Welcome

Welcome to NCPlot!

NCPlot v2.0
Copyright© 2005-2007
NCPlot Software, LLC

For the latest release information, news, or if you can't find what you need in this help file, please check the online forums at:

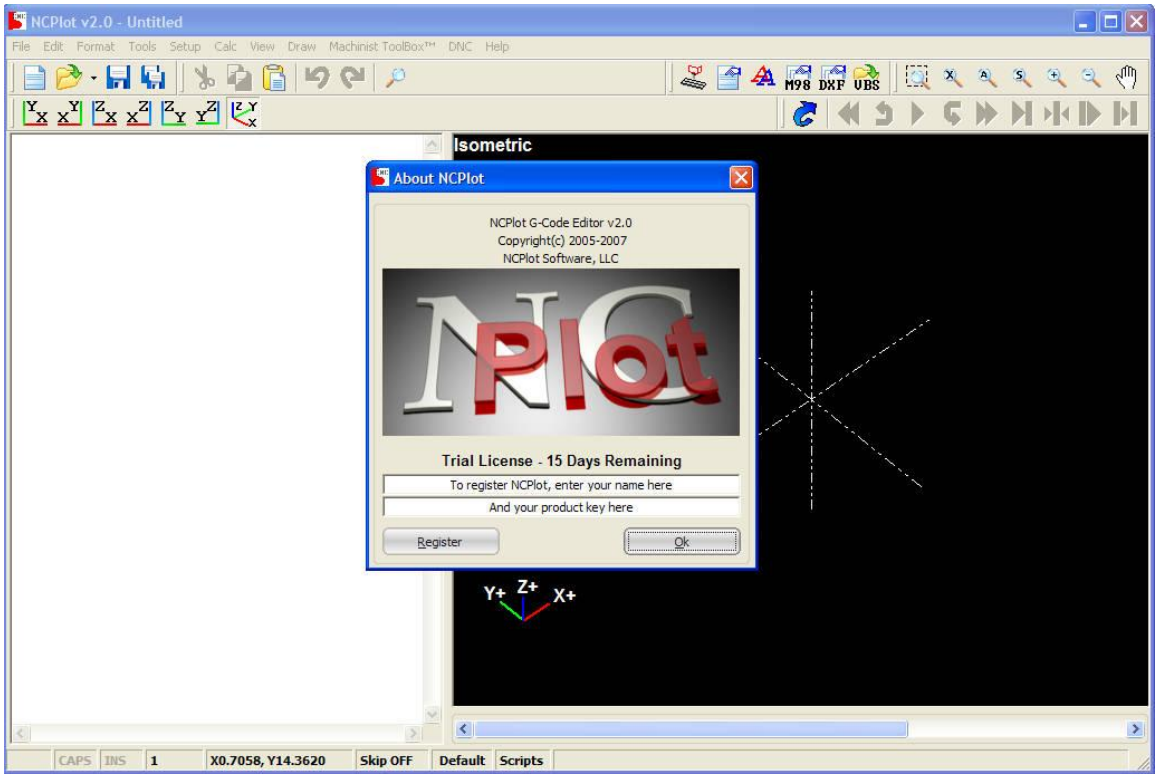
www.ncplot.com

We can also be contacted directly at:

Email: scottmartinez@ncplot.com

Getting Started

On startup, you are greeted with the following screen. If you have not registered the software, a window displays your remaining trial period time. You may click Ok to continue unregistered, or you may enter your name and key information.



The NCPlot window consists of:

The program edit area, where the G-Code program is shown.

The graphics viewport, where the graphic backplot appears.

The menu bar, which is used to access most of the program features.

The toolbars contain shortcut buttons to the most commonly used functions.

A status bar which displays information about the current state of NCPlot.

Registering NCPlot

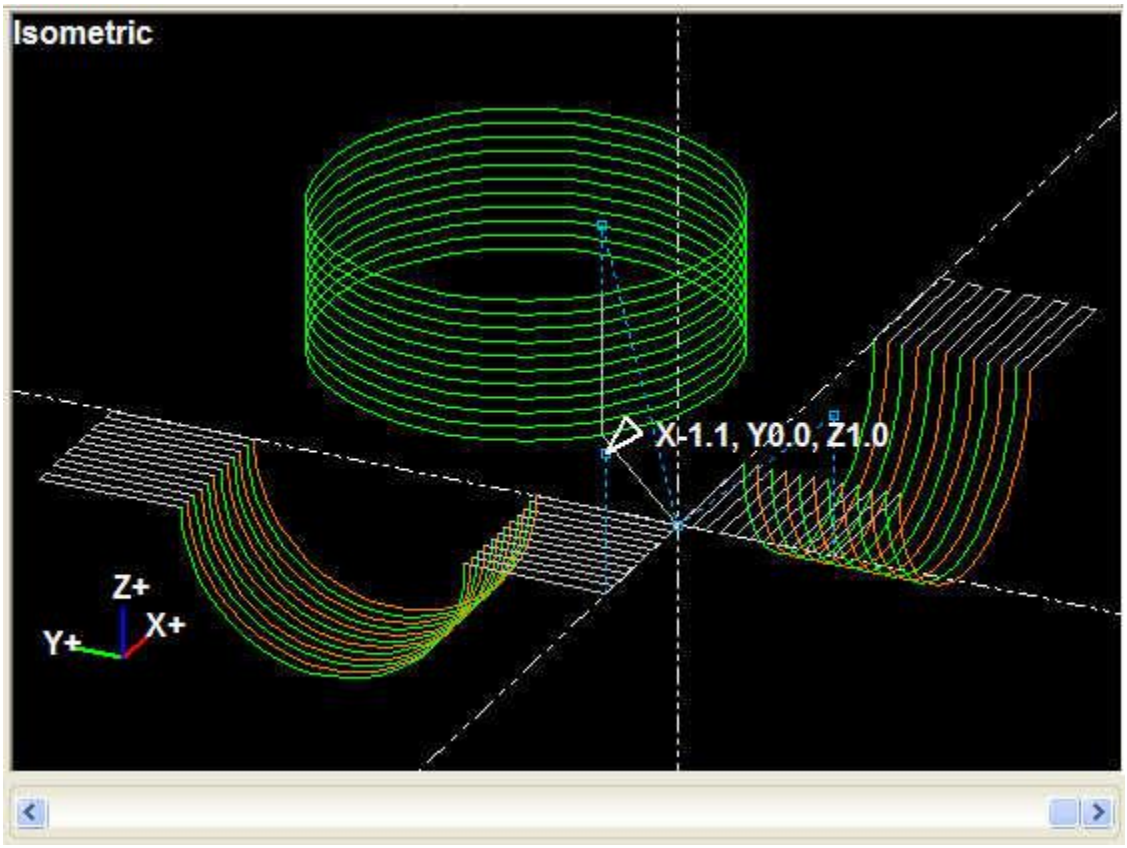
When NCPlot is first installed, you are given a 15-day trial period. During this time the software is fully functional, allowing you to evaluate it's suitability for your needs.

After the trial period has expired, NCPlot will no longer run without a registration key. These may be purchased from the online ordering page at http://www.ncplot.com/ The web site contains current pricing and ordering information.

To register NCPlot, enter your registration name and product key into the splash window. This window is displayed every time NCPlot starts, but may also be accessed from the menu ***Help*** **|** ***About***. Note that the splash window is not displayed at startup after the software has been successfully registered.

Using the Viewport

The graphics viewport in NCPlot displays the graphical representation of the programmed G-Code tool path. The viewport not only shows you if your program will work as expected, but it also provides help when the results are not what you expected. Using the dynamic pan, zoom and rotate you can quickly spot parts of the tool path that are not correct. The plot controls such as animate, step forward and step backward can then help you locate the trouble spots in the program.



In addition to the tool path graphics, there are several other useful items on the viewport:

The first is the **View Name**, which appears in the top left corner of the viewport and describes the active view orientation. This gives a reminder of which way the view is looking at the part.

The second item is the **Orientation Icon**, which, like the **View Name** serves to show you which way the part is oriented on the viewport. The icon appears in the lower left corner of the viewport and indicates the positive direction for each displayed axis.

The third item is the **Axis Lines**, which are drawn to indicate where the active zero point is. The zero point can represent the machine zero or any of six programmable work zero locations.

The fourth item is the **Marker Icon**. This icon is an arrow shaped pointer that appears on the viewport and shows the location of the active block endpoint. In addition to marking the location on the viewport, the marker also displays the coordinates of the active point.

The last item is the **Viewport Slider** control. The slider can be dragged with the mouse to quickly advance or rewind the plot to any point in the program. Likewise, when stepping or animating the graphics the slider moves to show the current progress.

Panning, Rotating and Zooming

The viewport can easily be manipulated using just the mouse or keyboard, no buttons or commands are required to activate these functions. The controls vary slightly between the 2D and 3D views. The 3D view refers to the Isometric view. This view mode is only available for Mill machine configurations. All other view modes are 2D views.

3D (Isometric)
Pan - Press and hold the "Shift" key and the right mouse button while moving the mouse.
Rotate - Press and hold the right mouse button while moving the mouse.

The view will rotate in two directions. Moving the mouse left or right rotates around the viewport vertical axis, and moving the mouse up or down rotates around the viewport horizontal axis. The up and down rotation can be locked by checking the menu item ***View | Lock Vertical Rotation***. When this is checked you can temporarily unlock it by holding the Ctrl key and rotating the view.

2D (All others)
Pan - Press and hold the right mouse button while moving the mouse.

There are several zoom tools which apply to all view modes. The keyboard shortcut keys require that the viewport be active, which means that you should click it before using them.

Mouse Wheel - The view zooms in or out as the mouse wheel is rolled.
Zoom Window - "Z" Key - Allows dragging a box around an area to fit into the viewport.
Zoom Extents - "X" Key - Fits the part drawing into the viewport, including rapid motions.
Zoom All - "A" Key - Fits the part drawing into the viewport, disregarding the rapid motions.
Zoom Selected - "F" Key - Fits only the selected entities into the viewport.
Zoom In - "C" Key - Increases the zoom magnification, making the part appear larger.
Zoom Out - "V" Key - Decreases the zoom magnification, making the part appear smaller.
Pan – “P” Key – Activates the viewport pan tool.
Step Forward – “S” Key – Draws the next motion block in the program.
Step Backward – “B” Key – Un-draws the previous motion block.
Measure – “M” Key – Activates the measure tool.

Adjusting the size of the viewport

The size of the program edit window and the viewport may be adjusted by clicking and dragging the vertical bar that divides the program editor and the viewport. When the mouse pointer is moved over this bar, the pointer changes to an arrow that points left and right. Click and hold the mouse button, move the mouse to the new location for the bar and release the button.

The viewport and the edit window may be swapped by opening the ***Preferences*** window and checking the option ***Viewport on the Left***.

Double-clicking the viewport will hide the program edit window to allow this area to be used by the viewport. This is quick and easy and will remember the size of the edit window when it is restored. Simply double-click it again to restore it.

Setting up viewport colors

Most of the viewport colors may be customized including the background, selected entities, marker icon and the part display. All of the color settings are defined by the machine configuration. To open, click the menu ***Setup*** then click ***Machine Configuration***. On the machine setup dialog click the ***Viewport Colors*** page.

There are two part coloring methods to select from, called ***Color by G-Code*** and ***Color by Tool***. The first method, ***Color by G-Code*** will draw the part using colors to represent the type of motion that each entity represents. There are four colors, one each for G00 Rapid, G01 Feed, G02 Clockwise arc and G03 Counterclockwise arc. The second method, ***Color by Tool*** will change color each time a tool change is encountered in the program. A tool change can be indicated by an "M06" tool change code or by any commanded "T" code. This method allows you to quickly see which areas of the part are being cut by each tool in the program. A list of colors are defined and each time a tool change occurs in the program, the viewport switches to the next color in the list.

The other selectable colors are the viewport background color, the color used for selected entities and the color used for the viewport marker icon. Only the axis lines, the axis orientation icon and the viewport title colors are not definable. The color of the axis lines and viewport title will either be black or white, depending on the background color of the viewport. Dark background colors will cause these items to display in white, and bright background colors will cause them to display in black.

Selecting entities on the viewport

The ability to select the entities that make up the backplot of your part is an extremely useful feature of NCPlot. When an entity is selected its color changes to the selection color, and a small square is drawn around its endpoint. The marked endpoint then gives a visual indication of the direction that the entity will machine in. There are three selection methods:

Clicking - Left clicking any of the entities on the viewport will do three things. (1) It will select the entity. (2) The block in the program that created it will be highlighted. (3) The entity info window will show the properties of the selected entity. Multiple entities may be selected by holding the Shift key while left clicking additional entities. Entities may also be unselected by holding the Ctrl key while left clicking selected entities.

Window selecting - Groups of entities can be selected by simply dragging a box around them. Click and hold the left mouse button at one corner of a box and drag the mouse and release the button at the opposite corner. A box will be drawn as the mouse is moved to indicate the area containing the entities to be selected. The box will have either a solid line border or a dashed line border depending on which direction you drag the box. A solid line border appears when you drag to the right and will select everything that is *completely inside* the box when the mouse button is released. A dashed line border appears when you drag to the left and will select everything that is *inside of* ***or*** *touching the borders of* the box when the mouse button is released. Holding the Shift key while window selecting will add the selected entities to the current selection set.

Calc tools - The "***Calc***" menu contains additional tools that make it easier to select groups of entities. For example, the "***Select Chain***" tool is useful for selecting an entire 2D

profile. Just select one entity that is part of the profile, then select this tool. All entities that are connected to the selected one and at the same Z depth are selected automatically. The "***Select by Z***" tool will go through the entire backplot and select all entities that are at the same Z depth as the currently selected entity. This makes it easy to select all profiles that are at a given depth. The "***Select Arc by Radius***" tool will select only arc entities that are the same radius as the currently selected arc. You could then create points at the center of each selected arc and use these points to create a drill program.

After a group of entities has been selected, there are a few things that you can do with them:

Export as DXF file - The ***File*** menu has an option called ***Export Selected as DXF File*** that will enable you save a DXF file that contains only the entities that you have selected. This can save a lot of work deleting unnecessary geometry from a drawing that contains the entire backplot.

Calc Tools - The Calc tools such as offset and blend radius require that one or more entities be selected. These tools are applied only to the selected entities.

Convert to G-Code - This tool will use the selected entities to create new G-Code snippets. This might not seem very useful at first, after all the selected entities were created from G-Code in the first place, right? Not necessarily, the Calc tools can be used to create new geometry which you can then turn into new G-Code with this tool. You could also take a backplot from a simple 2D profile and use it to create multiple Z passes.

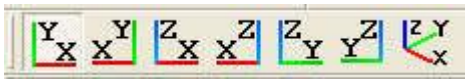
Delete – Pressing the DEL or Delete key on your keyboard will remove the selected entities from the viewport. This is useful for removing clutter when trying to isolate a particular area of the backplot. This does not remove them from your program and refreshing the viewport will restore the deleted entities.

Toolbars

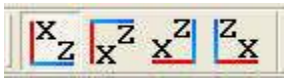
The ***File*** toolbar contains buttons for commonly used file functions. The ***File Open*** button also contains the list of recently opened files. This list may be cleared with the menu item ***File | Clear Recent Files***.

The ***Edit*** toolbar contains buttons for the clipboard operations ***cut***, ***copy*** and ***paste*** as well as ***undo***, ***redo*** and ***find***.

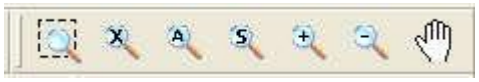
Mill Views



Lathe Views



The ***View*** toolbar contains buttons for changing the selected viewport orientation. This toolbar changes depending the currently configured machine type. There are seven view buttons for the **Mill** configuration and four for **Lathe**.



The ***Zoom*** toolbar contains buttons that change the graphic display size and location. There are buttons for ***zoom window, zoom extents, zoom all, zoom selected, zoom in, zoom out*** and ***pan***.

The ***Plot*** toolbar contains buttons that allow you to control the plotting of your program. There are buttons for ***refresh plot, rewind plot to beginning, step backward, animate, step forward, fast forward plot to end, plot to cursor, start at cursor, plot from cursor*** and ***plot selected blocks only***. When editing your program, the ***refresh plot*** button will change from blue to green to indicate that the graphics should be refreshed. You do not need to save the program before refreshing.

The toolbars may be rearranged to your liking by clicking and dragging the control handle on the left side of each toolbar.

Status Bar

The status bar is made up of ***panels***. Some panels contain information about the current state of NCPlot, while some allow quick access to NCPlot features. The panels from left to right are:

Edit Status indicates that changes have been made to the loaded file.

Caps Lock Status indicates when the keyboard caps lock is active.

Insert Status indicates when the keyboard insert is active.

Current Line Number indicates the line number that the cursor is on.

Position panel indicates the location of the mouse pointer in machine coordinates. This location will only update while the mouse pointer is within the viewport.

Block Skip Status indicates the current status of the block delete toggle. Clicking this panel toggles between the Skip ON and Skip OFF state. This toggle is equivalent to the Block Delete switch you would find on your machine control panel. When ON, this toggle causes NCPlot to ignore (skip) program blocks that begin with the block delete character "/".

Current Configuration indicates the name of the currently loaded machine configuration settings. Clicking this panel will display a list of the available configurations. Click one of the items on this pop up list and the selected configuration is then loaded.

Scripts panel is a shortcut to the available script files. Clicking this panel will display a list of the available scripts. Click one of the items on this pop up list and the selected script is executed. The available scripts are stored in the folder \Program files\NCPlot\Scripts

Messages panel displays messages related to the active operation.

Machinist ToolBox™ Addin

NCPlot supports the Machinist ToolBox™ ActiveX product from CNC Machinist Software.

http://www.machinist-toolbox.com/

When this product is installed, NCPlot will find it and include it in the menu bar. This gives you instant access to all of Machinist ToolBox's features right from NCPlot.

It is recommended that you configure NCPlot to be the default editor for Machinist ToolBox™. This allows posted G-Code to be sent right back to NCPlot, giving you an immediate backplot without needing to close Machinist ToolBox™.

Plot Settings

There are many settings that can affect the way your program will be plotted. Most settings can be found on the Machine Configuration dialog. This is found on the menu ***Setup | Machine Configuration***.

These options are found on the ***Draw*** menu. Each of these options will display a check mark next to them on the menu to indicate that they are enabled.

Axis Lines

Displays two or three (depending on the view) intersecting lines that indicate where 0,0,0 is on the viewport. The axis lines display may indicate the machine zero or any of the six work offsets.

Show Rapid Moves

Displays or hides rapid motion lines on the viewport.

Show Ticks

When enabled, a small square is drawn at the endpoint of rapid moves. This setting has no effect when ***Show Rapid Moves*** is not enabled.

Show Plunge Moves

When disabled, this option will hide Z axis movements in the negative direction. This only affects movements that are Z axis only.

Absolute Arc Centers

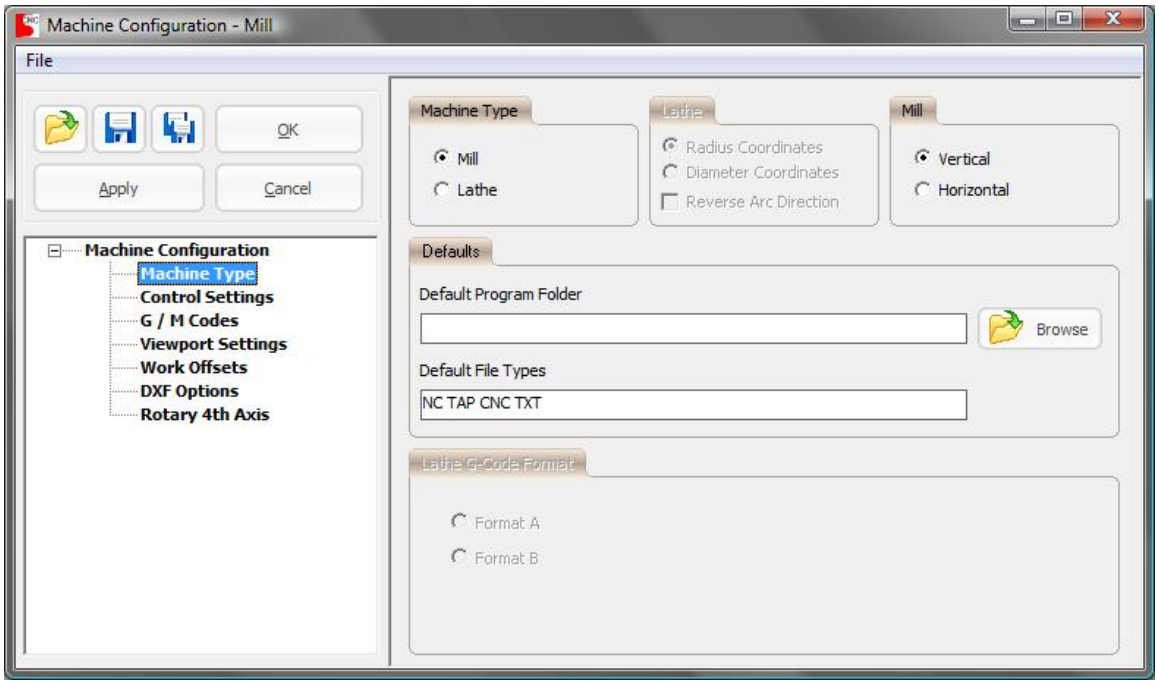
When checked, this option specifies how I / J / K specified arc centers are drawn. This option also affects the results of the arc conversion tools. This setting may also be changed on the ***Machine Configuration*** dialog under the ***Control Options*** tab.

Setting up a Machine Configuration

In order for the graphics viewport to properly display your G-Code program, it must first know a few things about the machine you intend to run it on. Since there are many different types of machines and CNC controls, NCPlot has options that allow it to mimic the way your particular CNC control reads G-Code. NCPlot doesn't recognize every G or M Code that your control does, but it should still be able to give you a good representation of your programs toolpath. Besides backplotting, the machine configuration is important for another reason. Some of the conversion tools require that the backplotter be properly configured in order to give the desired results. For example, if the arcs in your program do not look correct when plotted, the arc conversion tools will not work correctly. In general, if the plot looks correct, the conversion tools will work the way they're supposed to.

NCPlot comes with a handful of predefined machine configurations. These configurations represent the most common settings for a CNC control and should be good enough to get you started. Even so, you should check that these settings match the way your control works.

To open the machine configuration dialog, click the menu ***Setup***, then click ***Machine Configuration***. This dialog is made up of several pages, the first page you see is labeled ***Machine Type***. This page has settings that define the basic setup of your machine.



In addition to the configuration settings, there are a number of buttons that are for managing your configurations:

Open allows you to browse for and open an existing configuration.

Save will apply the current changes and save them to the active configuration.

Save As will allow you to specify a new configuration file and save your settings. Note that when a new configuration is created, its canned cycles are copied from the currently active configuration.

Machine Type Page

You must first select between Mill and Lathe. Choosing one or the other will change or enable/disable other settings on the dialog. If you selected Mill, you now have the option to select between Vertical spindle and Horizontal spindle. If you selected Lathe, you now have the option to select between Radius Coordinate values and Diameter coordinate values. This setting determines how NCPlot interprets the X/U axis command values. The Lathe type also has a check box that allows the direction of G2/G3 arc commands to be reversed.

Also on this page is a setting called ***Default Program Folder***. This setting can be set to point to a folder where the G-Code programs for this particular machine configuration are stored. Say for example you have a configuration for a Makino vertical machining center. All the programs for it are stored at "C:\Jobs\MakinoVMC". Simply set the default program folder to this folder, then any time you want to open a file, the File Open dialog will open right to this folder. Since this setting is part of the machine configuration, you can specify a different folder for each configuration. The ***Default File Types*** setting is a list of file extensions that you want to associate with your programs. This determines which file types are listed whenever you browse for a file to open or save. Wildcard characters may be used as part of the extensions.

For lathe configurations you may choose between G-Code ***Format A*** and G-Code ***Format B***. These two formats differ between some of the G-Codes. When you select one or the other, a list of the G-Codes and their function are displayed. Select the format that most closely matches your control.

Control Settings Page

This page contains some of the most important settings for determining how your G-Code programs are interpreted. First off is the ***Rapid Type*** setting. This setting should be set to match how your machine responds to a multiple axis simultaneous rapid move. Select ***Interpolated*** if all your machine axes arrive at their endpoints at the same time. If the axes reach the endpoints one at a time, this would be ***Non-Interpolated***, sometimes called "Dog-Leg". Some controls use a third method, which is generally safer than the other two. This method is called ***Z First/Last*** and will always move the Z axis by itself, either before or after the X & Y axes depending on which direction the Z is going.

If you intend to backplot programs in the Custom Macro B format you should set the ***ATAN Function Format*** setting. This setting determines the format that is expected when an ATAN function is encountered in the program. In general, Fanuc controls expect the two operand format, while Mitsubishi controls expect the single operand format. For others, check your control documentation to determine the correct setting.

The ***Coordinate Resolution*** setting determines how many decimal places to assume when a command value is given without a decimal point. For example, if you have a program that has commands like "Z-152500", then you would want to set the coordinate resolution to "0.0001" so that this would be properly interpreted as "Z-15.2500". Here are some more examples:

Command value	Coordinate Resolution	Interpreted value
X25	0.001	X0.025
X1	1.0	X1.0
Y1250	0.0001	Y0.1250
Y1.250	n/a	Y1.25

Since a decimal point was specified in the last value, the resolution setting is disregarded.

The ***G04 Dwell Address*** setting allows you to define which letter address your control uses as the dwell time. Common settings are X, P, F and T.

The arc settings determine how G02 and G03 arc commands are interpreted. If your control uses absolute arc centers, then check the setting ***Absolute Arc Centers***. When checked, the I, J and K values in a G02 or G03 command represent the location of the center of the arc in the current work coordinates. When unchecked, the I, J and K values represent the distance from the start point of the arc to the center point of the arc.

If your control uses absolute arc centers, it may also treat the center locations as modal. If this is the case, the control remembers the last center point you programmed and you don't have to include an I, J or K value in every arc command. If you have a control that behaves this way, check the setting ***I/J/K values are modal***.

When commanding an arc using IJK arc center designation, it's not uncommon for there to be a small difference between the arc's start radius and end radius. That is, the difference between the distances from the start point to the center and the distance from the end point to the center. Most controls will handle this without a problem up until the difference reaches a certain amount. Whether this amount is fixed in the control, or is parameter settable, you can enter this amount into the ***Arc Tolerance*** setting. When NCPlot encounters an arc where the start and end radius is different by more than this amount, an error message will be displayed.

When NCPlot begins to backplot a program, it starts from a fixed G-Code state. That is, certain G-Codes are active by default such as G00, G90, G54 etc. While this is acceptable for most controls, you may have a machine that defaults to some other active state, like G91. The ***Initial State*** setting is used to define the default state of your control. For example, if your control defaults to G91 you simple add "G91" to the Initial state setting.

G/M Codes Page

If you plan to backplot programs that use M98 for subprograms, then it's very important that you set the M98 command format to match your control. There are five different settings, so if you're not sure which one to use, you should consult your control's programming manual. For details about this setting, see the topic Plotting Subprograms.

The ***G-Code Macros*** setting is a list of G-Codes that NCPlot will call as subprograms when they are encountered in a program. When encountered, all other address values are written to local variables and a specially named subprogram is loaded. The name of the subprogram that is loaded is in the format "Gxxx.PRG", where "xxx" is the G-Code value times 10. For example, if you have G12 in the G-Code macro list and NCPlot encounters the block "G12 X0 Y0 I0.5", a subprogram named "G120.PRG" must be in the configuration folder. The values for X, Y and I are saved to local variables and can be used by the subprogram to simulate the motion for a G12 command. This method allows you to simulate G-Codes that are not handled internally by NCPlot.

If your control supports M-Code activated mirror image, then use this page to set the M-Codes that are used to activate this function.

Viewport Settings Page

This page contains settings that define the colors used to draw the backplot. You first must decide if you want to color by G-Code, or color by tool. To select one, check the box next to the header describing the method you want to use. When ***Color by G-Code*** is selected, the entities on the graphics viewport be will colored according to the type of motion it represents. There are 4 basic types of motion: G00 Rapid move, G01 Feed move, G02 Clockwise arc and G03 Counterclockwise arc. Each of these types of motion may be assigned a different color.

The ***Color by Tool*** option draws the backplot with different colors representing the range of motion for each tool used in the program. The ***Unspecified Tools*** color is used when the program commands motion before the first tool change or when there are more tools used in the program than have been defined. The color list contains the colors to use for each tool. The first color in the list is used after the first tool change, the second color after the second tool change, etc. If there are not enough colors in the list for all of the tool changes in the program, the ***Unspecified Tools*** color will be used for any remaining tool changes. You may also specify the type of command that is considered a tool change, either the M06 command or a T-Code.

In addition to the entity colors, you can also specify the background color of the graphics viewport as well as the color of entities that are selected.

The ***Top Viewport*** rotation setting allows you to re-orient the graphics display to match the way the part appears from the operator side of the machine. This is simply a convenience setting that only affects the graphics view.

Work Offsets Page

Just like your machine can accommodate multiple work offset coordinates, NCPlot can also be configured to recognize multiple work locations. This gives a backplot that accurately represents a multiple fixture setup.

DXF Options Page

Use this page to enter default DXF drawing conversion settings. When a DXF file is loaded, any layer that is not in your saved layer list will be given these settings. For more information see the section ***Converting DXF Drawing Files***.

Rotary 4th Axis Page

If your machine has a rotary 4th axis, use this page to define the settings for it. First, set the ***4th Axis Identifier*** to specify the letter address that commands the 4th axis. The most common settings are an "A" or "B" axis. Next, set the orientation of the rotary axis by specifying whether it rotates around the "X" or "Y" axis. By definition, an "A" axis rotates around the "X" axis and a "B" axis rotates around the "Y" axis. You must also set the ***Coordinate Resolution*** setting for the 4th axis command values. This works the same way as the setting on the ***Control Options 1*** tab.

The ***Rotary Centerline*** settings can be used to specify where on your machine the rotary axis is located. This tells NCPlot where the center of rotation is located on the machine.

Converting DXF Drawing Files

Load the drawing

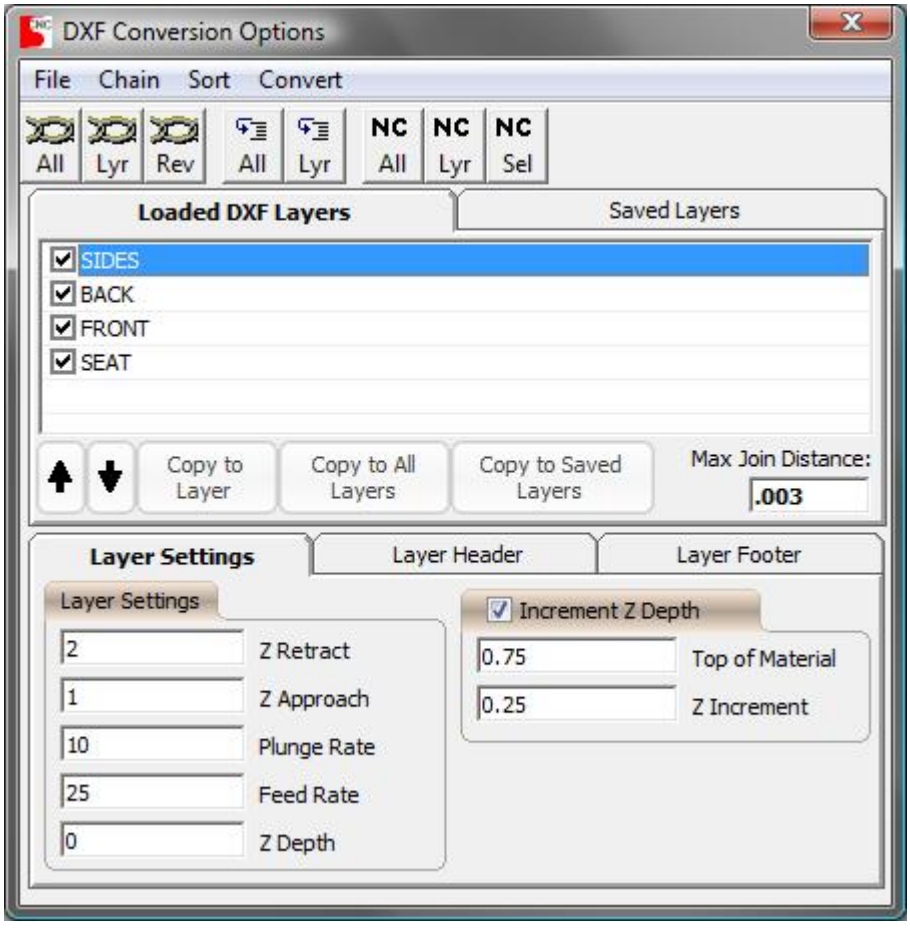
There are several ways a DXF drawing file can be loaded into NCPlot.

- Use the menu ***File / Import DXF file***
- Use the menu ***File / Open file*** or the ***Open File*** toolbar button then select ***DXF drawing files*** from the files of type list.
- Drag and drop a DXF file onto the NCPlot edit window

Opening a DXF file will not clear the loaded program. If adding code from a drawing into an existing program, move the cursor to the point in the program where you want the new code to be added. If you want to create a new program from a drawing, close the existing program before opening the drawing file.

Once a file has been selected, NCPlot loads it and displays it on the viewport. Once loaded, a dialog appears that will allow you to control how the drawing is to be converted to G-Code.

The DXF Conversion Options Dialog



The NCPlot DXF converter is layer based. That is, the same machining settings are applied to all drawing entities on the same drawing layer. This also affects geometry chaining and sorting (explained later). This should be taken into account when assigning layers to various parts of your drawing.

The top half of the conversion dialog consists of two tabs labeled *Loaded DXF Layers* and *Saved Layers*. Each of these tabs contains a list of layers. The Loaded DXF Layers tab contains the list of layers that was loaded when the drawing file was opened. The *Saved Layers* tab contains a list of commonly used layers that you can store machining settings for. Anytime a drawing is loaded that contains a layer with the same name as one of the saved layers, the saved settings for that layer are used.

The lower half of the conversion dialog contains the machining settings for the currently selected layer. The settings are divided among three tabs labeled *Layer Settings*, *Layer Header* and *Layer Footer*. The *Layer Settings* tab contains the machining settings which includes Z depths, feedrates and an option to create multiple passes at incrementally lower Z depths. You can also add header and footer text to the G-Code output of each layer, these settings are found under the *Layer Header* and *Layer Footer* tabs.

Turn off unwanted layers

Since the drawing file may contain information that you don't necessarily want converted to G-Code, the *Loaded DXF Layers* list provides a means to turn off unneeded layers. Clicking the checkbox next the layer name will either turn the layer off or on, a check mark indicating that the layer is on. The viewport graphics will update at the same time, displaying only the layers that are turned on.

Arrange the layer list

In addition to allowing you to turn off layers, the layer list also provides a means of controlling the order that the drawing is converted to G-Code. This gives you a level of control over the order that your part will machined in. The layers in the layer list are converted to G-Code in the order that they appear in the list. To move a layer, select it from the list and use the *up arrow* or *down arrow* buttons (not the keyboard keys) to change its order in the list.

Set the machining parameters

When each layer is loaded they are initially assigned the default layer settings, which comes from the current machine configuration. The exception to this is when a loaded layer name matches one of the saved layer names. In this case the layer is assigned the saved layer settings.

The layer settings that appear on the lower half of the conversion dialog are for the currently selected layer. To select a layer, click it's name in the layer list. When a layer is selected, it's name is highlighted in the layer list and the layer settings will update to show the settings for the selected layer.

Because of the way that NCPlot creates the G-Code output, it is important to set the Z depth settings in a logical order:

- Z Retract should be the highest (most positive) value, followed by:
- Z Approach

- Top of Material
- Z Depth should be the lowest (most negative) value.

Changing one of the layer settings only affects the currently selected layer. To copy settings from one layer to another, first select the layer you want to copy then click the button ***Copy to Layer***. This button turns green indicating that you should now click the name of layer you want to copy the settings to. To copy the same settings to all loaded layers, first select the layer you want to copy then click the button ***Copy to All Layers***.

If your drawing has layers that you use often, you can copy them to the ***Saved Layer*** list for later use. To copy a layer to the saved layer list, first select it from the layer list the click the button ***Copy to Saved Layers***. The layer name and all of it's settings are then copied to the saved layer list.

Chain the drawing

Since a DXF drawing file doesn't provide the geometry data in any particular order, we need a means of identifying which parts of the drawing are connected together to form a continuous path. This is done with the chaining tool. The chaining tool will scan each layer and find all the geometry that appears to be connected together. The ***Max Join Distance*** setting determines how close the endpoints of two entities must be in order to be considered joined. This lets NCPlot create more efficient G-Code without a lot of seemingly random cutting. Since the converter is layer based, the chaining tool will only join geometry that is on the same layer.

There are two chaining tools: ***Chain All***, which will chain all the layers in the drawing and ***Chain Layer***, which will chain only the currently selected layer.

An additional benefit to the chaining tool is that it allows reversing the direction of chained geometry. Since the direction of the geometry determines the cutting direction, this allows for control over the cutting direction. To see the current cutting direction, click any of the entities on the viewport that are part of the chain. When an entity is highlighted, a small square box is drawn around the end point of the entity, indicating its direction. To reverse the chain direction, click the *Chain Reverse* toolbar button. The chains belonging to any selected entities will be affected.

Note that chaining is required for layers that have the ***Increment Z Depth*** setting enabled.

Sort the drawing

The sorting tool provides an additional means of optimizing the G-Code output by attempting to arrange the drawing in a way that will result in less rapid motion between parts of the drawing. It does this by starting at one corner and finding the closest part of the drawing. The next closest part of the drawing is found next and so on. This tool works with chained geometry, so the chaining tool must be applied before this tool can be used.

Convert to G-Code

There are three conversion tools, giving different levels of control over the order that the drawing is converted in. The ***Convert All*** tool will convert the entire drawing to G-Code in the order that the layers are listed. The ***Convert Layer*** tool will convert only the selected layer to G-Code. The third tool, ***Convert Selected***, will convert only the chains belonging to any selected entities on the viewport. This gives the most control over the conversion process, but only works with chained geometry.

Plotting Subprograms

NCPlot provides you with the ability to backplot G-Code programs that make use of sub programming. This includes support for M98, G65 and M97 subprogram call codes. But, like the machine configuration, there are some settings that need to be configured to match the way your particular control works.

Of the three subprogram commands, M98 is the only one that varies between different controls. So, the ***Machine Configuration*** dialog contains an option that let's you tell NCPlot which command format your control uses. This is on the ***G/M Codes*** tab and there are five possible settings:

M98 P1 L1

This is the most common setting and works for most Fanuc controls. The "P" address is the program number to call as a subprogram, and the "L" address is the number of times to repeat the sub.

M98 P1 K1

Some older Fanuc controls use the "K" address as the repeat count instead of the "L" address.

M98 P010002

Some Fanuc controls combine the subprogram number and the repeat count into a single 6 digit number. The first two digits are the repeat count and the last four digits are the subprogram number. So, in this example the repeat count is 01 and the program number is 0002.

M98 P1 H1 L1

Mitsubishi controls add the use of the "H" address which represents the target block number in the subprogram being called. So, besides the program number and the repeat count, you can also specify a starting block number for the subprogram.

M98 O1

For some other types of controls, M98 is a local subprogram call that uses the "O" address to specify a target block number. This type of subprogram call cannot use other programs as subs, the subprogram must be within the current program.

The other two subprogram call commands use fixed formats, which are:

M97 P1

This code is used by the HAAS controls and is a local subprogram call that uses the "P" address to specify the target block number. This type of subprogram call cannot use other programs as subs, the subprogram must be within the current program.

G65 P1

This code is commonly used for macro programming because any address values that appear after the G65 are copied to local variables before the subprogram is called. The exception is the "P" address, which represents the program number being called. This makes it very useful for creating custom cycles that work much like the control's built-in cycles.

Once you've got NCPlot set up to read the subprogram calls correctly, you just need to be able to tell it where the subprograms are on your computer. On the CNC control, this is not an issue because the control knows where all the programs are, but on your PC this may not be so simple. If you only use the local subprogram call commands M97 or M98 O, no setup is required because the subprogram must be within the main program.

Where to put your subprograms

The simplest way to handle subprograms is to include them in the file you are editing. However, this is not always practical and so NCPlot provides a way to specify where the subprogram files are stored. The subprogram setup window can be opened by clicking the ***Setup*** menu and then **Subprograms**. This window lets you define where your subprograms are located as well as what action to take when a subprogram command is encountered. The possible actions can be set separately for M98 and G65 commands and are:

Error if sub not found - If the commanded subprogram could not be located, NCPlot stops plotting and displays an error message.

Ignore if sub not found - If the commanded subprogram is found it will be plotted, otherwise NCPlot will skip it.

Ignore all - All subprogram commands will be ignored.

When NCPlot encounters a subprogram call in your program, it begins looking for the requested program number. The order in which the search takes place is:

1. The current file is searched first.
2. The subprogram association list is searched next.
3. The subprogram default folder is searched next. This is set on the subprogram setup window.
4. The folder of the active edit program file is searched last.

The subprogram association list allows you to assign a specific file to a specific program number. To create a new association, click the "Add" button on the subprogram setup window. You will be prompted for the program number to assign. Enter the program number and you can then browse for the file that contains the G-Code program for the specified program number. There is no limit to the number of associations you can assign.

File Menu

Open File

This selection allows you to browse for an existing file to be loaded into the edit window. If any unsaved changes have been made to the current program, you are prompted to save it before loading a new file. The file browse window contains a drop down list that allows you to select the types of files to be displayed, either plain text files of various extensions, .DXF drawing files or all files. If a .DXF drawing file is selected, the .DXF conversion options window will open instead of the file being loaded into the edit window.

After a file is loaded, it will automatically be plotted and zoomed to the size of the viewport. Either plain text or rich text (RTF) files may be loaded, the format will be determined automatically.

Insert File

This selection allows you to browse for an existing file to be inserted into the edit window. The selected file will be inserted at the current cursor position. If any unsaved changes have been made to the current program, you are prompted to save it before loading a new file. The file browse window contains a drop down list that allows you to select the types of files to be displayed, either plain text files of various extensions, .DXF drawing files or all files. If a .DXF drawing file is selected, the .DXF conversion options window will open instead of the file being loaded into the edit window. Either plain text or rich text (RTF) files may be inserted, the format will be determined automatically.

Close File

This selection clears the current program and the viewport. If any unsaved changes have been made to the current program, you are prompted to save it before clearing the program.

Import DXF File

This feature allows reading files that are in the Drawing eXchange Format. You are first prompted to select a file for importing. The selected file is then loaded and displayed and you are presented with the DXF conversion options dialog.

Export DXF File

This option will save the current G-Code program as a DXF drawing file. The drawing file will include all axis motion except G00 rapid moves. When the machine type is configured for lathe, the exported motion is translated from the ZX plane to the XY plane.

Export Selected as DXF File

This tool is similar to the Export DXF File tool, except that only currently selected viewport entities will be saved.

Save File

Saves your current edits to the loaded file. If the current file is untitled, you will be prompted for a filename to save it as.

If you want to save document formatting with your file (colors, fonts, etc.), you can specify rich text format (RTF) as the file type. Note that formatted documents cannot be run by most machine controls, so this feature is primarily for documentation purposes.

Save As

Saves your program under a new name. If you want to save document formatting with your file (colors, fonts, etc.), you can specify rich text format (RTF) as the file type. Note that formatted documents cannot be run by most machine controls, so this feature is primarily for documentation purposes.

Save As Separate

Saves all programs in the loaded file as individual files. You will be prompted for a folder to save to, and the files are saved with the program names used as the file names. For example, if you had the following file the saved files would be O100.txt, O1000.txt and O1001.txt.

```
%
O100(PROGRAM_100)
G0X0Y0Z1.
M98 P1000
M98 P1001
M2

O1000(SUB_1)
G91G1X5.F100.
M99

O1001(SUB_2)
G91G1X-5.F100.
M99
%
```

NCPlot recognizes the start of a new program when it encounters an “O” word or a colon character “:” at the beginning of a line followed by a program number. For a report of all the programs in the file, see the “Show Programs in File” tool.

Show Programs in File

This tool reports a list of all programs it finds in the loaded file. Each program number is reported along with its approximate size and program comment. NCPlot recognizes the start of a program when it encounters an “O” word or the colon character “:” at the beginning of a line.

Here’s an example report:

```
   Program    Size  Description
==============================
      6000    4141  LINK MAIN-PALLET A
      6001    1669  LINK MAIN-PALLET B
      8000     198  PROG FOR PART RESTART-GENERIC
      8001     172  TEST TO QUALIFY TOOL-
```

Execute Script File

Allows you to browse for and execute a script file. NCPlot provides a folder for commonly used scripts in the “Scripts” subfolder located in the application install folder. The default location for this folder is “\Program Files\NCPlot\Scripts”. Script files that are stored in this folder can be quickly accessed through the scripts toolbar button or the scripts status bar panel.

Printer Setup

Opens the printer setup dialog.

Print Program

Sends the current program contents to the printer.

Print Viewport

Sends the current viewport to the printer. For printing, the view is drawn without the background.

Clear Recent Files

Clears the list of recently opened files. The last 20 files that were opened are stored in the recent files list for easy re-opening. The recent files list is located on the toolbar next to the ‘Open’ icon. Clicking on the arrow opens the list, and clicking on a file in the list will open the file. If there are unsaved changes, you are prompted to save before opening the selected file. If the file no longer exists, you receive an error message and the file is removed from the list. When a file is opened from the recent files list, that file is moved to the top of list.

Exit

Exits the application. If any unsaved changes have been made to the current program, you are prompted to save it before the application closes.

Edit Menu

Undo

Undoes the last edit to the program.

Redo

Redo the last undone change.

Cut

Cuts the selected text from the program and places it on the clipboard.

Copy

Copies the selected text from the program to the clipboard.

Paste

Pastes the current clipboard contents into the program at the current cursor location.

Select From

Use this tool to mark the current cursor location in the file. Then move the cursor to the end of the area you want to select and use the “Select To” tool to select the desired region.

Select To

Selects from the marked cursor location to the current cursor location. See Select From

Select All

Selects the entire contents of the edit window.

Find

Find text in the program. If any text is selected, it will be entered into the Find What field of the find dialog.

Find Next

Repeats the last Find operation. If the end of the program is encountered, you are asked if you want to repeat the search from the beginning.

Replace

Replaces text in the program. The selected program text is replaced by the text in the Replace With field.

Jump To Line Number

Jumps to the designated file line number.

Highlight Selected

This tool is used to locate the selected portion of your program on the viewport. To use this tool, select the portion of the program you want to locate and then click the Highlight Selected menu item. The entities on the viewport that correspond to the selected program will then be highlighted.

Highlight and Zoom to Selected

This tool is used to locate the selected portion of your program on the viewport. To use this tool, select the portion of the program you want to locate and then click the ***Highlight and Zoom to Selected*** menu item. The entities on the viewport that correspond to the selected program will be highlighted and the viewport will zoom to fit these entities.

Font

Opens the font dialog. The selected font and color settings may be applied to the entire program or just the selected text. If any part of the program text is selected, only the selected text will be affected. Otherwise, the settings will be applied to the entire program.

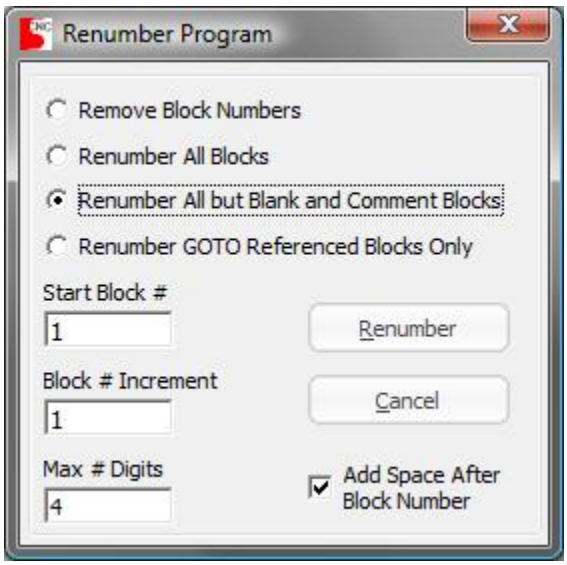
Colorize

Applies the address color settings to the current file. The color settings can be changed on the address color settings dialog found under the menu ***Setup | Address Colors***.

Format Menu

Renumber Blocks

Opens the Renumber Program dialog.



This dialog let's you set up how you would like your program blocks to be numbered.

Remove Block Numbers
This option will remove block numbers from the program, with the exception of block numbers that are being referenced.

Renumber All Blocks
This option will renumber all blocks including blank and comment lines.

Renumber All but Blank and Comment Blocks
This option will renumber all blocks except blank and comment lines.

Renumber GOTO Referenced Blocks Only
This option will renumber only those blocks that are being referenced by commands in other blocks. This includes the macro keyword "GOTO", local subprogram calls and canned cycles that reference block numbers.

The "Start Block #", "Block # Increment" and "Max # Digits" settings define how the program will be numbered.

Remove Block Numbers

This tool will remove all block numbers from the program except for block numbers that are referenced by a “GOTO” macro statement, a subprogram call or other program blocks.

Add Spaces

This tool inserts spaces between letter addresses and macro keywords to improve readability of the program. This tool does not affect text between parentheses (program comments).

```
(Testing)
G0X0Y0Z1
IF[#1GT[SQR[#2+#3]]]GOTO50
```

This becomes:

```
(Testing)
G0 X0 Y0 Z1
IF [#1GTSQR[#2+#3]] GOTO 50
```

Remove Spaces

This tool removes all spaces from the program except for comments. Comments are not affected.

Remove Trailing Spaces

This tool removes any spaces from the end of the block.

Remove Blank Lines

This tool removes blank lines from the program.

Remove Comments

This tool removes comments from the program. Comments are anything that is enclosed in parentheses ( ). Comments may also be enclosed in square brackets as long as they do not follow a letter address.

Remove Garbage

This tool fixes non-standard end of line characters. Any combination of carriage return and line feed characters are converted to standard CR/LF format. This tool also converts TAB characters (ASCII 09) to spaces and removes all other control characters. It also converts any extended characters (ASCII 128 thru 255) to standard characters.

Add Comment Chars

This tool adds comment characters ( ) to the beginning and end of each selected line in the program. If nothing is selected, then no changes are made.

Remove Comment Chars

This tool removes comment characters ( ) from each selected line of the program.

Add Block Skip Chars

This tool adds the block skip character "/" to the beginning of each selected line in the program. If nothing is selected, then no changes are made.

Remove Block Skip Chars

This tool removes the block skip character "/" from each selected line of the program.

Convert to All Caps

This tool converts the program to all caps, including comments.

Remove Redundant Endpoints

Many controllers only require axis endpoints that are changed from the previous block. This tool reduces program size by removing axis endpoints that are specified but not changed from the previous block.

Tools Menu

Display Precision

Selects the number of decimal places for coordinate display and G-Code editing tools. The display precision may be set to between 3 and 6 decimal places.

Macro Translator

This tool will execute a variable macro and translate it into standard G-Code blocks. Any blocks that contain variable commands will be output with the variables replaced with their current values. This will expand the macro program into an equivalent longhand G-Code program. This process will also expand any macro program loops and subprograms.

Before the macro is executed, you can set any required variable values on the translator dialog window. You may define the required variables right in the macro program. Comment blocks at the head of the macro are used to define the required variables and are displayed on the translator dialog when the macro program is selected.

As an example, the Bolt Circle cycle asks for the X, Y center location, radius, start angle and number of holes. After this information is entered, the cycle is executed and the resulting G-Code is inserted into your program. There are a number of cycles included with NCPlot, but you may also add your own.

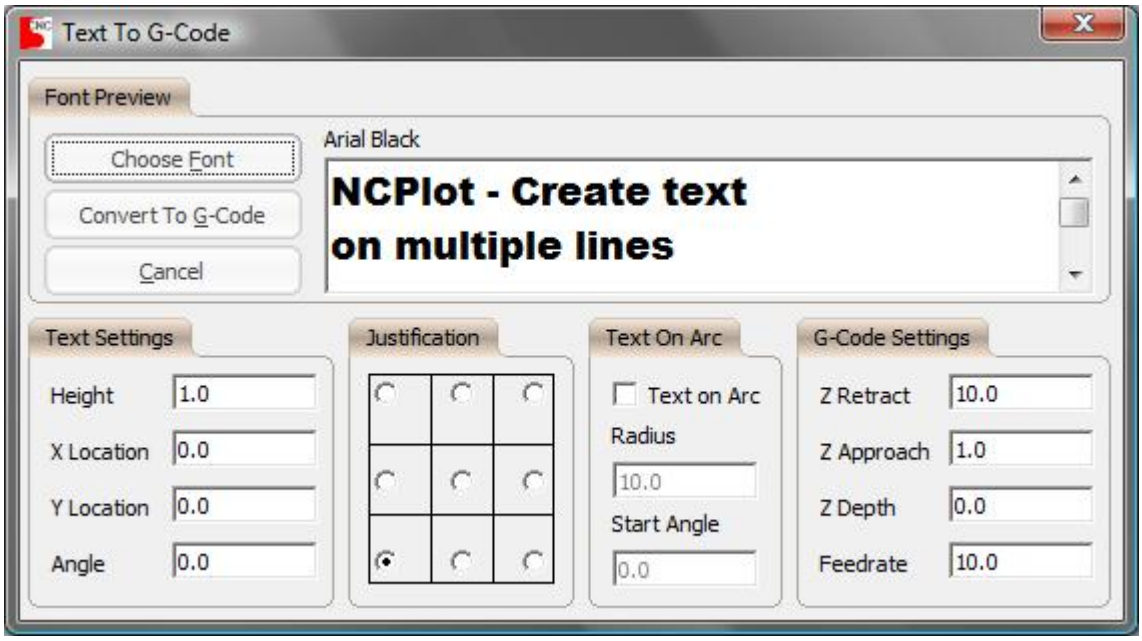
The header format for using macro programs with this tool is very simple:

```
(Bolt hole circle)           Name of macro or cycle

(#24 Center X)               Required variables and descriptions
(#25 Center Y)
(#4  Bolt circle radius)     These descriptions appear on the
(#5  Angle of first hole)    translator dialog.
(#6  Number of holes)
```

Text to G-Code

This tool generates G-Code that follows the outline of the entered lettering. The lettering may be in any font that you have installed on your computer.



Font Preview

Use the ***Choose Font*** button to select the desired font style and enter the text to convert into the font preview window. When all other settings have been made, use the ***Convert To G-Code*** button to convert the displayed text to G-Code. When entering text to convert, you may create multiple lines by pressing CTRL-ENTER to start a new line. This dialog does not close after the conversion process is complete, but the graphics view is refreshed so that you can immediately see the results of the conversion. If the resulting code is not what you want, simply undo the added code and make your changes.

Text Settings

These settings define the resulting size, location and orientation of the converted text. When the ***Text on Arc*** option is checked, the X and Y location settings become the X and Y arc center location and the ***Angle*** setting is disabled.

Justification

This setting allows you to define which location on the text will correspond to the X and Y location settings. This makes it much easier to center or align the output text. This setting is disabled when the ***Text on Arc*** option is checked.

Text On Arc

When checked, this option will create the output text on an arc. You just set the desired X and Y center of the arc under the ***Text Settings*** and then enter the radius and start angle of the arc.

G-Code Settings

These settings are used for the resulting G-Code output.

Convert Selected to G-Code

This tool makes it possible to create G-Code from any entity on the viewport. Since the viewport is made up of entities created from G-Code, this applies mainly to entities created by the Calc tools. To use this tool, first select all the entities that you want to convert to G-Code. When this tool is picked, a dialog appears that allows you to define the G-Code settings that will be used for the conversion. The selected entities are then chained together before being converted to G-Code.

Mirror

This tool changes the program endpoints to create a 'Mirror image' of the original program. The program can be mirrored in either the X axis, Y axis or both. You can also set the program coordinate that will act as the center of the mirror axis.

Rotate

This tool rotates program endpoints in the XY plane. Simply set the X and Y center of rotation and the desired rotation angle. Please note that if your program contains arcs in the G18 or G19 planes, the resulting program will probably not function correctly. By definition, these arcs must be in the ZX or YZ planes and rotating them will create illegal arcs.

Shift

This tool adds a specified shift amount to each of the axes.

Scale

This tool applies the specified scale factor to the program.

Address Adjustments

This tool allows you to apply math operations to specific program addresses. You can add, subtract, multiply or divide the program values by a given adjustment amount. The Value Format string determines the format of the resulting values. If no adjustment is applied (by specifying * 1 or + 0 as the operation) the specified addresses will simply be reformatted. This makes it possible to change the value format of any address in the program.

Address Replace

This tool allows you to replace any program address with another. This is useful for changing the address letter for a 4th axis or when converting from one program format to another. This tool does not affect comments so it's smarter than a simple find / replace operation.

Address Remove

This tool removes the selected addresses and their values from the program. This also does not affect comments.

Address Swap

This tool allows you to swap any two program addresses. This tool does not affect comments so it's smarter than a simple find / replace operation.

Convert Arc Centers to ABS

This tool converts arc centers specified with incremental I / J / K to absolute. This tool also turns on the Absolute Arc Centers option of the Draw menu.

Convert Arc Centers to INC

This tool converts arc centers specified with absolute I / J / K to incremental distance from the arc start point. This tool turns off the Absolute Arc Centers option of the Draw menu.

Convert Arc R to I/J/K

This tool calculates the center point for R specified arcs and replaces 'R' with I / J / K values in either incremental or absolute. Incremental or absolute is selected via the Draw menu and is indicated by the check next to the Absolute Arc Centers menu item.

Convert Arc I/J/K to R

This tool calculates the radius of the arc and replaces I / J / K values with an R value. For arcs less than or equal to 180 degrees, the R value is positive and for arcs greater than 180 degrees, the R value is negative. This tool will not convert arcs that result in a full circle. If any arcs could not converted for this reason, a message will indicate how many arcs could not be converted.

Break Arcs Into Lines

This tool will break arcs into line segments. Any G02/G03 blocks are replaced with a sequence of G01 moves that approximate the arc. You will be prompted for a maximum deviation distance that determines how closely the line segments will follow the arc. This also determines how many line segments are required to approximate the arc. This tool will also break helical arcs up to 360 degrees.

Show Part Info

This tool shows a window that displays the extents and estimated run time of the loaded file. Also displayed is the maximum arc error. This value is the greatest encountered difference in radius between an arc start and end point.

You can enter your machine's rapid rate and approximate tool change time to more accurately estimate the required cycle time.

Show Entity Info

This tool shows a window that displays information about an entity on the viewport. This window pops up automatically whenever an entity is picked on the viewport. The auto pop-up can be disabled on the preferences page.

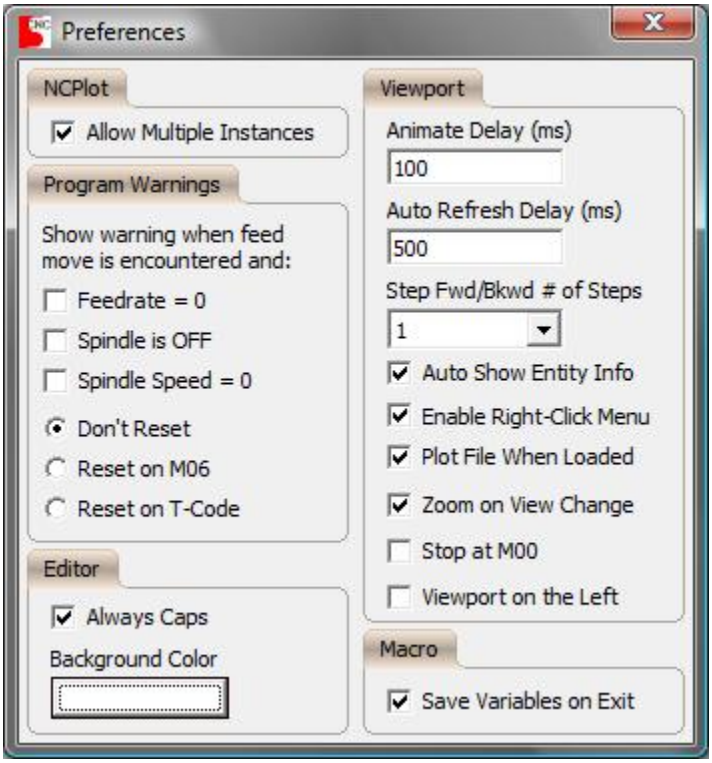
Show Modal State

This tool reports the active modal state at the current cursor location. The graphics backplot is refreshed and a window appears that contains the active modal information at the current line. This information includes all active G-Codes, the active values of all other address values and any active canned cycle.

Setup Menu

Preferences

Opens the Preferences dialog. This dialog contains default behavior settings for the editor and viewport.



NCPlot

Allow Multiple Instances – When checked, you may have more than one copy of NCPlot open at a time.

Editor

Always Caps – When checked, this setting forces all keystrokes in the edit window to be in upper case.

Background Color – This button allows you to change the background color of the edit window.

Macro

Save Variables on Exit – Check this setting if you want common variables to be saved on exit.

Viewport

Animate Delay – This number is a delay time in milliseconds. When the animate button is pressed the program is drawn one step at a time with this amount of delay time between steps.

Auto Refresh Delay – This setting determines how long the viewport will wait before refreshing after a change is made to the program. This setting is ignored if the ***Auto Refresh Viewport*** setting is not enabled.

Step Fwd/Bkwd # of Steps – This setting determines how many entities are drawn each time the program is stepped forward or backwards.

Auto Show Entity Info – When checked the entity info window is automatically opened when an entity on the viewport is clicked. Likewise, when all entities are unselected, the window will close.

Enable Right-Click Menu – When checked, enables the right click popup menu.

Plot File When Loaded – When checked, enables automatically backplotting a file when it is loaded into the editor.

Zoom on View Change – When checked the viewport will re-zoom to the part extents anytime a new view orientation is selected.

Stop at M00 – When checked the backplotter will pause and display a message each time it encounters an M00 in the program. When paused the backplot may be cancelled or resumed.

Viewport on the Left – This option will exchange the viewport and editor displays.

Program Warnings

When checked, these settings will enable warning messages that indicate when a feedrate move is encountered in the program and no feedrate or spindle command has been programmed.

Address Colors

Opens a dialog that allows you to setup how the program is colored when the Colorize tool is applied to it. To change the color settings, double-click the colored box next the address letter you want to change. A color picker dialog opens and you can select the color for that address. To see your changes, select the menu ***Edit / Colorize***.

Subprograms

Opens the sub program assignment window. This window is used to tell NCPlot where to find any sub programs that are called from your G-Code program using M98 or G65 codes.

M98 and G65 Handling

These settings determine what action NCPlot will take when it encounters a subprogram command in the program:

Error if sub not found – If the commanded subprogram cannot be found, NCPlot will display an error and program processing ends.

Ignore if sub not found – If the commanded subprogram is found, it will be plotted, otherwise it will be ignored.

Ignore all – All subprogram calls will be ignored.

Default Search Path:

If you have many sub programs, or sub programs that you use often, you can keep them together in the same folder and set the default search path to point to this folder. In order for NCPlot to find it, the file name must begin with the letter 'O' followed by the program number. Any extension may be used, for example: O1.txt is a valid file name.

Associations:

You may also assign files at random to individual program numbers. To do this, click the 'Add' button. You will be prompted for the program number, enter the number that follows the 'P' address in the M98 or G65 block. You will then be presented with the file browse dialog, use this to select the file that contains the G-Code for the entered program number.

DXF Layer Setup

Opens the DXF Layer Setup dialog so that stored layer data may be modified without actually opening a drawing file.

Toolbars

The toolbars submenu allows you to turn off or on the various toolbars. A check mark next to each toolbar name indicates that the toolbar is on.

Machine Configuration

Opens the machine configuration dialog. This dialog is used to customize NCPlot to accurately emulate the way your control handles certain G-Code functions.

For more information, see ***Setting up a Machine Configuration***.

Calc Menu

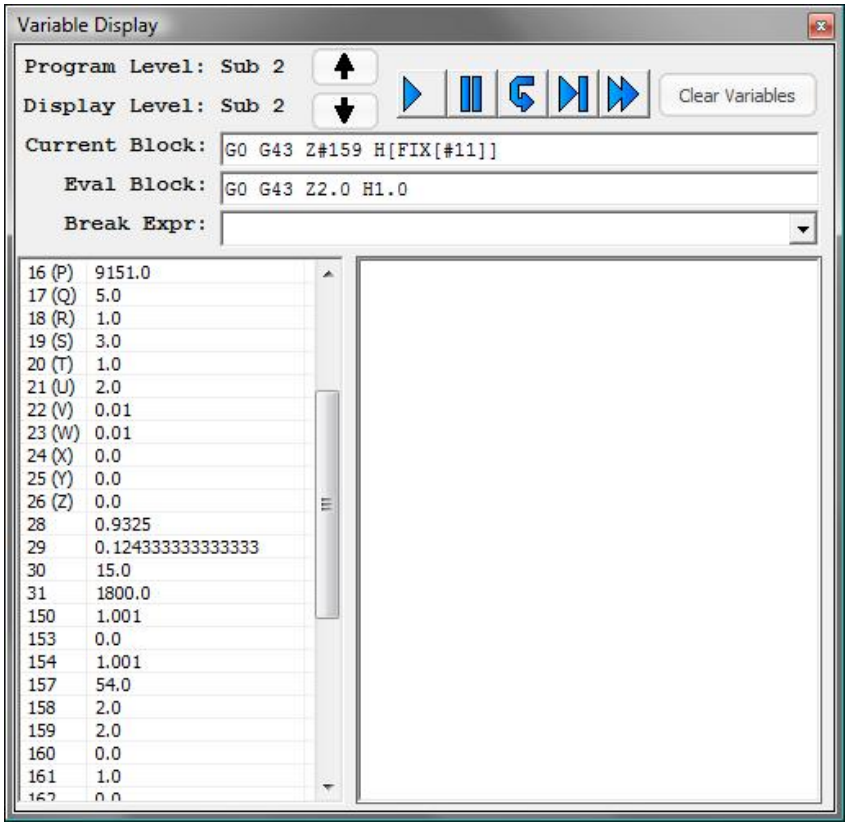
Expression Calculator

This calculator solves mathematical expressions. There are 15 functions and 9 operators and allows unlimited bracket groupings. The expression and the result are added to the result window for easy referencing back to any earlier expression. A typical expression would look something like:

```
SQR[[6-2]^2+[9-11]^2]
=4.47213595499958
```

Show Variables

The **Variable Display** window is an indispensable tool for anyone writing macro programs. This window not only displays the current state of program variables, it enables you to execute macro programs one block at a time. Local variables may be displayed for each of the subprogram levels. This window also allows you to follow program execution into subprograms not contained in the loaded program file. This program stepping differs from the main window. When stepping from the variable display window, the program is executed as it is stepped so you can see the results of variable commands. On the main window, the program is fully executed before stepping begins, so the state of the variables reflect the end result of the program.



This window provides several important tools for debugging macro programs. Several controls are combined into a toolbar:

Macro Animate – Executes the program one block at a time with a timed interval in between blocks. The variable display is updated after each block.

Pause – Stops a program being executed.

Macro Step – Executes one block of the program and updates the variable display.

Run to Cursor – Executes the program at high speed until it reaches the block that the cursor is on.

Run to Break – Executes the program at high speed until the break expression becomes True. The break expression is tested after every program block.

Clear Variables – Allows clearing of all local and common variables to empty.

The ***Current Block*** display shows the next block to be executed as it appears in the program. The ***Eval Block*** display shows the same block with any expressions replaced with their evaluated values. This allows you to see the resulting block before it is executed.

The local variable display level may be changed using the up and down arrow buttons. This allows checking local variables for each of the 6 subprogram levels.

Right-clicking the variable display will pop up a menu with these options:

Add System Variable – This option allows adding any system variable to the variable display. You will be prompted for the variable to add, simply enter the variable number you want to add. Once added it will remain until you remove it.

Remove System Variable – This option will remove the selected system variable from the variable display. If the selected variable is not a system variable, nothing is removed.

Copy Variables to Clipboard – Copies the current variable list to the clipboard as text. You can then paste it into any text editor.

Print Variables – Sends the current variable list to the default printer.

Windows™ Calculator

This menu item launches the Calculator application.

Select Entities

These selection tools make it easy to select logical groups of entities on the viewport. This menu contains three submenus:

Select Chain – This tool is useful for selecting 2D profiles. Just select one entity that is part of the profile, then select this tool. All entities that are connected to the selected one and at the same Z depth are selected automatically.

Select by Z – This tool will go through the entire backplot and select all entities that are at the same Z depth as the currently selected entity. This makes it easy to select all profiles that are at a given depth.

Select Arcs by Radius – This tool will select only arc entities that are the same radius as the currently selected arc. You could then create points at the center of each selected arc and use these points to create a drill program.

Offset

This tool will create new entities at the given offset distance from the selected entities. To use this tool, select the desired entities by either clicking them on the viewport or by selecting lines in the program and using the Highlight Selected tool on the edit menu. After the desired entities are selected, go to the menu ***Calc | Offset***. You will be prompted for the desired offset distance. Entering a positive distance will offset to the left, while entering a negative distance will offset to the right. Once created, the new entities will stay on the viewport until it is refreshed with the plot button or a new file is loaded.

Intersect

This tool will find the intersection points between two entities. To use this tool, first select the two entities you want to find the intersection points for. Then select the Intersect tool. The point or points are calculated and displayed on the viewport. You also get a message on the status bar that tells you the coordinates of the points. This tool will solve for Line / Line, Line / Arc, and Arc / Arc intersections even if the selected entities do not visibly touch each other.

Blend Radius

This tool will create an arc of the specified radius that is tangent to two lines. To use this tool, first select two intersecting lines. After selecting this tool, you will be prompted for the desired blend radius. Enter the radius and an arc will be created on the viewport that is tangent to both lines at the given radius.

Point at Center

This tool will create points representing the center of each of the selected entities.

Measure

This tool will report the X, Y, Z and overall distance between the endpoints of the two selected entities.

View Menu

Top View

The **Top** view mode displays the axes with the X+ direction toward the right side of the screen and the Y+ toward the top. This view mode is only available when the Mill machine type is active.

Bottom View

The **Bottom** view mode displays the axes with the X+ direction toward the left side of the screen the Y+ toward the top. This view mode is only available when the Mill machine type is active.

Front View

The **Front** view mode displays the axes with the X+ direction toward the right side of the screen and the Z+ toward the top. This view mode is only available when the Mill machine type is active.

Back View

The **Back** view mode displays the axes with the X+ direction toward the left side of the screen and the Z+ toward the top. This view mode is only available when the Mill machine type is active.

Left View

The **Left** view mode displays the axes with the Y+ direction toward the left side of the screen and the Z+ toward the top. This view mode is only available when the Mill machine type is active.

Right View

The **Right** view mode displays the axes with the Y+ direction toward the right side of the screen and the Z+ toward the top. This view mode is only available when the Mill machine type is active.

Isometric

The **Isometric** view mode displays the X, Y and Z axes in a 3D view. This view mode is different from all others because it can be rotated to show the part from any angle. This view mode is only available when the Mill machine type is active.

Front Turret View

The **Front Turret** view mode displays the axes with the X+ direction toward the bottom of the screen and the Z+ direction toward the right. This represents a machine where the tool approaches the work from the front of the machine. This view mode is only available when the Lathe machine type is active.

Back Turret View

The **Back Turret** view mode displays the axes with the X+ direction toward the top of the screen and the Z+ direction toward the right. This represents a machine where the tool approaches the work from the back of the machine. This view mode is only available when the Lathe machine type is active.

Vertical Left

The **Vertical Left** view mode displays the axes with the X+ direction toward the left side of the screen and the Z+ direction toward the top. This represents a vertical lathe where the tool approaches the work from the left side of the machine. This view mode is only available when the Lathe machine type is active.

Vertical Right

The **Vertical Right** view mode displays the axes with the X+ direction toward the right side of the screen and the Z+ direction toward the top. This represents a vertical lathe where the tool approaches the work from the right side of the machine. This view mode is only available when the Lathe machine type is active.

Pan

After selecting this tool, use either mouse button to drag the viewport to the desired view center. This tool can also be activated by pressing the ‘P’ key when the viewport is active.

Zoom Extents

Sets the view center and zoom size to fit the entire program in the viewport. This tool is also activated by pressing the ‘X’ key when the viewport is active.

Zoom All

Sets the view center and zoom size to fit the entire program in the viewport. This tool is different from the **Zoom Extents** tool in that this tool will not include rapid motions when fitting the view. This tool is also activated by pressing the 'A' key when the viewport is active.

Zoom Selected

Sets the view center and zoom size to fit only the selected entities. This tool is also activated by pressing the 'S' key when the viewport is active.

Zoom In

Enlarges the view size while keeping the current view center. This tool is also activated by pressing the 'C' key when the viewport is active.

Zoom Out

Reduces the view size while keeping the current view center. This tool is also activated by pressing the 'V' key when the viewport is active.

Zoom Window

Using the left mouse button, drag a box around the desired view area. This tool is also activated by pressing the 'Z' key when the viewport is active.

Lock Vertical Rotation

When this item is checked, the isometric viewport will only rotate around the viewport vertical axis. This can be overridden by holding the Ctrl key while rotating the viewport. After rotating the view and releasing the Ctrl key, the viewport will maintain the new horizontal rotation.

Draw Menu

Animate

Draws the loaded file while highlighting the corresponding blocks in the program window. There is a delay between steps so you can see the motion that is taking place. This delay can be set on the preferences dialog.

Pause

Pauses an animation in progress. Once paused, pressing Animate again will resume the program.

Rewind to Beginning

Clear the backplot graphics and sets the cursor back to the first motion block in the program.

Forward to End

Draws the backplot graphics from the current start position up to the end of the program and sets the cursor to the last motion block in the program.

Step Forward

Draws the next motion block in the program and highlights the corresponding block in the program window.

Step Backward

Un-draws the current motion block and highlights the previous block in the program window.

Refresh Plot

Refreshes the viewport with the current contents of the program window.

Plot to Cursor

Draws the file from the beginning to the current cursor location.

Start at Cursor

This item clears the viewport and sets the current program step point to the line in the program that the cursor is on. You can then animate, step forward or step backward from this point.

Plot from Cursor

Draws the file from the current cursor location to the end.

Plot Selected

This tool draws just the selected portion of the program.

Auto Refresh Viewport

This menu item is an on/off toggle setting that enables automatic refreshing of the viewport graphics after program edits are made. A check mark next this menu item indicates that auto refreshing is enabled.

When enabled, this feature will automatically refresh the graphics whenever a change is made to the program. There is a setting on the ***Preferences*** dialog that determines how long NCPlot will wait between a change being made and refreshing the graphics.

Axis Lines

Displays two or three (depending on the view) intersecting lines that indicate where 0,0,0 is on the viewport. The actual location of the axis lines depends on the selected submenu item. A check mark indicates which item is selected.

Off – No axis lines are displayed

Machine Zero – The axis lines represent the Machine Zero location

G54-G59 Work Zero – The axis lines represent the select Work Zero location

Show Rapid Moves

When checked, this option enables drawing of G00 rapid moves.

Show Ticks

When checked, this option enables drawing of tick marks at the endpoints of rapid moves.

Show Marker

When this menu item is checked, an arrow is drawn on the viewport which indicates the current plot endpoint. This marker also displays the coordinate values of the indicated endpoint.

Show Plunge Moves

When checked, this option enables drawing of G1 moves in the Z- direction.

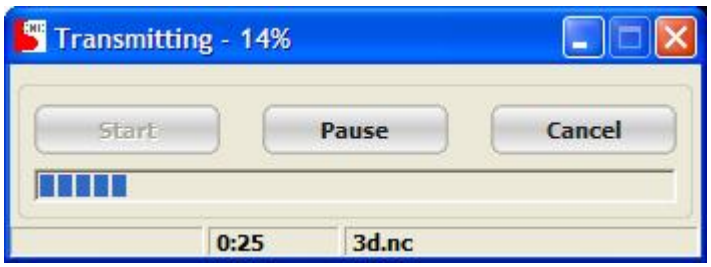
Absolute Arc Centers

When checked, this option specifies how I / J / K specified arc centers are drawn. This option also affects the results of the arc conversion tools. This setting may also be changed on the Machine Configuration dialog under the Control Options tab.

DNC Menu

Send

Sends the entire contents of the edit window to the DNC tool. This tool buffers the data to be sent so that you can continue to work in NCPlot while the transfer is taking place.



When the DNC tool is open, click “Start” to begin the transmission, “Pause” to pause the transmission, or “Cancel” to abort. There are three status panels at the bottom of the window. The left panel displays the word “Waiting” whenever an XOFF character is received to indicate that the transmission has paused for a handshaking signal. The middle panel displays the elapsed transmission time, and the right panel displays the name of the program being sent.

Send Selected

Similar to the ***Send*** tool, but only the selected program text is sent to the DNC tool.

Send File

Similar to the ***Send*** tool, but allows you to browse for and send a file other than the current edit file.

Receive

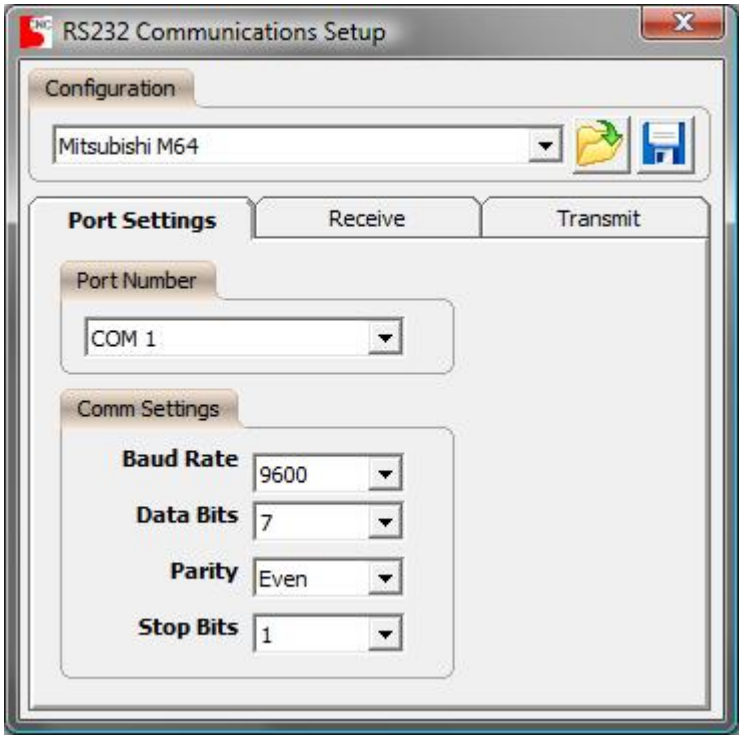
Opens the COM port for receiving. Once opened, any received data is inserted into the edit window at the current cursor location.

Receive as New

Similar to the ***Receive*** tool, but closes the current edit file before opening the port for receiving.

COM Setup

Opens the RS232 setup dialog.



Configuration – The currently active configuration name.

The remaining settings are organized into three tabs: ***Port Settings***, ***Receive*** and ***Transmit***.

The ***Port Settings*** tab contains the basic port setup such as the COM port number, baud rate, data bits, etc.

Port – Selects the active COM port number.

Comm Settings – Selects the desired baud rate, number of data bits, parity and number of stop bits.

The ***Receive*** tab contains timeout settings for receiving data from the machine.

Timeout before receive – After the COM port has been opened and nothing is received within this set amount of time, the COM port will close. If this setting is zero then NCPlot will wait indefinitely for data to be received.

Timeout after receive – Once data has been received, the COM port will close after nothing more has been received for this set amount of time.

The ***Transmit*** tab contains settings for transmitting data to the machine.

Handshaking – This setting is currently ignored, the XON/XOFF handshaking method is always used.

Remove spaces while transmitting – When this setting is enabled any spaces in the program will not be transmitted to the machine.

Wait for XON before transmitting – Enabling this setting will cause NCPlot to wait for an XON character to be received from the machine before beginning transmission. This allows you to start the data transmission from NCPlot and then go to the machine to begin receiving.

Help Menu

Quick Start Reference

This menu item will open the help file to the “Getting Started” topic.

Macro Programming Reference

This menu item will open the help file to the “Macro Programming Reference” topic.

Scripting Reference

This menu item will open the help file to the “Scripting Reference” topic.

Release Notes

Opens the release notes. This is a notepad document that details the most recent changes to NCPlot.

About NCPlot

Displays a window showing the NCPlot version number and your license status.

NCPlot on the Web

This menu item will open your default web browser to the NCPlot home page www.ncplot.com.

Ordering NCPlot

This menu item will open your default web browser to the NCPlot online ordering page.

Appendix A

Supported G & M Codes

NCPlot can plot programs that are in RS274 standard G and M code format. This table describes the codes that are supported by NCPlot. Unrecognized codes will be ignored by the backplotter. There are three distinct G-Code sets that are supported: Mill, Lathe Format A and Lathe Format B. In addition to the G-Codes listed here, NCPlot can be configured to simulate additional codes by providing an appropriate macro for each additional code.

Mill G-Codes

G00	Rapid motion

NCPlot supports displaying rapid motion as interpolated (straight line), dog leg or Z first / last.

G01	Feed motion

NCPlot supports G01 corner rounding and corner chamfering. Including an R value in a G01 block will create a radius that is tangent to the lines created by the current block and the next block. Including a C value in a G01 block will create a chamfer between the current block and the next block.

G02	Clockwise arc
G03	Counterclockwise arc

The G02 and G03 arc commands support G17, G18 and G19 modes, absolute and incremental I / J / K specified center point, R specified center point, and helical interpolation.

G15	Polar Coordinate System Off
G16	Polar Coordinate System Command
G17	XY Plane Selection
G18	ZX Plane Selection
G19	YZ Plane Selection
G28	Reference Point Return
G50	Coordinate System Scaling Off
G51	Coordinate System Scaling
G50.1	Mirror Image Off
G51.1	Mirror Image On
G52	Local work shift
G53	Position Relative to Machine Zero
G54.1 P1 – P300	Extended Work Offsets
G54 P1 – P300	Extended Work Offsets
G54	Work Offset 1
G55	Work Offset 2
G56	Work Offset 3
G57	Work Offset 4
G58	Work Offset 5
G59	Work Offset 6
G65	Macro Sub program call

```
G68                     Coordinate System Rotation
G69                     Coordinate System Rotation Off
G80                     Cancel canned cycle
G70 – G89               Canned cycles
G90                     Absolute coordinate system
G91                     Incremental coordinate system
G92                     Coordinate system set
G98                     Canned cycle initial point return
G99                     Canned cycle R point return
```

NCPlot does not handle the following G-Codes internally. Instead, they are simulated by external macro programs. This allows them to be customized for a particular control.

```
G12                     Clockwise circle mill
G13                     Counterclockwise circle mill

G34                     Bolt circle canned cycle
G35                     Holes on line at angle canned cycle
G36                     Holes on arc canned cycle
G37.1                   Grid pattern canned cycle

G70                     Bolt circle cycle
G71                     Bolt hole arc cycle
G72                     Holes on line at angle cycle
```

Lathe Format A

```
G00                     Rapid motion
G01                     Feed motion
G02                     Clockwise arc
G03                     Counterclockwise arc
G28                     Reference Point Return
G50                     Coordinate system set
G52                     Local coordinate system set
G53                     Position relative to machine coordinates
G54.1 P1 - P300         Extended work offsets
G54 P1 – P300           Extended work offsets
G54                     Work Offset 1
G55                     Work Offset 2
G56                     Work Offset 3
G57                     Work Offset 4
G58                     Work Offset 5
G59                     Work Offset 6
G65                     Macro Sub program call
G70 - G89               Canned cycles
G80                     Canned cycle cancel
G90                     Turning cycle
G92                     Thread cutting cycle
G94                     Facing cycle
G96                     Spindle CSS mode
G97                     Spindle RPM mode
G98                     Feed per minute mode
G99                     Feed per revolution mode
```

Lathe Format B

G00	Rapid motion
G01	Feed motion
G02	Clockwise arc
G03	Counterclockwise arc
G28	Reference point return
G52	Local coordinate system set
G53	Position relative to machine coordinates
G54.1 P1 - P300	Extended work offsets
G54 P1 – P300	Extended work offsets
G54	Work Offset 1
G55	Work Offset 2
G56	Work Offset 3
G57	Work Offset 4
G58	Work Offset 5
G59	Work offset 6
G65	Sub program call
G90	Absolute coordinate system
G91	Incremental coordinate system
G92	Coordinate system set
G94	Feed per minute mode
G95	Feed per revolution mode
G70 - G89	Canned cycles
G80	Canned cycle cancel
G96	Spindle CSS mode
G97	Spindle RPM mode
G98	Canned cycle initial point return
G99	Canned cycle R point return

M-Codes

M02, M30	End of program

The spindle control M-Codes are used to provide warning messages when a program commands feed motion without the spindle running. These messages may be enabled on the ***Preferences*** dialog.

M03	Spindle forward
M04	Spindle reverse
M05	Spindle stop
M13	Spindle forward & coolant ON
M14	Spindle reverse & coolant ON

The mirror image M-Codes may be changed to match your control. This may be set on the ***Machine Configuration*** dialog.

M87	X axis mirror ON
M88	Y axis mirror ON
M89	X & Y axis mirror OFF

Multiple subprogram call formats are supported. Use the ***Machine Configuration*** dialog to select the format that your control recognizes. For details about choosing a subprogram format, refer to the ***Plotting Subprograms*** topic.

```
M97 P                   Local subprogram call
M98 P L
M98 P K
M98 Pxxxxyyyy
M98 P H L
M98 O

M99 P                   Return from sub program
```

When M99 is commanded in a sub program, the P value specifies the block number to return to. When commanded in the main program, the P value causes execution to jump to the specified block number.

Macro Keywords

In addition to G&M codes, NCPlot also recognizes variables and commands in the Fanuc Macro B format. Refer to the ***Macro Programming*** section for details about the supported commands.

Appendix B

Canned Cycles

The backplotting of canned cycles in NCPlot is not hard coded. These cycles are external macro programs that may be customized to match your particular machine. The canned cycle macros are located at:

\Program Files\NCPlot\Config*config name*\

The Config folder contains the machine configuration files along with a corresponding sub folder for each. These sub folders contain the canned cycle macro programs. Every machine configuration uses it's own set of canned cycle macros, which means that you can have different canned cycles for each machine configuration.

The G-Codes that are handled as external macros by default are:

```
G70 – G79, G81 – G89          Standard canned cycles

G12                           Clockwise circle cutting
G13                           Counterclockwise circle cutting
```

The G34 through G37.1 cycles are included in the default mill configurations.

```
G34                           Bolt circle cycle
G35                           Holes on line at angle cycle
G36                           Holes on arc cycle
G37.1                   Grid pattern cycle
```

The G70 through G72 bolt pattern cycles are included in the HAAS mill configuration.

```
G70                           Bolt circle cycle
G71                           Holes on arc cycle
G72                           Holes on line at angle cycle
```

Additional cycles may be added to simulate G-Codes not supported by NCPlot. Open the ***Machine Configuration*** dialog to the ***Control Options 2*** tab and enter additional G-Codes in the ***G-Code Macro*** list. The active configuration folder must contain a corresponding program file to handle the simulation of the additional codes.

The macro program file name is the corresponding G-Code value times 10. For example, the file name for G81 is G810.PRG. This is to accommodate decimal G-Codes such as G37.1. You can add cycles for G-Codes that fall within these ranges by simply adding an appropriately named G-Code program file to the desired configuration folder.

Appendix C

What is Macro Programming

The macro language is a programming language that gives the CNC programmer the ability to write very flexible programs. This is done through the use of variables, mathematical expressions and program flow control statements. The macro language combined with standard G-code programming can create reusable programs much like canned cycles. These programs can do many useful things like custom pocketing or automatic tool measurement. As they say, you are limited only by your imagination.

This document describes how NCPlot processes macro programs and may differ from your specific control.

What are Variables

Variables are the heart of macro programming. Variables are like numbered storage units, each of which can either be empty or contain a number. When they contain a number, the variables can be used in place of almost any numeric value in your G-code program.

Variables are designated with the "#" symbol and are followed by a number or expression that designates the variable number. Variables can used as the value following any letter address except “N”.

Examples of how variables can be used:

`G1 X#100 Y#101 F10.`	In this example of a feed move, the X and Y endpoints are determined by the values contained in the variables #100 and #101.
`G#100 X#101 Y#102`	In this example, the G-code to be executed is determined by the value contained in variable #100.

What makes them such a powerful tool is the fact that new values can be assigned to variables by your program.

Here is an example of assigning a value to a variable:

`#100=10.0`	The value 10.0 is stored into variable #100. Once this has been set, the value 10.0 will be used in place of #100.
`G1 X#100`	This command is then equivalent to `G1 X10.0`

By simply changing the value stored in this variable, you can make the same program do different things.

Multiple variables may be assigned in the same block. However, variable assignments must be on a block by themselves and may not be combined with G-code blocks or other macro statements.

Examples:

`#101=1 #102=2 #103=3`	This is valid
`IF [#1EQ0] THEN #105=5`	This is the only macro statement where variables can be assigned.
`G01 X#101 #102=4`	This is not valid
`WHILE [#1NE0] DO1 #104=4`	This is not valid

You, as the programmer, should decide the meaning of your variables before you begin writing. For example, you may want variable #100 to be used to specify a pocket width, or you may want variable #250 to specify a corner radius. It is completely up to you what the variables mean, but it is good programming practice to decide before you begin writing.

What are Expressions

Expressions are formulas that mathematically combine values to get a resulting number. The result of an expression is then assigned to a variable or used in a conditional statement.

Example of using an expression to assign a value to a variable:

`#100 = #101+1` The expression is the part following the "=" sign and is "#101+1". So 1 is added to the contents of #101, and this value is then stored in variable #100.

Expressions may use any combination of operators, functions and comparisons. If no brackets are used, the values are calculated in the standard arithmetic order. That is, multiply and divides are performed first, followed by addition and subtraction.

Examples of expressions:

```
#100 = 10 + 2 * 3 - 5          The value 11 is stored in variable #100
#101 = 8 * 3 – 3 * 4           The value 12 is stored in variable #101
```

Operators are basic mathematical operations and include:

`+`	Addition
`-`	Subtraction
`*`	Multiplication
`/`	Division
`^`	Raised to the power of
`MOD`	Modulus, the remainder of a division operation
`XOR`	Bitwise XOR
`OR`	Bitwise OR
`AND`	Bitwise AND

Brackets may be used to change the order that the expression is evaluated in. When brackets are used in an expression, the calculations inside the brackets are performed first, then the rest of the expression is calculated. The calculations inside the brackets are still performed in standard arithmetic order:

```
#100= [10 + 2 * 2] * 3 - 5          This expression evaluates to 37
#100= 8 * [3 - 3] * 4               This expression evaluates to 0
```

The variable numbers themselves may be replaced with expressions, as long the expression evaluates to a valid variable number. For example:

```
#[100+5] = 10                  Variable number #105 is assigned the value 10
```

Variables may also be used to store variable numbers. This is called variable indirect and is a very useful feature. It can also be confusing, so here's some examples:

```
#100 = 105
```

```
#[#100] = 10          The value 10 is assigned to variable number #105
```

Here's what happens:

```
#100 = 105            Variable #100 now contains the value 105
#[105] = 10           The expression in brackets is evaluated first
#105 = 10             The variable is assigned using the result of the expression as the
```
variable number.

It is possible to use multiple levels of indirection:

```
#1 = 2
#2 = 3
###1 = 10             Variable #3 ends up with the value 10.
#[#[#1]] = 10         The same expression using brackets.
```

Functions

In addition to operators, there are many functions that may be used in an expression. A function takes a value and calculates a resulting value based on it's particular function. Unlike an operator, a function only needs one number to work from. The values used for functions must be enclosed in brackets.

For example:

`#100 = SQR[2]`	Variable #100 is assigned the value 1.414213 which is the square root of 2.

Here is a complete list of available functions:

`SIN`	Calculates the Sine of an angle in degrees
`COS`	Calculates the Cosine of an angle in degrees
`TAN`	Calculates the Tangent of an angle in degrees
`ATN, ATAN`	Calculates the ArcTangent
`ACOS`	Calculates the ArcCosine
`ASIN`	Calculates the ArcSine
`LN`	Calculates the natural logarithm
`SQR, SQRT`	Calculates the square root
`ABS`	Calculates the absolute value
`BIN`	Converts decimal to hexadecimal
`BCD`	Converts hexadecimal to decimal
`RND, ROUND`	Value rounding
`FIX`	Returns the integer portion of a value
`FUP`	Fractional values are rounded up to the next whole number
`EXP`	Exponent

The value passed to a function may be any valid expression:

`#102 = SQR[#100 + #101]`	The contents of variables #100 and #101 are added and the square root of this result is calculated.

Comparisons

The comparators are typically used with the IF - GOTO or IF - THEN macro statements. However, they may be used in any expression. When used as part of an expression, they return a value of 1 for TRUE or 0 for FALSE as their result. Comparators are like operators in that they require two operands.

List of comparators:

`EQ`	Equal to
`NE`	Not equal to
`LT`	Less than
`LE`	Less than or equal to
`GT`	Greater than
`GE`	Greater than or equal to

Comparison examples:

`#100 = #101 EQ #102`	If #101 equals #102 then #100 is assigned a value of 1. If not, it is assigned a value of 0.

If comparators are used as part of a larger expression, the comparison should be enclosed in brackets like this:

`#100 = 10 * [#101 EQ #102] + 5`	If #101 equals #102 then the expression evaluates to 15, otherwise it evaluates to 5.

Bitwise operators

Bitwise operators are a convenient way of combining comparisons. The bitwise operators are OR, AND and XOR.

"OR" operator:

`#100=[#101EQ#102]OR[#103EQ#104]`	Variable #100 is assigned a value of 1 if #101 equals #102 OR #103 equals #104, otherwise it gets a value of 0.

"AND" operator:

`#100=[#101EQ1]AND[#102EQ2]`	Variable #100 is assigned a value of 1 if #101 equals 1 AND #102 equals 2, otherwise it gets a value of 0.

"XOR" operator:

`#100= #101EQ1]XOR[#102EQ2]`	Variable #100 is assigned a value of 0 if both expressions are true or both expressions are false. If one is true and the other is false, #100 is assigned a value of 1.

Program Levels and Local Variables

Local variables are variable numbers #1 to #99 and are typically used as temporary use variables for subprograms. Even though the variable numbers are the same, each program level has it's own set of local variables. The program level changes any time you call or return from a subprogram. The main program is always level 0. A subprogram call from the main program is level 1.

The G65 macro call command allows passing values to the subprogram through local variables. This table shows the correspondence between the letter addresses and the variable numbers. The "x" denotes letters that cannot be used to pass variables.

Variable	Address
#1	A
#2	B
#3	C
#7	D
#8	E
#9	F
#10	G x
#11	H
#4	I
#5	J
#6	K
#12	L x
#13	M
#14	N x
#15	O x
#16	P x
#17	Q
#18	R
#19	S
#20	T
#21	U
#22	V
#23	W
#24	X
#25	Y
#26	Z

Local variable example:

```
G65 P9000 X10 Y5 Z1 A2.5 B3.6
```

When this block is executed, these values are assigned to local variables before program 9000 begins. Program 9000 can then read these values in these variables:

```
#24=10      (X)
#25=5       (Y)
#26=1       (Z)
#1=2.5      (A)
#2=3.6      (B)
```

These are called local variables because these values are only valid for the program they are passed to. If program 9000 also used a G65 macro call, the local variables are saved

and the new subprogram gets it's own set of local variables. They do not overwrite the values being used by program 9000. So when the new subprogram is done, the variables being used by program 9000 have not changed even though both subprograms use the same variable numbers. When a subprogram is finished, it's local variables are cleared.

Common and System Variables

Common variables are variable numbers #100 to #999 and, unlike local variables, can be used and set by any program. These variables are also retained on exit.

System variables are variable numbers #1000 on up. The actual range will depend on your specific control. System variables are used by the CNC to store internal values needed for operation. These values are things like tool length offsets, diameter offsets and machine positions. These variables may be set by your program, but care should be taken when doing so.

Using Variable #0

Variable #0 is a special variable that cannot be set. Instead, it's value is always <empty>. A variable that is <empty> is not the same as a variable that has been set to 0. An <empty> variable is a variable that contains no value, not even 0.

This variable cannot be set, but may be assigned to other variables and used in comparisons. This is especially useful for subprograms in determining if a value has been given for all required addresses:

```
G65 P9000 A0 B2

#1=0 (A)
#2=2 (B)
```

Variables #1 and #2 contain values because values were given in the G65 block. All other local variables are cleared to <empty>. If, for example, the address C is also required by the subprogram, it can check to see if a value was given by comparing it to variable #0. If #3 is equal to #0, then address C was not included in the G65 block that called the subprogram. Even though #1 was set to 0, this is a valid value and so it is not equal to #0.

Variables may be cleared to <empty> by assigning them the variable #0.

For example:

```
#100 = #0
```
The variable #100 is cleared to <empty>.

This also works using the variable indirect method:

```
#100 = 0
#110 = #[#100]
```
Variable #110 is cleared to <empty> because #100 points to variable #0.

When variable #0 is used in an expression, it is handled as the value 0 except for the comparisons EQ and NE.

Examples of using #0 in expressions:

```
#0 + 1 = 1
#0 * 10 = 0
#0 + #0 = 0
#[#101-#101] * 50 = 0

[#0 LT 0]  FALSE
[#0 LE 0]  TRUE

[#0 EQ 0]  FALSE
[#0 NE 0]  TRUE

[#0 GT 0]  FALSE
[#0 GE 0]  TRUE
```
These two comparisons are the only two that handle #0 and 0 as being different.

In general, the variable #0 shouldn't be used in expressions. It's use should be limited to EQ and NE comparisons and variable assignments.

Macro Statements

In addition to variables and expressions, the macro language uses a few macro statements that can control the flow of the program.

Here is a list of the macro statements:

```
IF - GOTO
IF - THEN
GOTO
WHILE - DO
DO
END
```

You'll notice that a few of the statements are grouped together. This is because the statements work together to determine the exact function performed.

These macro statements are recognized but not supported by NCPlot:

```
POPEN
DPRNT
BPRNT
PCLOS
```

These statements are recognized by the interpreter and the formatting tools, but are ignored by the backplotter.

Examples:

```
IF [#100EQ1] GOTO100
```

When this block is executed, the expression "[#100EQ1]" is evaluated, and if it is TRUE then the statement GOTO100 causes the program to jump to N100. If it is FALSE, the program execution continues to the next block.

The expressions used by IF statements do not have to be comparisons. The expression is considered to be TRUE if the result of the expression is non-zero. So any valid expression may be used with the IF statement.

```
IF [#100LT#101] THEN #102=5
```

When this block is executed and the expression "[#100LT#101]" is TRUE, then the variable #102 is assigned a value of 5. Otherwise, #102 is not changed and execution continues with the next block.

```
IF [#100GT#101] THEN G0 X#100
```

If the expression "[#100GT#101]" is TRUE, then the X axis moves to the position contained in variable #100. Any valid G-code block may follow the THEN statement.

```
GOTO200
```

When this block is executed, the program jumps to N200. Since no IF statement is used, this is called an unconditional jump.

```
WHILE [#100LT#101] DO1
...
...
...
END1
```

The WHILE - DO statements set up a conditional loop. When the WHILE - DO block is first encountered, the expression "[#100LT#101]" is evaluated. If this expression is FALSE, the program jumps to the block that contains the END statement. If this expression is TRUE, the program continues until the END statement is reached. When the END is reached, the WHILE expression is evaluated again. If this expression is still TRUE, the program jumps back to the WHILE - DO block, and the process repeats. So essentially, the program blocks between the WHILE - DO block and the END block are repeated until the expression evaluates to FALSE.

The DO and END statements have numbers after them that identify which END block belongs to which WHILE - DO block. This is because WHILE - DO loops may nested inside each other. The valid loop numbers are from 1 to 30.

```
DO2
...
...
...
END2
```

This is a WHILE - DO loop without the WHILE. It's conditional expression is always TRUE, which sets up a never ending loop. This type of loop must have some other means of breaking out of the loop, such as an IF - GOTO statement or even an M02 or M30 to end the program.

Examples

Macro programming is very flexible. The examples given here are not necessarily the right way or the only way to do something. This section is simply to help you understand how the macro language works.

Example 1:

This example clears variables #100 through #199 to <empty>.

```
#1 = 100                  (Assign the value 100 to variable #1)
N1 ##1 = #0               (The variable pointed to by variable #1 is cleared to <empty>)
#1 = #1 + 1               (Variable #1 is incremented by 1)
IF [#1 LT 200] GOTO 1     (This jump is taken as long as variable #1 is less than 200)
```

Here is another way to do the same thing using a WHILE – DO loop.

```
#1 = 100
WHILE [#1 LT 200] DO 1
##1 = #0
#1 = #1 +1
END1
```

Appendix D

Variable Map - Mill

This map details the variables recognized by NCPlot. While most controls that can be programmed in Macro B format will conform to this layout, you should check your controls documentation to be sure.

#0 Always <empty>

#1-#99 Local variables – Note that on some controls this is #1-#33

Variable	Address	Variable	Address
#1	A	#14	N x
#2	B	#15	O x
#3	C	#16	P x
#7	D	#17	Q
#8	E	#18	R
#9	F	#19	S
#10	G x	#20	T
#11	H	#21	U
#4	I	#22	V
#5	J	#23	W
#6	K	#24	X
#12	L x	#25	Y
#13	M	#26	Z

#100 - #999 Common variables

The number and range of common variables will depend on your control. On many controls, adding more common variables is an extra cost option.

#1000 - ? System variables

System variables are used by the CNC and should only be changed by your macro program with great care. The actual range of variables depends on your particular control, but the variables recognized by NCPlot are described here.

#3000 Macro Alarm

Assigning a value to this variable causes NCPlot to display a macro message. This is usually in the format #3000=100(MACRO MESSAGE).

#4201 - #4220 G Modals

The group 0 codes are non-modal and do not appear in the system variables. They are included here for the sake of completeness. NCPlot does not support all of the G-Codes listed here, but will store them in the proper group when encountered in a program.

Group 0 G4, G9, G10, G11, G12, G13, G28, G34, G35, G36, G37.1, G52, G53, G65, G92

#4201	Group 1	G0, G1, G2, G3
#4202	Group 2	G17, G18, G19
#4203	Group 3	G90, G91
#4205	Group 5	G94, G95
#4206	Group 6	G20, G21
#4207	Group 7	G40, G41, G42
#4208	Group 8	G43, G44, G49
#4209	Group 9	G70 – G79, G81 – G89, G80
#4210	Group 10	G98, G99
#4211	Group 11	G50, G51
#4212	Group 12	G54 – G59, G54.1
#4214	Group 14	G66, G67
#4216	Group 16	G68, G69
#4218	Group 18	G15, G16
#4219	Group 19	G50.1, G51.1

#4301-#4320 Other Modals

#4301	A	#4311	H
#4302	B	#4312	L
#4303	C	#4313	M
#4304	I	#4314	N
#4305	J	#4315	O
#4306	K	#4316	P
#4307	D	#4317	Q
#4308	E	#4318	R
#4309	F	#4319	S
#4310	G	#4320	T

Axis positions

	X axis	Y axis	Z axis	4th axis
Last block endpoint	#5001	#5002	#5003	#5004
Machine coordinate	#5021	#5022	#5023	#5024
Work coordinate	#5041	#5042	#5043	#5044

Work Offsets

	X axis	Y axis	Z axis	4th axis
G54	#5221	#5222	#5223	#5224
G55	#5241	#5242	#5243	#5244
G56	#5261	#5262	#5263	#5264
G57	#5281	#5282	#5283	#5284
G58	#5301	#5302	#5303	#5304
G59	#5321	#5322	#5323	#5324

Extended Work Offsets – 48 Offsets

	X axis	Y axis	Z axis	4th axis
G54.1 P1	#7001	#7002	#7003	#7004
G54.1 P2	#7021	#7022	#7023	#7024
G54.1 P3	#7041	#7042	#7043	#7044
…				
…				
…				
G54.1 P46	#7901	#7902	#7903	#7904
G54.1 P47	#7921	#7922	#7923	#7924
G54.1 P48	#7941	#7942	#7943	#7944

Extended Work Offsets – 300 Offsets

	X axis	Y axis	Z axis	4th axis
G54.1 P1	#14001	#14002	#14003	#14004
G54.1 P2	#14021	#14022	#14023	#14024
G54.1 P3	#14041	#14042	#14043	#14044
…				
…				
…				
G54.1 P298	#19941	#19942	#19943	#19944
G54.1 P299	#19961	#19962	#19963	#19964
G54.1 P300	#19981	#19982	#19983	#19984

Variable Map - Lathe Format A

This map details the variables recognized by NCPlot. While most controls that can be programmed in Macro B format will conform to this layout, you should check your controls documentation to be sure.

#0 Always <empty>

#1-#99 Local variables – Note that on some controls this is #1-#33

Variable	Address	Variable	Address
#1	A	#14	N x
#2	B	#15	O x
#3	C	#16	P x
#7	D	#17	Q
#8	E	#18	R
#9	F	#19	S
#10	G x	#20	T
#11	H	#21	U
#4	I	#22	V
#5	J	#23	W
#6	K	#24	X
#12	L x	#25	Y
#13	M	#26	Z

#100 - #999 Common variables

The number and range of common variables will depend on your control. On many controls, adding more common variables is an extra cost option.

#1000 - ? System variables

System variables are used by the CNC and should only be changed by your macro program with great care. The actual range of variables depends on your particular control, but the variables recognized by NCPlot are described here.

#3000 Macro Alarm

Assigning a value to this variable causes NCPlot to display a macro message. This is usually in the format #3000=100(MACRO MESSAGE).

#4201 - #4220 G Modals

The group 0 codes are non-modal and do not appear in the system variables. They are included here for the sake of completeness. NCPlot does not support all of the G-Codes listed here, but will store them in the proper group when encountered in a program.

Group 0 G4, G9, G10, G11, G28, G50, G52, G53, G65

#4201	Group 1	G0, G1, G2, G3
#4205	Group 5	G98, G99
#4206	Group 6	G20, G21
#4207	Group 7	G40, G41, G42
#4209	Group 9	G70 – G79, G81 – G89, G80, G90, G92, G94
#4212	Group 12	G54 – G59, G54.1
#4214	Group 14	G66, G67
#4217	Group 17	G96, G97

#4301-#4320 Other Modals

#4301	A	#4311	H
#4302	B	#4312	L
#4303	C	#4313	M
#4304	I	#4314	N
#4305	J	#4315	O
#4306	K	#4316	P
#4307	D	#4317	Q
#4308	E	#4318	R
#4309	F	#4319	S
#4310	G	#4320	T

Axis positions

	X axis	Z axis
Last block endpoint	#5001	#5002
Machine coordinate	#5021	#5022
Work coordinate	#5041	#5042

Work Offsets

	X axis	Z axis
G54	#5221	#5222
G55	#5241	#5242
G56	#5261	#5262
G57	#5281	#5282
G58	#5301	#5302
G59	#5321	#5322

Extended Work Offsets – 48 Offsets

	X axis	Z axis
G54.1 P1	#7001	#7002
G54.1 P2	#7021	#7022
G54.1 P3	#7041	#7042
…		
…		
…		
G54.1 P46	#7901	#7902
G54.1 P47	#7921	#7922
G54.1 P48	#7941	#7942

Extended Work Offsets – 300 Offsets

	X axis	Z axis
G54.1 P1	#14001	#14002
G54.1 P2	#14021	#14022
G54.1 P3	#14041	#14042
…		
…		
…		
G54.1 P298	#19941	#19942
G54.1 P299	#19961	#19962
G54.1 P300	#19981	#19982

Variable Map - Lathe Format B

This map details the variables recognized by NCPlot. While most controls that can be programmed in Macro B format will conform to this layout, you should check your controls documentation to be sure.

#0 Always <empty>

#1-#99 Local variables – Note that on some controls this is #1-#33

Variable	Address	Variable	Address
#1	A	#14	N x
#2	B	#15	O x
#3	C	#16	P x
#7	D	#17	Q
#8	E	#18	R
#9	F	#19	S
#10	G x	#20	T
#11	H	#21	U
#4	I	#22	V
#5	J	#23	W
#6	K	#24	X
#12	L x	#25	Y
#13	M	#26	Z

#100 - #999 Common variables

The number and range of common variables will depend on your control. On many controls, adding more common variables is an extra cost option.

#1000 - ? System variables

System variables are used by the CNC and should only be changed by your macro program with great care. The actual range of variables depends on your particular control, but the variables recognized by NCPlot are described here.

#3000 Macro Alarm

Assigning a value to this variable causes NCPlot to display a macro message. This is usually in the format #3000=100(MACRO MESSAGE).

#4201 - #4220 G Modals

The group 0 codes are non-modal and do not appear in the system variables. They are included here for the sake of completeness. NCPlot does not support all of the G-Codes listed here, but will store them in the proper group when encountered in a program.

```
Group 0     G4, G9, G10, G11, G28, G52, G53, G65, G92

#4201       Group 1     G0, G1, G2, G3
#4203       Group 3     G90, G91
#4205       Group 5     G94, G95
#4206       Group 6     G20, G21
#4207       Group 7     G40, G41, G42
#4209       Group 9     G70 – G79, G81 – G89, G80
#4210       Group 10    G98, G99
#4212       Group 12    G54 – G59, G54.1
#4214       Group 14    G66, G67
#4217       Group 17    G96, G97
```

#4301-#4320 Other Modals

#4301	A	#4311	H
#4302	B	#4312	L
#4303	C	#4313	M
#4304	I	#4314	N
#4305	J	#4315	O
#4306	K	#4316	P
#4307	D	#4317	Q
#4308	E	#4318	R
#4309	F	#4319	S
#4310	G	#4320	T

Axis positions

	X axis	Z axis
Last block endpoint	#5001	#5002
Machine coordinate	#5021	#5022
Work coordinate	#5041	#5042

Work Offsets

	X axis	Z axis
G54	#5221	#5222
G55	#5241	#5242
G56	#5261	#5262
G57	#5281	#5282
G58	#5301	#5302
G59	#5321	#5322

Extended Work Offsets – 48 Offsets

	X axis	Z axis
G54.1 P1	#7001	#7002
G54.1 P2	#7021	#7022
G54.1 P3	#7041	#7042
…		
…		
…		
G54.1 P46	#7901	#7902
G54.1 P47	#7921	#7922
G54.1 P48	#7941	#7942

Extended Work Offsets – 300 Offsets

	X axis	Z axis
G54.1 P1	#14001	#14002
G54.1 P2	#14021	#14022
G54.1 P3	#14041	#14042
…		
…		
…		
G54.1 P298	#19941	#19942
G54.1 P299	#19961	#19962
G54.1 P300	#19981	#19982

Appendix E

Scripting

NCPlot supports scripts written in the VBScript language. This gives you a very powerful tool useful for automating common tasks. This manual assumes you already know how to write scripts in the VBScript language and only provides a reference for the functions made available by NCPlot.

VBScripting extends the capabilities of NCPlot by giving you, the script writer, access to many of NCPlot's internal functions.

This tool makes it possible to:

- Automate common conversion tasks, such as converting code that is written to run on one machine into code for another machine.
- Create custom code generation tools that can accept user input.
- Batch process whole file folders using the provided functions for wildcard file searching.

When used in your script, the function names listed in this manual must be preceded by the keyword 'ncplot', for example:

Ncplot.NCPFileLoad "c:\test.txt"

To execute a script file in NCPlot, go to File | Execute Script File. Browse to the file you want to execute and click 'OK'. Script files may be edited with any text editor.

NCPSetConfig

NCPlot.NCPSetConfig cfgName

Set machine configuration. The value of cfgName should be the name of an existing configuration, “Mill” for example.

NCPSelectLines

NCPlot.NCPSelectLines lngStart, lngEnd

Select range of program lines. Selects from line lStart to lEnd in the edit window. This is useful if you want to apply formatting or conversion operations to only part of a program.

NCPSetSelection

NCPlot.NCPSetSelection lngStart, lngLength

Set selection start and length. Selects text beginning at character index lStart for lLength characters.

NCPInsertText

NCPlot.NCPInsertText strText

Insert text into edit window. The given string txtString is inserted into the program at the current insertion point. The insertion point can be changed with NCPSetSelection by setting lLength to 0. If any text is currently selected, txtString will replace it.

NCPInsertLine

NCPlot.NCPInsertLine strText

Insert line into edit window. Same as NCPInsertText except this function also adds a carriage return / line feed to the end of the line.

NCPGetNumLines

NCPlot.NCPGetNumLines

Returns the number of lines in the program.

NCPGetLine

NCPlot.NCPGetLine lngNumber

Returns the contents of the requested program line number.

NCPGetLineIndex

NCPlot.NCPGetLineIndex lngNumber

Returns a pointer to the beginning of the requested line number.

NCPFileNew

NCPlot.NCPFileNew

Clears the edit window and graphics viewport of the current program. Use with Caution!! This function does not prompt you to save current edits before clearing the file. Be sure the loaded file is saved before using this function.

NCPFileLoad

NCPlot.NCPFileLoad strPath

Loads a file into the edit window. Use With Caution!! This function does not prompt you to save current edits before loading the file. Be sure the loaded file is saved before using this function.

NCPFileSave

NCPlot.NCPFileSave

Saves the currently loaded file.

NCPExportDXF

NCPlot.NCPExportDXF strPath

File export as DXF. The currently loaded file is converted to DXF and saved to the filename in fPath.

NCPRenumber

NCPlot.NCPRenumber StartBlock, BlockIncrement, MaxDigits, Rstyle, AddSpace

Renumbers the currently loaded file.

StartBlock	The starting block number.
BlockIncrement	Block number increment.
MaxDigits	Maximum number of digits in block number.
RStyle	Renumber style 0 = Remove block numbers 1 = Renumber all blocks 2 = Renumber all but blank and comment blocks 3 = Renumber GOTO referenced blocks only
AddSpace	0 = No Space after block number 1 = Add space after block number

NCPColorize

NCPlot.NCPColorize

Applies the address color settings to the loaded program.

NCPAddSpaces

NCPlot.NCPAddSpaces

Equivalent to the Add Spaces tool.

NCPRemoveSpaces

NCPlot.NCPRemoveSpaces

Equivalent to the Remove Spaces tool.

NCPRemoveTrailingSpaces

NCPlot.NCPRemoveTrailingSpaces

Equivalent to the Remove Trailing Spaces tool.

NCPRemoveBlankLines

NCPlot.NCPRemoveBlankLines

Equivalent to the Remove Blank Lines tool.

NCPRemoveComments

NCPlot.NCPRemoveComments

Equivalent to the Remove Comments tool.

NCPRemoveGarbage

NCPlot.NCPRemoveGarbage

Equivalent to the Remove Garbage tool.

NCPAllCaps

NCPlot.NCPAllCaps

Equivalent to the Convert To All Caps tool.

NCPRemoveRedundant

NCPlot.NCPRemoveRedundant

Equivalent to the Remove Redundant Endpoints tool.

NCPMirror

NCPlot.NCPMirror Xpoint, Ypoint, Zpoint

Mirror axis endpoints. The MirrorPoint values specify the center point for mirroring on that axis. If a value is omitted, no mirroring is performed on that axis.

NCPRotate

NCPlot.NCPRotate xCenter, yCenter, Angle

Rotate X, Y coordinates. Specify X rotation center, Y rotation center and rotation angle.

NCPShift

NCPlot.NCPShift xShift, yShift, zShift

Shift axis coordinate values. The specified values are added to the programmed coordinates.

NCPScale

NCPlot.NCPScale xScalePoint, yScalePoint, ScaleFactor

Scale axis coordinate values. The ScalePoint values are the X, Y center point for scaling.

NCPAddressAdjust

NCPlot.NCPAddressAdjust adjAddrList, adjOp, adjValue, adjFormat

Apply address value adjustments and formatting.

adjAddrList	String that contains the list of addresses to adjust. For example, the string "XYZ" will apply the specified adjustment to all X, Y and Z coordinates.
AdjOp	Value that specifies the type of adjustment: 0 = add, 1 = subtract, 2 = multiply, 3 = divide.
adjValue	The adjustment value.
adjFormat	Formatting string to be applied to the result of the adjustment.

NCPAddressReplace

NCPlot.NCPAddressReplace strFind, strReplace

Replaces address identifiers.

strFind	String that contains the address character to find.
strReplace	String that contains the address character to replace it with.

NCPAddressRemove

NCPlot.NCPAddressRemove strFind

Removes address identifiers and their values.

strFind	String that contains the addresses to remove.

NCPConvertArcsToAbsolute

NCPlot.NCPConvertArcsToAbsolute

Equivalent to the Convert arc centers to absolute tool.

NCPConvertArcsToIncremental

NCPlot.NCPConvertArcsToIncremental

Equivalent to the Convert arc centers to incremental tool.

NCPConvertArcsToIJK

NCPlot.NCPConvertArcsToIJK

Equivalent to the Convert arc R to IJK tool.

NCPConvertArcsToR

NCPlot.NCPConvertArcsToR

Equivalent to the Convert arc IJK to R tool.

NCPBrowseForFile

NCPlot.NCPBrowseForFile

Opens the File Browse dialog for allowing the user to select a file. Returns the selected filename, with path. Returns a Null string if the user clicks Cancel on the browse dialog.

NCPBrowseForFolder

NCPlot.NCPBrowseForFolder

Opens the Folder Browse dialog for allowing the user to select a file. Returns the selected path without the ending "\" delimiter. A Null string is returned if the user clicks 'Cancel' on the browse dialog.

NCPGetFirstMatchingFile

NCPlot.NCPGetFirstMatchingFile strPath

Returns the first matching directory entry that matches a given path string, including wildcard characters. This function returns the filename of the first matching entry, or a Null string if no matches were found.

NCPGetNextMatchingFile

NCPlot.NCPGetNextMatchingFile

Returns the next matching directory entry. When the function NCPGetFirstMatchingFile is used to start a wildcard file search, this function will return subsequent matches or a Null string when no more matches are found.

NCPSelectAll

NCPlot.NCPSelectAll

Selects the entire contents of the edit window.

NCPGetAll

NCPlot.NCPGetAll

This function returns the entire contents of the edit window as a string.

NCPGetSelected

NCPlot.NCPGetSelected

This function returns the selected portion of the edit window as a string.

NCPReplaceAll

NCPlot.NCPReplaceAll str1, str2

This function finds all occurrences of the string **str1** and replaces them with the string **str2**.

NCPPlot

NCPlot.NCPPlot

Refreshes the graphics viewport. Equivalent to the Refresh Plot tool.

NCPAddressSwap

NCPlot.NCPAddressSwap str1, str2

This function calls the address swap tool. The string values **str1** and **str2** should each contain a single address character “A” to “Z” excluding the address “G”. This tool will scan the program, swapping these two address values.