



DELTA TAU

Power PMAC-NC 2016

Motion Commander Foundation
© 2016 Greene & Morehead Engineering, Inc.



.ini Configuration Manual

Power PMAC-NC16



Delta Tau Data Systems, Inc.

April, 2016

0018-E-01

COPYRIGHT INFORMATION

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

Software User Manual: © 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

Version	Date	Description
1.0	4/27/2016	Initial release.

Power PMAC-NC16™ - .ini Configuration Manual

Contents

.ini File Location	5
.ini Parameters	5
Machine Constructor	5
Axis Labels	6
MotorNumbers	6
Native Length Units.....	6
Native Length Decimal Places	7
Velocity Time Units	7
Controller Type	8
Tool Offsets	8
Work Offsets	8
Jog Speed Options.....	9
Extraneous Groups.....	10
Multiple Instances.....	10
MTConnectAgent	11
NC Files.....	12
Main/MDI Program Number	12
Subprogram Folder	12
Coordinate System	13
CutFolder.....	13
Macro Substitutions.....	13
Sub Program Base Number	14
Nonvolatile Subprogram Minimum/Maximum Number	14
Volatile Subprogram Minimum/Maximum Number	15
Parser Options.....	15
Fixed Cycles	15
Feed Rate (F) Command At The End	15
Allowed G-Codes and M-Codes	15
Allowed G/M Codes	16
External Assemblies	16
Center Custom Frame	16

Left Custom Frame.....	17
Right Custom Frame.....	18
Custom Tab	18
Private Label.....	19

PowerPmacNC.ini Configuration File

The Power Pmac-NC16 application reads the "PowerPmacNC.ini" configuration file at start-up to obtain its configuration data. The PowerPmacNC.ini is used to configure the HMI for number of axes, axis labels, number of work/tool offsets, etc. The .ini file also is used to activate external C# assemblies for custom tabs, panels, screens, and other customer created functionality.

.ini File Location

..GitHub/PowerPmacNC16-Runtime/Reference PowerPmacNC.ini

..GitHub/PowerPmacNc16-SDK/Reference PowerPmacNC.ini

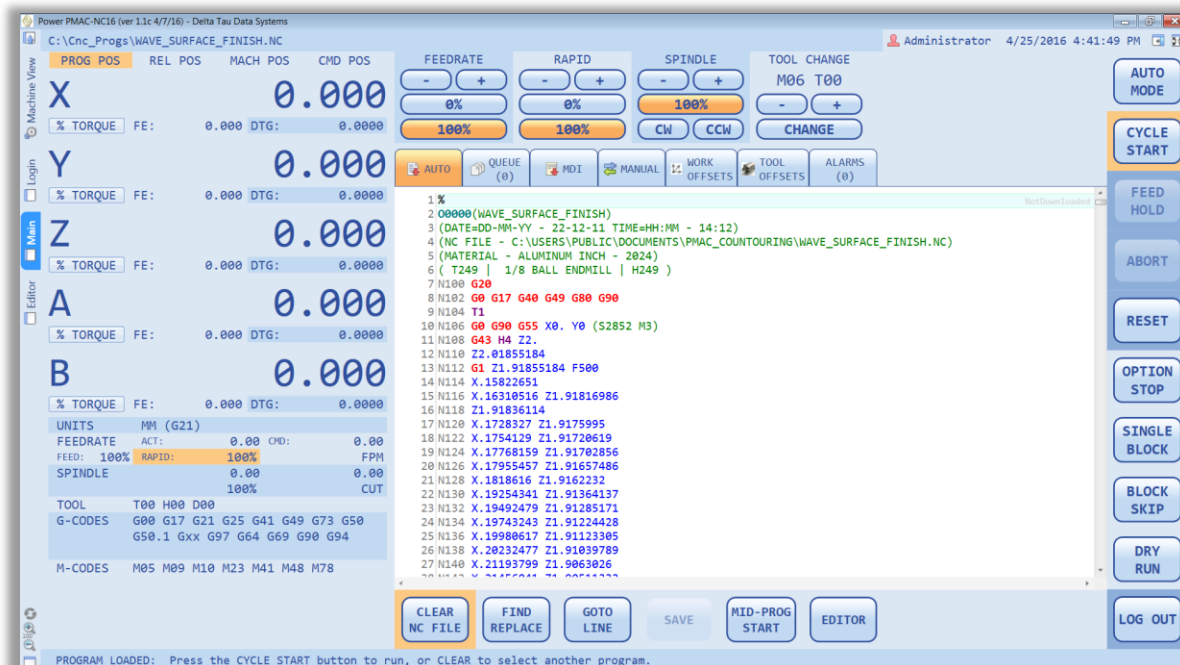
Locate the "Reference PowerPmacNC.ini" file located as shown above. Copy the file to the exe directory, rename it to "PowerPmacNC.ini", and edit it to specify your machine type, axis definitions, units, velocity units, and other important parameters.

.ini Parameters

Machine Constructor

```
; "PowerPmacNC.ini" - Configuration file for the Power PMAC-NC16 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 (Standard)  
MachineType=Standard
```

The current version allows for a *Standard* configuration. Future version will include various other types of machines to choose from. The Standard configuration enables a multi-motor (2-10) configuration as shown below.

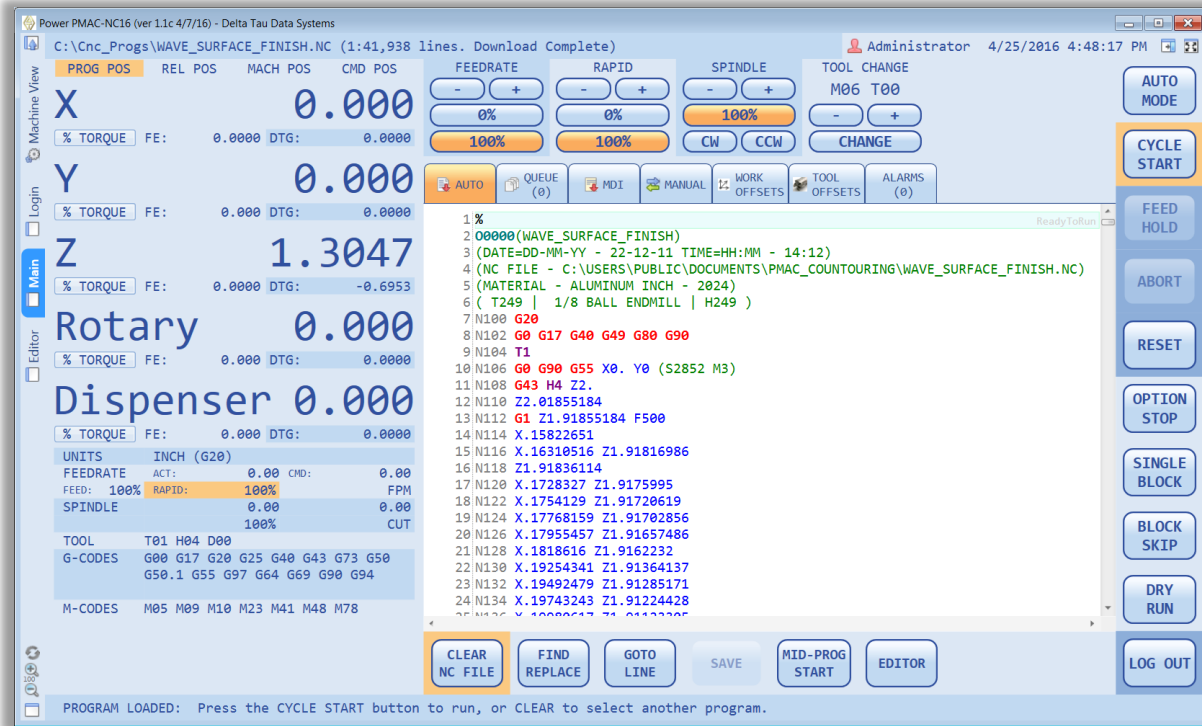


Axis Labels

```
; 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
```

This section configures the axis label displayed in the HMI. Any text is allowed and can include words such as “Rotary” or “Dispenser”.

Note: Each motor defined must have an axis label. If the number of axes does not match the number of labels the application will terminate in error.



MotorNumbers

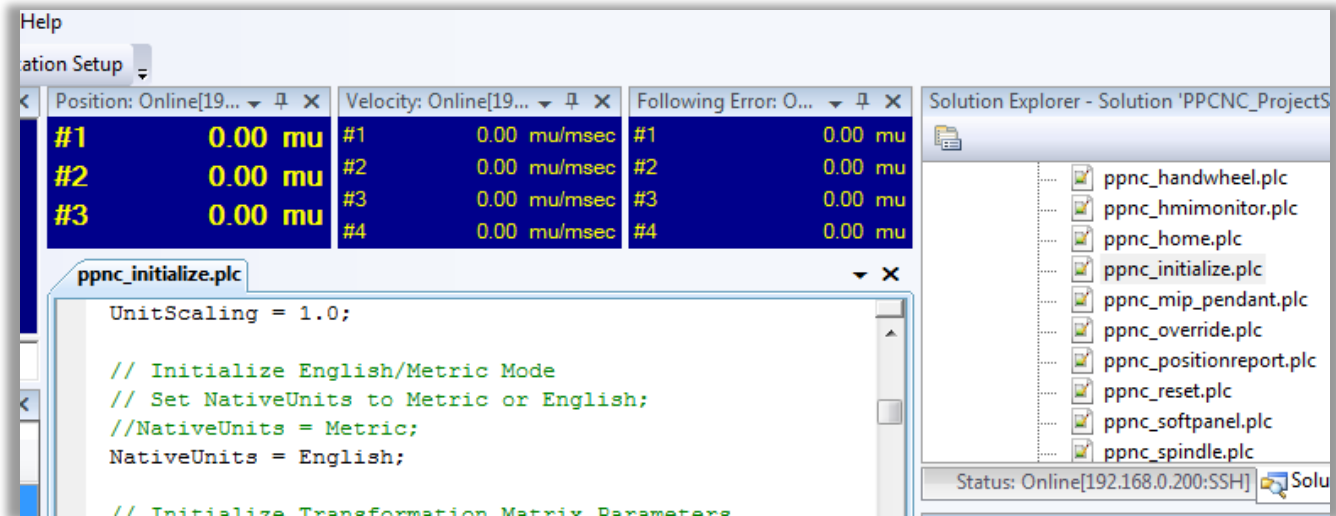
```
; TODO: Specify motor numbers separated by commas (for status  
monitoring).  
; The first motor number will be used to monitor the status of the first  
axis, etc.  
MotorNumbers=1,2,3,4,5
```

The *MotorNumbers* parameter defines not only how many axes are present, but which motor number corresponds to which axis. PMAC’s Sys.MaxMotors may need to be modified to match or exceed this setting (see ppnc_csparameters.pmh). This setting may not be in synchronous order depending on your particular PMAC motor setup.

Native Length Units

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

Set the native length units to either **INCH** (standard) or **MM** (metric). In order for the application to function correctly the *NativeUnits* parameter found in the ppnc_initialize.plc must match this setting.



Native Length Decimal Places

```

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

Set the number of decimal places you wish to display (6 places maximum).



Velocity Time Units

```

; TODO: Specify the time units to display in velocity labels (min, sec, etc)
VelocityTimeUnits=min
  
```

The velocity time units can be set to **min**, **sec**, **etc**. This is used for display purposes only and can be set to any value appropriate for the machine. The value will display exactly as input.

Controller Type

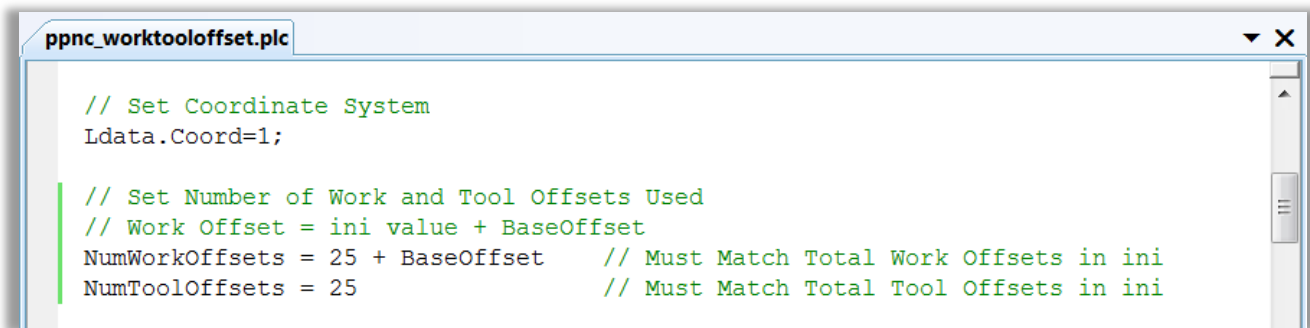
```
; TODO: Select the controller (PowerPmacController, TurboPmacController  
or MockController)  
Controller=PowerPmacController
```

The controller setting establishes the type of controller and what method of communications will be used. Settings include **PowerPmacController**, **TurboPmacController**, or **MockController**. When set to **MockController** the HMI may be used in a DEMO configuration, no actual functionality is enabled.

Tool Offsets

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

This section defines the number of tool offsets the HMI will display and manage. When this parameter is set to zero, the **TOOL OFFSETS** tab will disappear. The maximum value is 25. If this parameter has been modified from the default value of 25, the **NumToolOffsets** parameter in the **ppnc_workoffset.plc** must be modified to match as shown below.



```
ppnc_worktooloffset.plc  
  
// Set Coordinate System  
ldata.Coord=1;  
  
// Set Number of Work and Tool Offsets Used  
// Work Offset = ini value + BaseOffset  
NumWorkOffsets = 25 + BaseOffset    // Must Match Total Work Offsets in ini  
NumToolOffsets = 25                // Must Match Total Tool Offsets in ini
```

Work Offsets

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

This section defines the number of G54.1 Pxx work offsets the HMI will display and manage. If this parameter is set to zero, all additional G54.1 Pxx work offsets will disappear. Note the standard G54-G59 offsets will remain.

Offset	X	Y	Z
G54	39.918000	28.509000	-1.000000
G55	5.000000	36.903000	-44.369000
G56	5.000000	66.000000	-21.440140
G57	-7.320280	-18.787110	23.318470
G58	0.600000	-44.797980	3.000000
G59	0.000206	18.388783	-44.369000
G54.1 P1	1.000000	-5.217000	0.000000
G54.1 P2	6.986270	0.393700	0.694000
G54.1 P3	-2.374900	0.000000	0.000000
G54.1 P4	-20.920000	27.975000	20.942000
G54.1 P5	0.000000	0.000000	0.000000
G54.1 P6	0.000000	0.000000	0.000000
G54.1 P7	0.000000	21.240750	-0.843280
G54.1 P8	0.000000	0.000000	0.000000
G54.1 P9	0.000000	0.000000	0.000000

If this parameter has been modified from the default value of 25, the **NumWorkOffsets** parameter in the **ppnc_workoffset.plc** must be modified to match as shown below.

```

ppnc_worktooloffset.plc

// Set Coordinate System
ldata.Coord=1;

// Set Number of Work and Tool Offsets Used
// Work Offset = ini value + BaseOffset
NumWorkOffsets = 25 + BaseOffset    // Must Match Total Work Offsets in ini
NumToolOffsets = 25                // Must Match Total Tool Offsets in ini

```

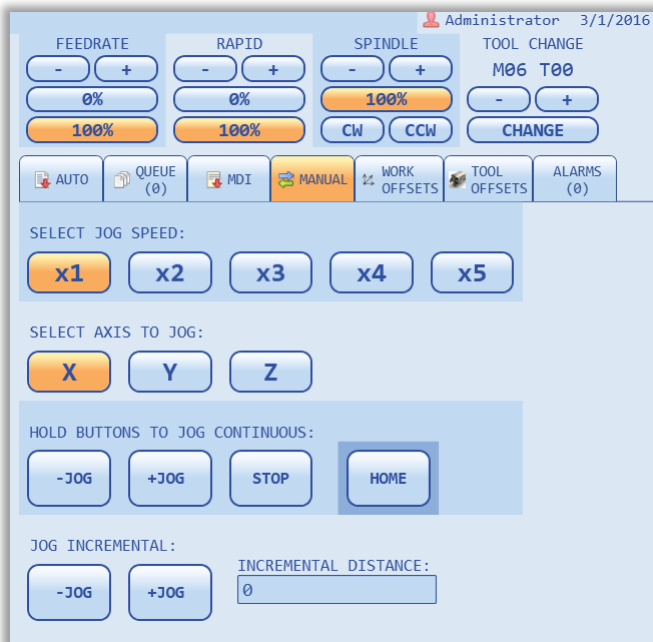
Jog Speed Options

```

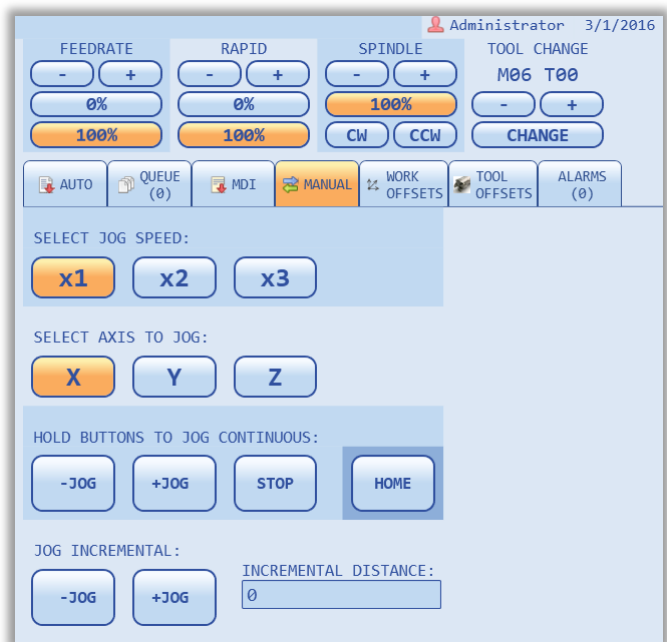
; TODO: Specify either three or five jog speed buttons to match the
pendant.
ThreeJogSpeeds=false

```

The **ThreeJogSpeeds** parameter defines how many jog speed selection buttons will be visible if the software MANUAL control panel is visible. This is intended to allow the coexistence of hardware pendants which in general either have three or five jog speed selections. This parameter can be set to true or false. The default value of false will display five jog speed buttons.



ThreeJogSpeeds=false

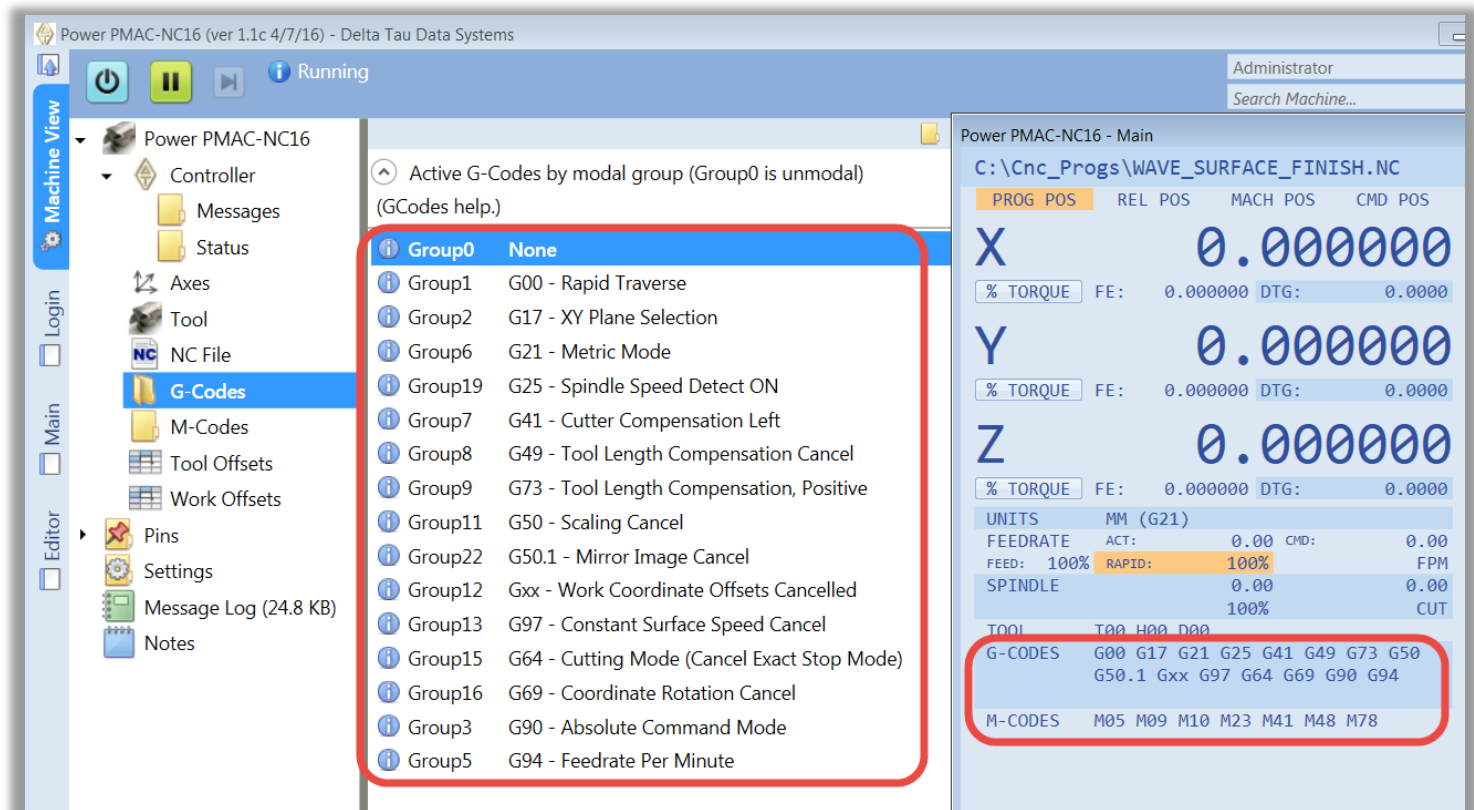


ThreeJogSpeeds=true

Extraneous Groups

```
; 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
```

The **ExtraneousGroups** parameter allows the integrator to exclude G-code groups from the display. The G-code groups are displayed both in Machine View and the Main screen as shown below. By default no groups are excluded and this parameter is commented.



The Following G-Code Groups may be Excluded:

Group1	Group11
Group2	Group12
Group3	Group13
Group5	Group15
Group7	Group16
Group8	Group19
Group9	Group22

The Following M-Code Groups may be Excluded:

SpindleGroup
CoolantGroup
ChuckGroup
ThreadingGroup
GearRangeGroup
FeedrateOverrideGroup
BAxisGroup

Multiple Instances

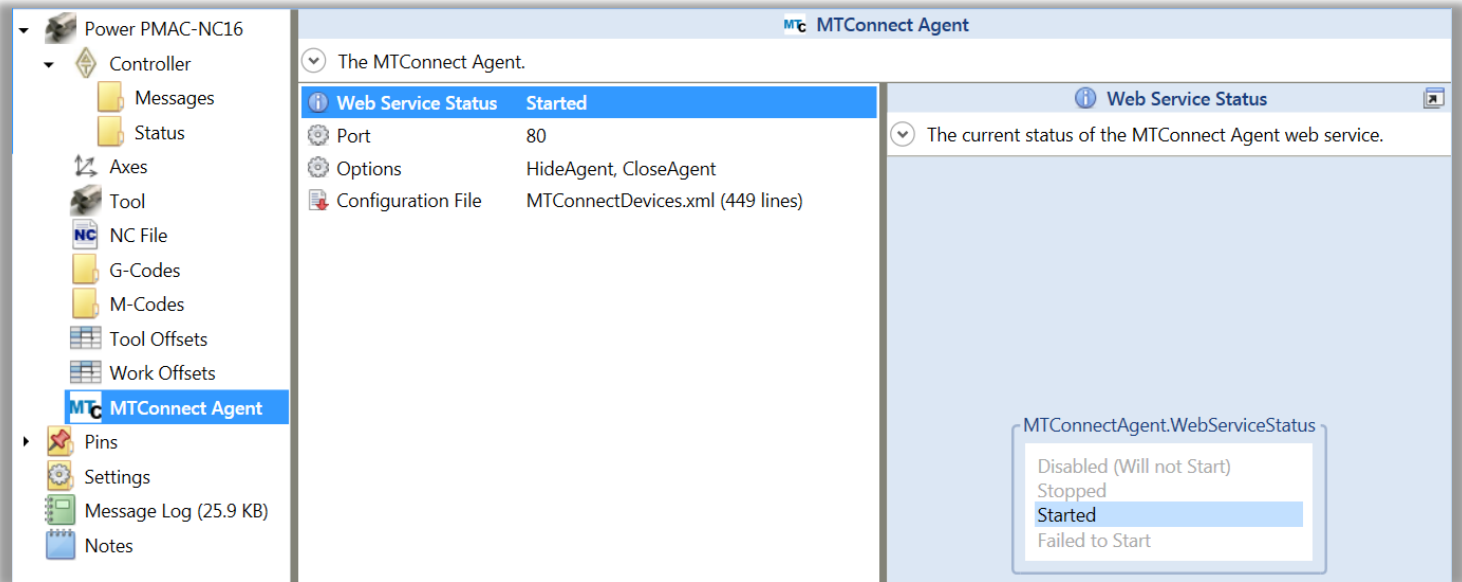
```
; TODO: Allow more than a single instance of the application to run.
AllowMultipleInstances=false
```

This section provides an option to run multiple instances of the Power PMAC-NC16 HMI on a single PC. This can be useful for special situations such as multiple coordinate system processing. Contact the factory for more information regarding this capability.

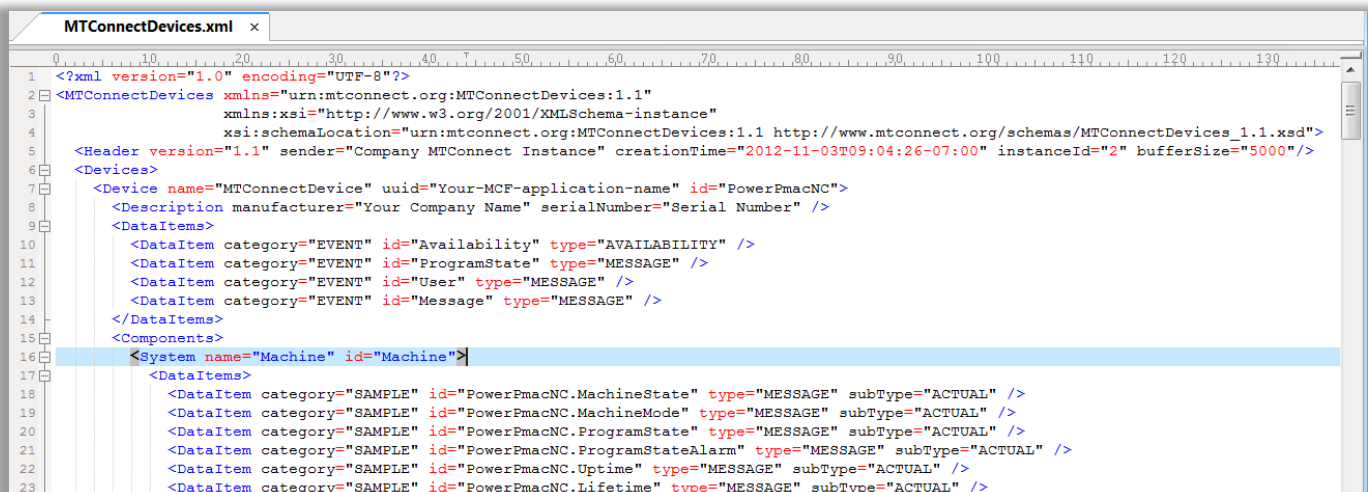
MTConnectAgent

```
; TODO: Add MTConnect agent support to the application.  
MTConnectAgent=false
```

This section allows users to activate the built in MTConnect agent. MTConnect is an open source standard which facilitates the organized retrieval of process information from numerically controlled machine tools. NC16 also will show a message that MTConnect agent is activated and running.



The MTConnect agent provides status, port, options, and configuration file settings. The Web Service Status reports if the MTConnect agent is disabled, stopped, started, or has failed to start. The Port is a 32-bit integer that will be used specifically by the agent to stream live information via HTTP protocol. As soon as the agent is activated you will see the **MTConnectDevices.xml** in the main NC16 folder. Refer to the Power PMAC-NC16 Software User Manual for complete details on how to configure this file.



NC Files

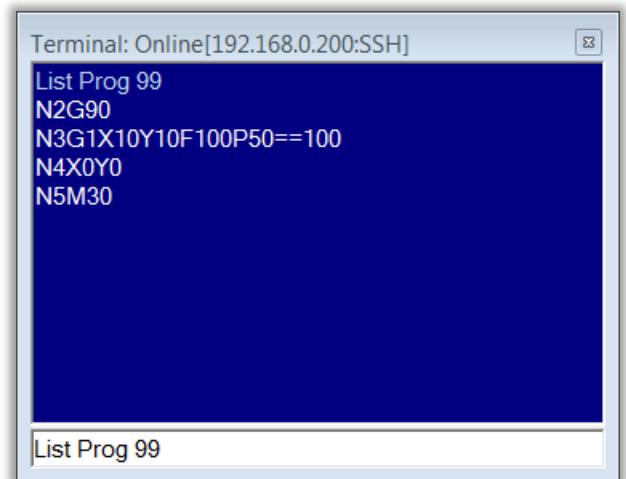
Main/MDI Program Number

```
[NC Files]
; Main and MDI program numbers are absolute.
MainProgramNumber=100
MdiProgramNumber=99
```

The **MainProgramNumber** parameter determines which program buffer the HMI will use for fixed buffer AUTO mode programs. In general this should not be modified but can be for special circumstances.

The **MdiProgramNumber** parameter determines which program buffer the HMI will use for MDI mode programs. In general this should not be modified but can be for special circumstances.

The user may reference the parsed program of either the Main or MDI programs for troubleshooting purposes directly in the IDE as shown below.



Subprogram Folder

```
[NC Files]
; Main and MDI program numbers are absolute.
MainProgramNumber=100
MdiProgramNumber=99
CoordinateSystem=1
SubprogramFolder="C:\NC"
```

The **SubprogramFolder** parameter specifies the folder location in which the HMI will automatically look for subprograms when the path is not explicitly defined in the NC part program. If the referenced subprogram does not exist in this folder the HMI will show an error when attempting to parse the subprogram.

Coordinate System

```
[NC Files]
; Main and MDI program numbers are absolute.
MainProgramNumber=100
MdiProgramNumber=99
CoordinateSystem=1
```

The **CoordinateSystem** parameter determines which PMAC coordinate system will be used for all aspects of the application. This will affect status monitoring, rotary buffer downloading, etc.

CutFolder

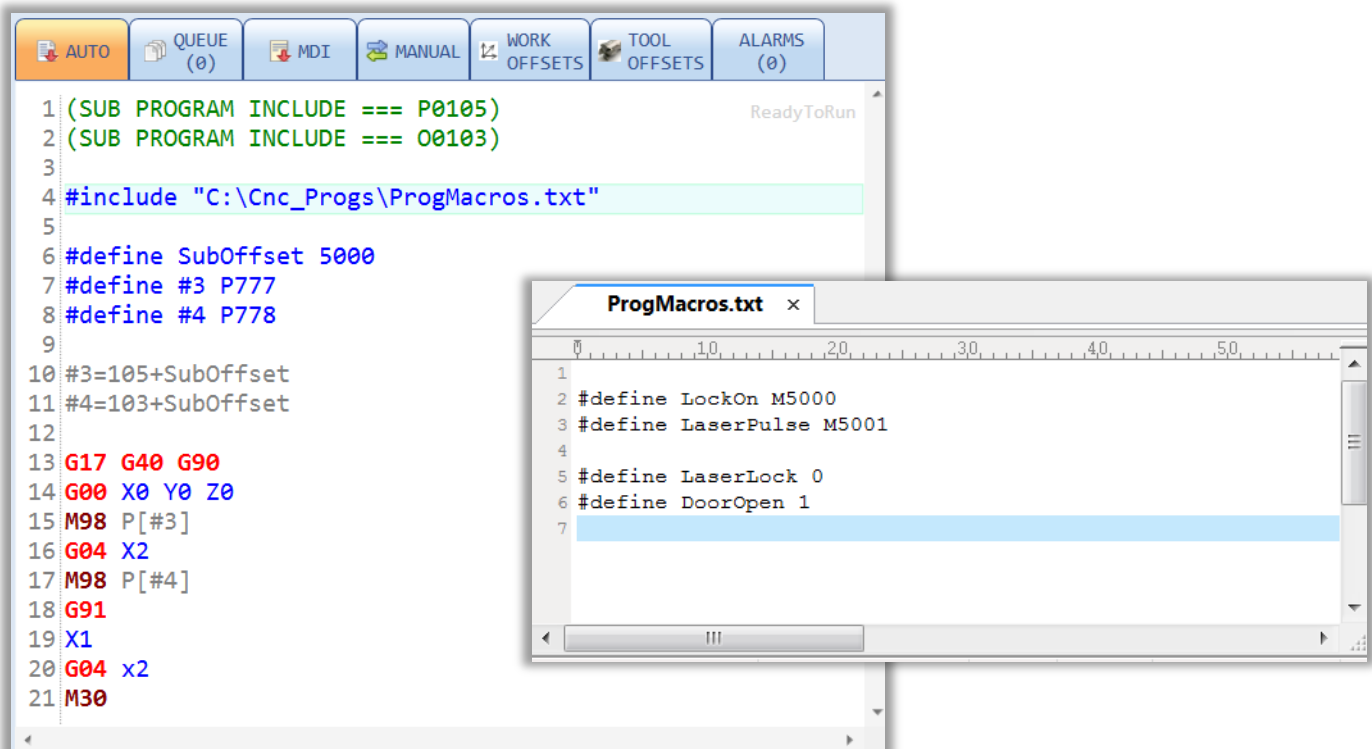
```
; NC files copied to the Cut Folder will automatically be queued and then
archived after they are run.
;CutFolder="C:\NC\Cut Folder"
```

When the **CutFolder** parameter is specified, NC part programs which are copied to this location will automatically be queued and run. Once the part program is complete it will be deleted and copied to the Archive sub-folder. This can be very useful for system automation and unattended operation.

Macro Substitutions

```
; Option to allow #define macro substitutions in NC files.
AllowMacros=true
```

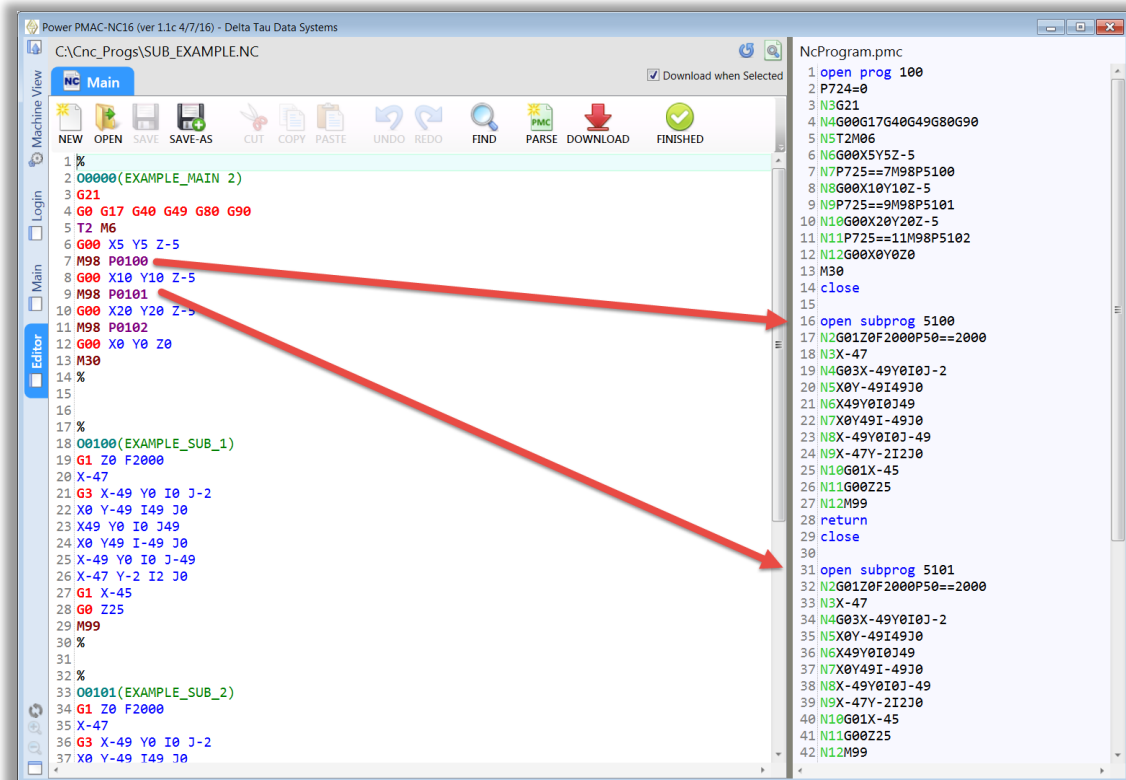
When enabled the **AllowMacros** parameter will allow the use of “Macro” style parametric programming in the HMI. Parametric style programming can be very helpful when coding NC part programs. In addition to embedded #define statements, the program can utilize the #include command directly in the NC part program to reference a file which includes the actual #define statements.



Sub Program Base Number

```
; Subprogram numbers are relative to the specified Subprogram Base Address.  
SubprogramBaseAddress=5000
```

The **SubprogramBaseAddress** defines the offset which will be added to the actual subprogram number when parsed. This is required by the control for real time tracking. For example if you are calling subprogram **P0104**, the HMI will parse the code into PMAC program buffer 5104 when the SubprogramBaseAddress is at the default value of 5000. The HMI's full screen editor includes a powerful code parser view where you can see exactly what is being downloaded to the control. Notice the subprogram values are added to the SubprogramBaseAddress and downloaded to the appropriate buffer.



Nonvolatile Subprogram Minimum/Maximum Number

```
; 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=699
```

The NonvolatileSubprogramMin/Max parameters define the range from which the HMI will allow non-volatile subprograms to be called. Non-volatile subprograms differ from volatile subprograms in that they are not included in the NC part program, or downloaded during program execution. They are pre-loaded and saved into the controller. These are usually created by the machine builder and often considered special features of the machine integration. These subprograms follow the same rules as the volatile subprograms and use the SubprogramBaseAddress in the same

way. If these parameters are set to zero, non-volatile subprograms will not be allowed. See the Power PMAC-NC16 User Manual for instructions on how to download non-volatile subprograms to the control. If the non-volatile subprograms are not downloaded using the special Utility Functions Tool they will not be able to be called.

Volatile Subprogram Minimum/Maximum Number

```
; 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
```

The VolatileSubprogramMin/Max parameters define the range from which the HMI will allow volatile subprograms to be called. Volatile subprograms differ from non-volatile subprograms in that they are loaded during the NC part program parse and are not retained in memory through a power cycle. If these parameters are set to zero, volatile subprograms will not be allowed. Volatile subprograms are typically called via an M98 command.

Parser Options

Fixed Cycles

```
[Parser Options]  
; Uncomment the desired NC parser options.  
;IgnoreFixedCycles=true
```

The **IgnoreFixedCycles** parameter allows the user to exclude fixed, or canned, cycles from the NC part program. This can be used to simplify the operation of special types of machines. If the parameter is set to true, the parser will ignore fixed cycle NC part Program lines.

Feed Rate (F) Command At The End

```
[Parser Options]  
; Uncomment the desired NC parser options.  
;IgnoreFixedCycles=true  
;FCodeAtEnd=true
```

The **FCodeAtEnd** parameter forces the parser to place any F code encountered at the end of the NC block. This is necessary for some applications. In the default state the parser will output the F code at the location encountered.

Allowed G-Codes and M-Codes

```
[Parser Options]  
; Uncomment the desired NC parser options.  
;IgnoreFixedCycles=true  
;FCodeAtEnd=true  
;OnlyAllowedGMCodes=true
```

The **OnlyAllowedGMCodes** parameter allows the integrator to specify which G&M codes outside the standard set are able to pass through the parser. Any G/M code outside the specified range will not be allowed and will cause an error message. This can be used by the integrator or machine builder to limit or restrict the functionality of the machine. The usage is explained in the following sections.

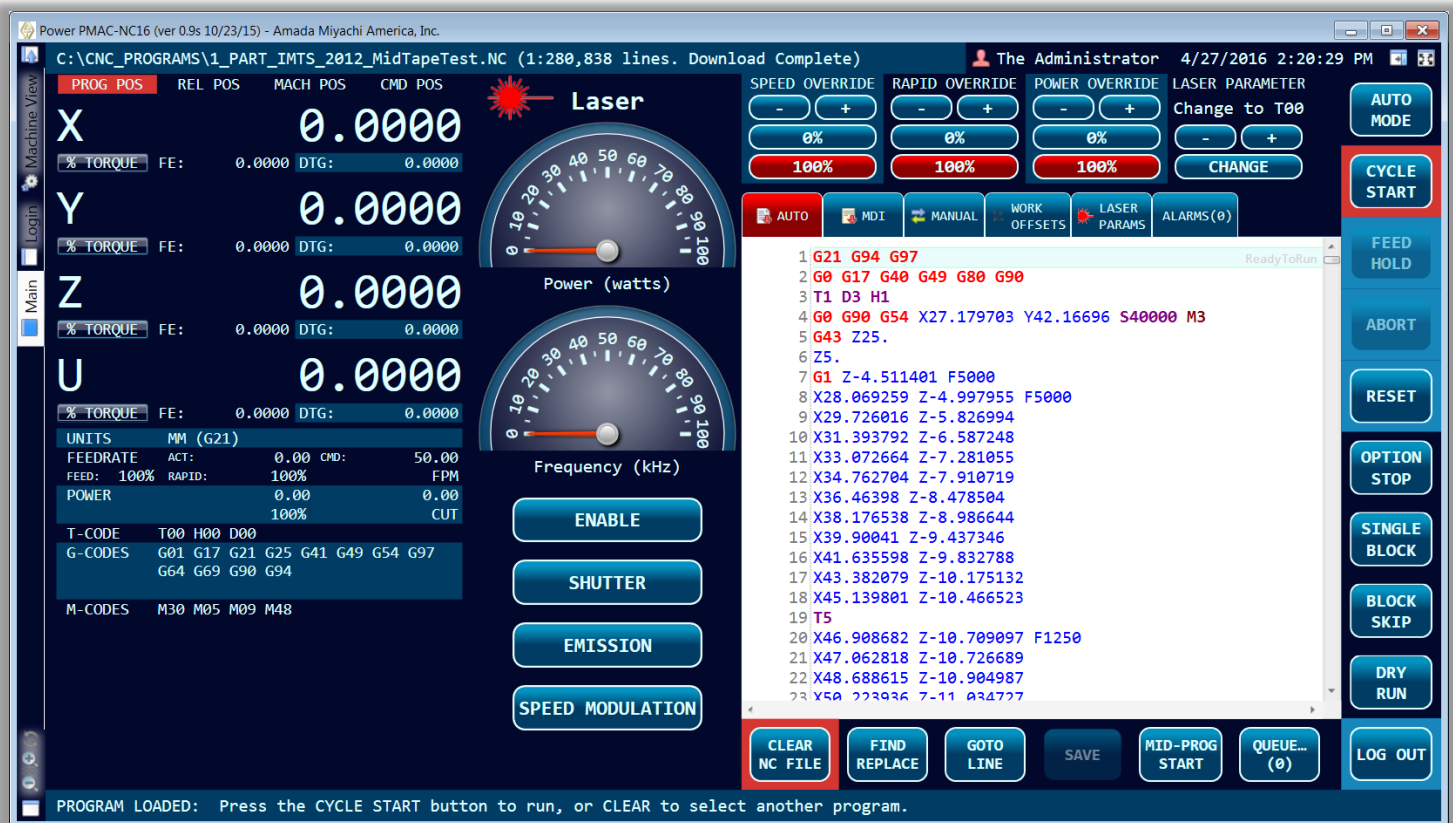
Allowed G/M Codes

```
;OnlyAllowedGMCodes=true  
; Allowed G/M-codes appear in Machine View plus these optional comma-separated lists.  
;AllowedGCodes=G5426,G5427  
;AllowedMCodes=M17,M18
```

The integrator should list all custom G&M codes which should pass in this parameter. The G&M codes should be listed as they would appear in the NC part program separated by commas. Refer to the Software User Manual for the standard G & M code list.

External Assemblies

Custom external assemblies can be created and referenced by the main Power PMAC-NC16 HMI. This is a powerful feature which allows customization of the screens without affecting the main source code of the project. See the Software User Manual for detailed information on how to implement these assemblies. An example of the use of an external assembly is shown below:



Center Custom Frame

```
Object="..\..\..\CenterPanelExample\bin\Debug\CenterPanelExample.dll;Cent  
erPanelExample.CustomObject"  
CenterCustomFrame="..\..\..\CenterPanelExample\bin\Debug\CenterPanelExamp  
le.dll;CenterPanelExample.PageCenterPanel"
```

The above custom center panel syntax activates a user created panel in the center of the application. Example code is included in the SDK package.



Left Custom Frame

```
Object="..\..\..\CenterPanelExample\bin\Debug\CenterPanelExample.dll;CenterPanelExample.CustomObject"
LeftCustomFrame="..\..\..\CenterPanelExample\bin\Debug\CenterPanelExample.dll;CenterPanelExample.PageCenterPanel"
```

The above custom left panel syntax activates a user created panel on the left side of the application. Example code is included in the SDK package.



Right Custom Frame

```
Object="..\..\..\CenterPanelExample\bin\Debug\CenterPanelExample.dll;CenterPanelExample.CustomObject"  
RightCustomFrame="..\..\..\CenterPanelExample\bin\Debug\CenterPanelExample.dll;CenterPanelExample.PageCenterPanel"
```

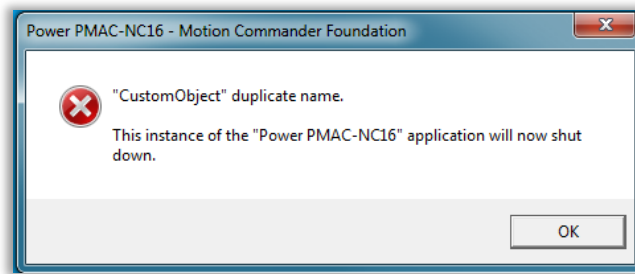
The above custom right panel syntax activates a user created panel on the right side of the application. Example code is included in the SDK package.



Custom Tab

```
Object="..\..\..\CenterPanelExample\bin\Debug\CenterPanelExample.dll;CenterPanelExample.CustomObject"  
CustomTab="..\..\..\CenterPanelExample\bin\Debug\CenterPanelExample.dll;CenterPanelExample.PageCenterPanel"
```

This section is designed to allow an integrator or user to create a custom tabbed view of a custom object. Procedures are as same as adding a custom frame. As seen above, two lines are required to be uncommented. If both lines including a **CustomTab** and **CenterCustomFrame** are uncommented, NC16 will not run and gives out the following error:



Private Label

```
[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"
```

This section is designed to provide a simple custom labeling and graphical identity option. As shown in the above figure, **CompanyName**, **SplashImage**, and **LoginImage** can be specified by the user. Examples are shown below. The image files should be located in the same folder as the executable file.

