubprogramGroup are used by the program and may not be removed.)

```

[Machine Constructor]
ExtraneousGroups=Group11, Group22, ThreadingGroup, GearRangeGroup, BAxisGroup

```

## Mid-Program Start

The line of NC where the cursor is located will become the *Mid-Program Start line* when the "MID-PROG START" button is pressed. Pressing the button a second time clears the Mid-Program Start function.

When the CYCLE START button is pressed, the NC file will be scanned down to the Mid-Program Start line and a block of NC will be generated which sets all of the modal G and M-codes and D/F/H/S/T values encountered in the scan.

Mid-Program Starts from within subprograms are not supported at this time.

Mid-Program Start at line 13

```

1 %
2 (Stress Test Main1.nc)
3 G21
4 G0 G17 G40 G49 G80 G90
5 T1 M6
6 G0 G90 G54 X27.179703 Y42.16696 (S40000 M3)
7 G43 Z25.
8 Z5.
9 G1 Z-4.511401 F5000
10 G04 P4
11 Y42.16696
12 X28.069259 Z-4.997955 F5000
13 X34.924836 Z-7.966783
14 X34.762704 Z-7.910719
15 G04 P3
16 X33.512909 Z-7.445079
17 X33.172713 Z-7.319875

```

The "NcProgram.pmc" download

```

open prog 100
G21G01G17G40G43G90G54F5000T1M06
N13X34.924836Z-7.966783
N14X34.762704Z-7.910719
N15G04P3
N16X33.512909Z-7.445079
N17X33.172713Z-7.319875
...
M30
close

```

## Fixed Cycles

Fixed Cycles ("Canned Cycles") supported include G73, G74, G76, and G81 through G89. G80 cancels the fixed cycle function, as shown in the following example:

The NC File

```

3 G99 G81 X29.726016 Y-5.826994 R2.5 Z-17.5 F4000
4 X31.393792 Y-6.587248
5 X33.072664 Y-7.281055
6 X34.762704 Y-7.910719
7 G80 X47.062818 Y-10.726689
8

```

The Resulting Parser Output

```

N3G99G81X29.726016Y-5.826994R2.5Z-17.5F4000
N4G99G81X31.393792Y-6.587248R2.5Z-17.5F4000
N5G99G81X33.072664Y-7.281055R2.5Z-17.5F4000
N6G99G81X34.762704Y-7.910719R2.5Z-17.5F4000
N7G80X47.062818Y-10.726689

```

## Using M99 to Repeat the Main Program

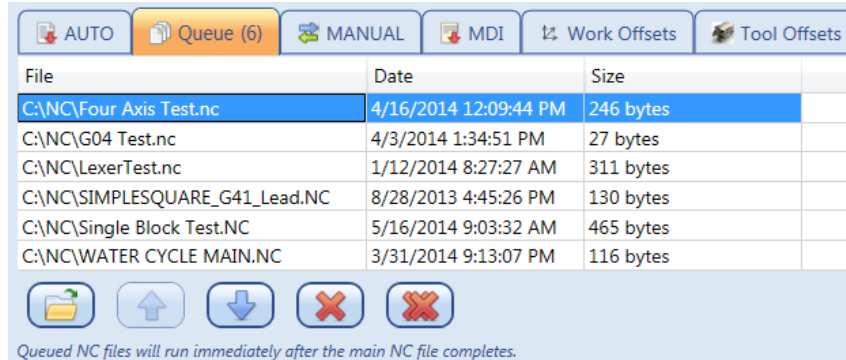
The main NC program will be repeated in an infinite loop if it is ended with M99 instead of the usual M30. A repeat count may be specified using the 'L' parameter. "M99 L5" will cause the main NC program to be repeated five times.

## The NC Program Queue

The first NC file loaded will always appear in the editor. Additional NC files will be loaded into the *Queue* as shown. The Queue includes tools for re-ordering and deleting files.



Drag-and-drop from *Windows Explorer* may be used to load multiple NC files.



File	Date	Size
C:\NC\Four Axis Test.nc	4/16/2014 12:09:44 PM	246 bytes
C:\NC\G04 Test.nc	4/3/2014 1:34:51 PM	27 bytes
C:\NC\LexerTest.nc	1/12/2014 8:27:27 AM	311 bytes
C:\NC\SIMPLESQUARE_G41_Lead.NC	8/28/2013 4:45:26 PM	130 bytes
C:\NC\Single Block Test.NC	5/16/2014 9:03:32 AM	465 bytes
C:\NC\WATER CYCLE MAIN.NC	3/31/2014 9:13:07 PM	116 bytes

Queued NC files will run immediately after the main NC file completes.



Queued NC files will be downloaded and executed immediately after the initial program completes without the need for further CYCLE START commands.

## NC File Comments

Comments (displayed in green in the editor) are stripped before the NC program is downloaded. NC-style () comments, C-style /\*\*/ comments and CPP-style // comments are supported, as shown in the example.

C-style /\*\*/ comments may not span multiple lines.

Subprogram paths are also surrounded by parentheses in M98 calls as shown in the example.

```
12 X34.924836 Z-7.966783
13 M98 P0109 // CPP style comment
14 X33.072664 /*C style comment*/ Z-7.281055
15 M98 P0109 (NC style comment)
16 X31.393792 Z-6.587248
17 M98 (C:\NC\00105.nc)
18 X29.726016 Z-5.826994
```

## NC File Size Limitations


Informal testing indicates that NC files larger than around 100 MB will cause the text editor to run out of memory and crash the program.



## Customizing the Application

There are three ways that the Power PMAC NC program can be customized to suit a particular machine.

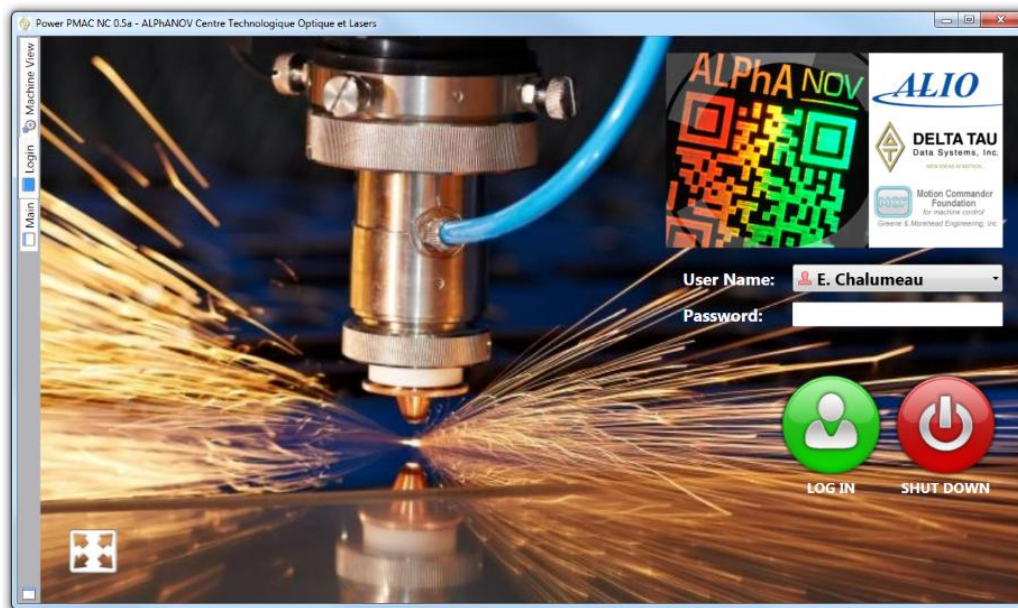
1. *Configure* the application by editing "PowerPmacNC.ini" and "Messages.xml".
2. *Extend* the application by creating external assemblies (plugin DLL's).
3. *Modify* the application by editing its source code.

 The configuration file and external assemblies are the preferred methods for customization because you'll be able to upgrade the main application without the need to merge your source code changes back in. If you do need to modify the main application, you'll want to clearly mark your edits with block-start and block-end comments to make merges easier in the future.

## Private Labeling

You may add the following lines to your "PowerPmacNC.ini" file to private-label the program and specify your own splash image and login screen background. Images should be PNG or JPEG format and must be located in the exe directory. The splash image should be approximately 500x300 pixels and the login image should be around 1000x700.

```
[Private Label]
CompanyName="ALPhANOV Centre Technologique Optique et Lasers"
SplashImage="LaserSplashImage.png"
LoginImage="LaserLoginImage.jpg"
```

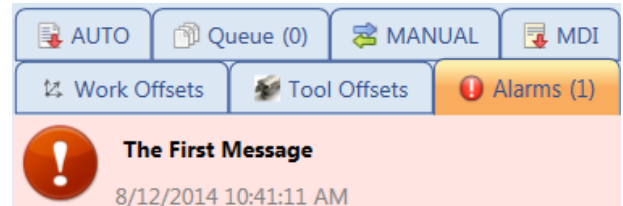


## Messages.xml

The Power PMAC NC program includes three bitwise message registers (Fatal, Warning and Information) that the controller can use to display messages to the operator. The message strings are specified in the "Messages.xml" file in the exe directory. A *Reference* copy of this file is included in the project for convenience.

```
<FatalMessages>
  <!-- Alarms activate on the rising edge of the bits and deactivate on the falling edge. -->
  <Message Bit="0" Description="The First Message">The is the first fatal message.</Message>
  <Message Bit="1" Description="The Second Message">The is the second fatal message.</Message>
</FatalMessages>
```

"Fatal" messages are marked with a red icon and are flashed in the status bar, while "Warning" and "Information" messages are marked with yellow or blue icons and are highlighted in the status bar.



Each message register is linked to a PMAC variable specified in "DeviceMembers.xml".

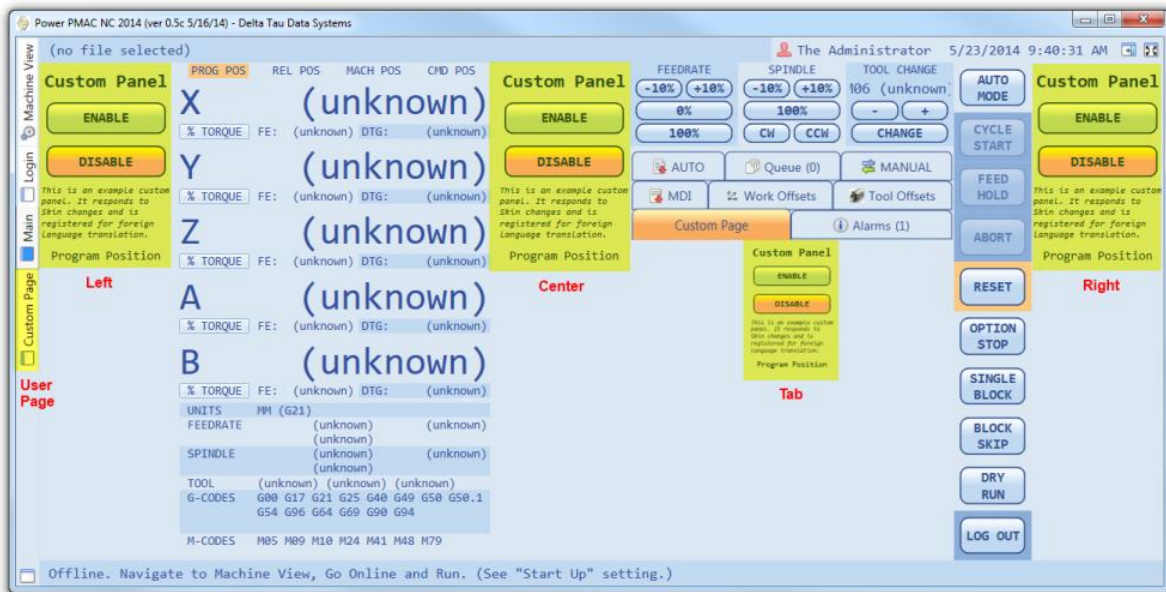
```
<Member Name="Controller.Messages.FatalMessages" Getter="M100" TimeBetweenUpdates="500" />
<Member Name="Controller.Messages.WarningMessages" Getter="M120" TimeBetweenUpdates="500" />
<Member Name="Controller.Messages.InformationMessages" Getter="M140" TimeBetweenUpdates="500" />
```

## External Assemblies

A C#/WPF class library project called "CustomExamples" is included with the SDK as a template for your custom plugin DLL's. The source files are well commented for convenience. A custom object adds its own device members to the Machine View hierarchy and also attaches "Changed" handlers to members created by the main program. A custom WPF Panel responds to UI Skin changes and is registered for foreign language translation. Simply add the following lines to your "PowerPmacNC.ini" file to test the custom plugin DLL.

```
[External Assemblies]
Object="..\..\..\CustomExamples\bin\Debug\CustomExamples.dll;CustomExamples.CustomObject"
CenterCustomFrame="..\..\..\CustomExamples\bin\Debug\CustomExamples.dll;CustomExamples.PageCustom"
CodeGroups="..\..\..\CustomExamples\bin\Debug\CustomExamples.dll;CustomExamples.CustomCodeGroups"
```

Your custom WPF Pages can be displayed by the main program in five different locations as shown in the illustration. Panels designed for the left, center and right columns should be tall and narrow, panels designed for the main screen tab area should be square, and panels designed to be full-screen *User Pages* may be much larger and more complex. Custom panels are hosted inside WPF *Viewboxes* so that they will be sized to fit the available screen area.



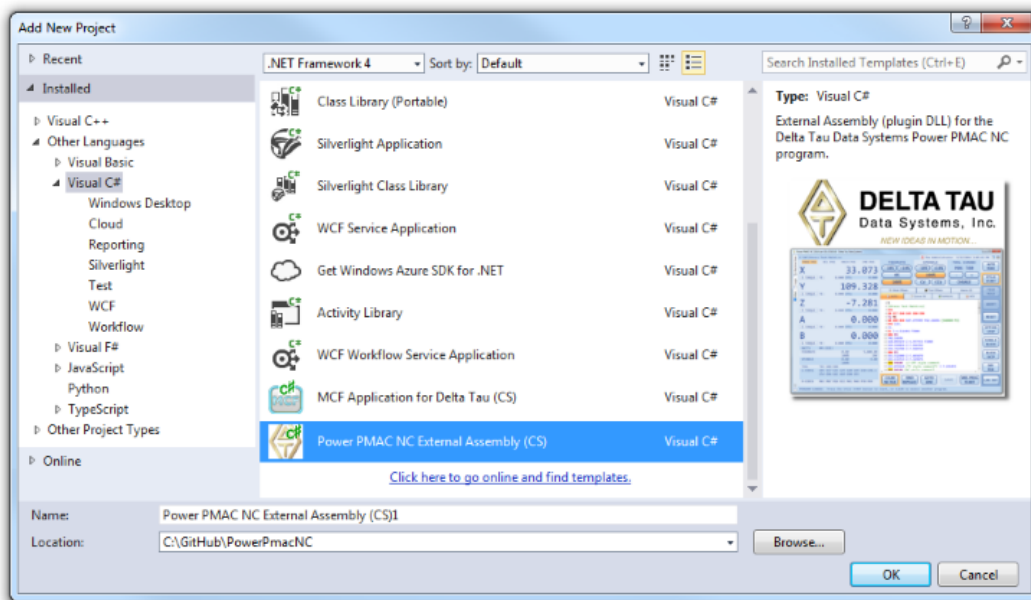
## The Visual Studio Project Template

A Visual Studio project template is provided that will add a *Power PMAC NC External Assembly* project to your solution. To use this template, simply put a copy of the included "Power PMAC NC External Assembly (CS).zip" file in your "Documents\Visual Studio 201x\Templates\ProjectTemplates\Visual C#" folder.

1. In Visual Studio, right-click on your "PowerPmacNC" solution and select "Add | New Project".
2. In the Visual C# templates, select "Power PMAC NC External Assembly (CS)".
3. Give it a name and start customizing!
4. Add your external assembly specification to your application's "PowerPmacNC.ini" file.

[External Assemblies]

CenterCustomFrame="mypath\MyExternalAssembly.dll;MyExternalAssembly.MyCustomPage"



## Appendix A. The Source Files

### GitHub\PowerPmacNcRelease

This folder will be copied to your PC when you “Clone” the repository from either the *GitHub* website or the *GitHub for Windows* application. You will want to use the *GitHub for Windows* application to periodically “Sync” to update your local copy to the most recently released version.



It is highly recommended that you make working copies of both the *PowerPmacNC* Visual Studio solution and the Power PMAC project in order to avoid losing your edits when you Sync. If a Sync fails for any reason, simply delete the entire “GitHub\PowerPmacNcRelease” folder and Clone again.

README.md, Banner.png, .gitattributes, .gitignore  
The Git repository configuration files. Do not edit.

GitHub\PowerPmacNcRelease\PowerPmacNC  
Source code for the Power PMAC NC C# application. (See detail below.)

GitHub\PowerPmacNcRelease\PMAC Source Code\PowerPMAC  
The Power PMAC project that works together with the host PC application to run NC files. The *Delta Tau Power PMAC IDE* will be used to download this project to the controller.

GitHub\PowerPmacNcRelease\PMAC Source Code\TurboPMAC  
The Turbo PMAC project that works together with the host PC application to run NC files. The *Delta Tau PWin32-PRO2 PMAC Executive Program* will be used to download this project to the controller.

GitHub\PowerPmacNcRelease\PowerPmacNC Demo Build  
A demonstration version of the Power PMAC NC program intended for marketing and training purposes.

GitHub\PowerPmacNcRelease\CustomExamples  
This example C# project produces an external (plugin) DLL that can be loaded by the Power PMAC NC program to extend its functionality with custom operator panels, variables, and G/M-code groups.

### GitHub\PowerPmacNcRelease\PowerPmacNC

Source code for the *Power PMAC NC* application. This is a .NET 4.0/C#/WPF application based on the *Motion Commander Foundation* .NET framework. Visual Studio 2010 or newer (Express or Professional) can be used to build this project.

PowerPmacNC.sln, PowerPmacNC.csproj, DeltaTau.ico  
The Visual Studio solution and project files, and the application icon.

Main.cs  
The entry point for MCF-based applications, including the *Machine* definition.

Enumerations.cs  
The enumerations used by the application.

PowerPmacController.cs, TurboPmacController.cs  
*Power PMAC* and *Turbo PMAC* CNC controller support classes.

DeviceMembers.xml  
The PMAC variable assignments and update rates for each device member in the *Machine* definition.

GCodes.cs, MCodes.cs

The standard G and M-code group definitions. Custom G and M-code groups should be added via external (plugin) DLL's instead of directly editing these files.

PageLogin.xaml, PageLogin.xaml.cs

The user login WPF page. Alternative background images may be specified in the application's configuration file.

PageMain.xaml, PageMain.xaml.cs

The main operator panel WPF page. Most of the process logic that operates the CNC machine is in the code-behind of this WPF page.

Reference PowerPmacNC.ini

The application reads the "PowerPmacNC.ini" configuration file in its exe directory at start-up to obtain its configuration data (machine type, axis definitions, units, and other important parameters). A *Reference* copy of this file is included in the project for convenience.

Reference PowerPmacNC\_Settings.xml

The application saves the values of all persistent variables in the "PowerPmacNC\_Settings.xml" file in its exe directory. A *Reference* copy of this file is included in the project for convenience.

Reference Messages.xml

The app includes three bitwise message registers (Fatal, Warning and Information) that the controller can use to display messages to the operator. The message strings are specified in the "Messages.xml" file in the exe directory. A *Reference* copy of this file is included in the project for convenience.

SecureDongle\_Control32.dll, SecureDongle\_Control64.dll

The hardware key (dongle) driver libraries for both 64-bit Windows and 32-bit Windows.

GitHub\PowerPmacNcRelease\PowerPmacNC\Properties

Standard C# project directory that contains assembly information.

GitHub\PowerPmacNcRelease\PowerPmacNC\References

This folder contains the MCF libraries and other DLL's required by the application.

GitHub\PowerPmacNcRelease\PowerPmacNC\SupportClasses

This folder contains C# and WPF support classes for the application.

GitHub\PowerPmacNcRelease\PowerPmacNC\Skins

This folder contains the user interface "Skins" WPF resource dictionaries.

GitHub\PowerPmacNcRelease\PowerPmacNC\Images

This folder contains image files for the application. Do not edit these images. Alternative background images may be specified in the application's configuration file.

GitHub\PowerPmacNcRelease\PowerPmacNC\Languages

This folder contains the foreign language files. MCF generates a language file the first time that a foreign language user logs in. This file may then be edited by a skilled human translator to refine the machine translations. For example, the German translation of FEEDRATE must be abbreviated to display correctly.

English: FEEDRATE

German: VORSCHUBGESCHWINDIGKEIT

## Appendix B. The Configuration File

```
;
; "PowerPmacNC.ini" - Configuration file for the Power PMAC NC 2014 program.
; This file is read by PowerPmacNC.exe at startup and must be in the exe directory.
; This file will NOT be overwritten by MCF and should be well commented.
;

[Machine Constructor]
; TODO: Specify the machine type (Mill or Laser)
MachineType=Mill

; TODO: Specify from one to ten axis labels separated by commas.
; Axis labels can be more than one character but they must be short. Suggest two characters max.
AxisLabels=X,Y,Z,A,B

; TODO: Specify motor numbers separated by commas (for status monitoring).
; The first motor number will be used to monitor the status of the first axis, etc.
MotorNumbers=1,2,3,4,5

; TODO: Specify the application's native length units (INCH or MM) and decimal places of precision (0-6).
NativeLengthUnits=INCH
NativeLengthDecimalPlaces=4

; TODO: Select the controller (PowerPmacController, TurboPmacController or MockController)
Controller=PowerPmacController

; TODO: Specify quantity of tool offsets (0 min, 25 max)
ToolOffsets=25

; TODO: Specify quantity of G54.1 work offsets (0 min, 48 max)
G541=25

; TODO: List G and M-code group names that are NOT required by the application (separated by commas).
; Note: Group0, Group6, ProgramGroup and SubprogramGroup may not be removed.
;ExtraneousGroups=Group11,Group22,ThreadingGroup,GearRangeGroup,BAxisGroup

[NC Files]
; Main and MDI program numbers are absolute.
MainProgramNumber=100
MdiProgramNumber=99
CoordinateSystem=1
SubprogramFolder="C:\NC"
; Subprogram numbers are relative to the specified Subprogram Base Address.
SubprogramBaseAddress=5000
; To disallow subprograms, set range to (0,-1)
NonvolatileSubprogramMin=0
NonvolatileSubprogramMax=99
VolatileSubprogramMin=100
VolatileSubprogramMax=199


[External Assemblies]
; TODO: Specify the assembly path and type name in quotes separated by a semicolon
; Object="path to DLL;FObject class name"
; UserPage="path to DLL;Page class name"
; CustomTab="path to DLL;Page class name"
; LeftCustomFrame="path to DLL;Page class name"
; CenterCustomFrame="path to DLL;Page class name"
; RightCustomFrame="path to DLL;Page class name"
; CodeGroups="path to DLL;class name"
; Example: Object="..\..\..\MySolution\MyProject\bin\Debug\MyProject.dll;MyNamespace.MyFObjectClass"
; Example: CenterCustomFrame="..\..\..\MySolution\MyProject\bin\Debug\MyProject.dll;MyNamespace.MyWpfPage"
; Example: CodeGroups="..\..\..\MySolution\MyProject\bin\Debug\MyProject.dll;MyNamespace.MyClass"

[Private Label]
; Optional private labeling. Images should be PNG or JPEG format and must be in the exe directory.
; Splash image should be around 500x300 pixels and login image should be around 1000x700.
;CompanyName="My Company Name"
;SplashImage="MySplashImage.png"
;LoginImage="MyLoginImage.jpg"
```

## Appendix B. Turbo PMAC Support

If your version of the Power PMAC NC program includes Turbo PMAC support then you may specify that controller in the application's "PowerPmacNC.ini" file.

```
[Machine Constructor]
; TODO: Select the controller (PowerPmacController, TurboPmacController or MockController)
Controller=TurboPmacController
```

 The Delta Tau Pro2 (*PcommServer*) communications library must be installed on the PC and properly configured. If your Turbo PMAC is not device 0 then the device number may be changed in the Power PMAC NC program by selecting the **Controller** object in *Machine View*.

### The Turbo PMAC Project

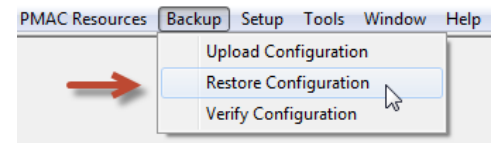
The "TurboPMAC\Configuration" folder contains a configuration file that can be downloaded to prepare your Turbo PMAC for use with the Power PMAC NC program.

"GitHub\PowerPmacNcRelease\PMAC Source Code\TurboPMAC\Configuration\TurboNcPlus\_Build.cfg"

Open this file in a text editor and uncomment the "tpnc\_umacdemobox.cfg" include line for a Demo Box test system or "tpnc\_virtualmotors.cfg" for a controller-only test setup running software-simulated motors.


```
//Comment in for UMAC Demo Box
//#Include "tpnc_umacdemobox.cfg"
//Comment in for virtual motors 1-8
//#Include "tpnc_virtualmotors.cfg"
```

Run the *PEWin32-PRO2 PMAC Executive Program* and use the Terminal window to issue a "\$\$\$\*\*" command to initialize the controller, then select the "Configure|M-variables" menu option and click the "Download Suggested M-variables" button. Now select the "Backup|Restore Configuration" menu option to download the "TurboNcPlus\_Build.cfg" configuration file.



Verify no warnings or errors in the Results window.

```
Device 0-> Total Warnings: 0
Device 0-> Total Errors: 0
Device 0-> END.
```

 After downloading the configuration, use the Terminal window to issue a "**SAVE**" command to copy to nonvolatile flash memory, then issue a "\$\$\$" command to reset the controller.

The Turbo PMAC controller is now ready to work with the Power PMAC NC program!

## Appendix C. Source Code Exclusions

### Included in the Source Code version:

- The WPF operator screens (Login, Main, and future screens) and associated logic (code-behind).
- The logic that defines the operation of the NC application.
- The device member definitions and G and M-code definitions.
- CS and Motor status monitoring.
- The NC file editor (AvalonEdit).
- Application skins support.
- Initialization file configuration support.
- The Mock Controller.
- Example External Assembly projects that demonstrates how to add custom panels and device members.

### NOT Included in the Source Code version:

- The MCF source code (Machine View, runtime engine, logging, alarming, foreign language support, MTConnect, etc)
- The low-level Power PMAC communications (sending commands, downloading files, etc)
- The NC file parsing (support for execution monitoring, subprograms, mid-tape start, etc)