PostProcessor 2014

Reference Help

Issue: 2014 *Released: 12/05/2014*

Important Notices

Delcam plc has no control over the use made of the software described in this manual and cannot accept responsibility for any loss or damage howsoever caused as a result of using the software. Users are advised that all the results from the software should be checked by a competent person, in accordance with good quality control procedures.

The functionality and user interface in this manual is subject to change without notice in future revisions of software.

The software described in this manual is furnished under licence agreement and may be used or copied in accordance with the terms of such licence.

Delcam plc grants permission for licensed users to print copies of this manual or portions of this manual for personal use only. Schools that are licensed to use the software may make copies of this manual or portions of this manual for students currently registered for classes where the software is used.

Acknowledgements

This documentation references a number of registered trademarks and these are the property of their respective owners. For example, Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States.

Delcam PostProcessor 2014 Date: 12/05/2014 11:00

Contents

System requirements

Supported Operating System

Windows 7 SP1, Windows Server 2008 R2 SP1, Windows Server 2008 SP2, Windows Vista SP2

- Windows Vista SP2 (x86 and x64)
- Windows 7 SP1 (x86 and x64)
- Windows Server 2008 R2 SP1 (x64)
- Windows Server 2008 SP2 (x86 and x64)

Hardware Requirements:

- 1 GHz or faster processor
- 512 MB of RAM
- 850 MB of available hard disk space (x86)
- 2 GB of available hard disk space (x64)

The additional software components are installed automatically if necessary.

Delcam PostProcessor is compatible with Windows 7.

What's New

General enhancements

- Improved presentation of conditional statements in command blocks
- Customisable header for erroneous files
- Folders for CLDATA files
- Commands and parameters are shown according to machine capabilities
- Merging in PostComparer
- Contact point parameters
- Commands and parameters for rotary axis move detection
- Static axes in MTD considered

Features

Delcam PostProcessor is a tool for generating controller-specific NC programs from CLDATA files produced by CAM systems. It also includes an editor for creating and editing the option files that are used to convert the CLDATA files, and an application for comparing different versions of option files.

It has the following features:

- A graphical user interface for postprocessing CLDATA files, such as the ***.cut** files generated by **PowerMILL**;
- Access to the internal structure of the **Postprocessor** and a link to the **Editor**, which shows a hierarchical representation of the option file structure;
- Simple and efficient handling of post-processor commands;
- Continuous 5-axis support;
- Local coordinate system transformation support;
- Probing support, integration with PowerINSPECT;
- A search tool for finding elements in option files;
- Option file reports;
- Debug mode for output files;
- Option file comparison and merging;
- **Script functions** for sophisticated handling of post-processor commands. Based on Microsoft Active Scripting, and using JScript and VBScript, scripting enables you to implement specific tasks for any controller.

Backward compatibility with v4.8

Working with PostProcessor 2010 many users experienced troubles with option files written in PostProcessor 4.8 or earlier. The main reason is that v2010 and onward applies some new approaches and response order is quite different. Versions of Delcam PostProcessor before v2014 processed old option files (*.pmopt) in a compatibility mode preserving the behaviour of v4.8.

From v2014 onward this compatibility mode is DISCONTINUED.

An attempt to process with an old option file will cause the following error message:

Option files of this version must be re-saved in new format and checked carefully.

In order to get rid of this message you have to save the option file in the new format. It's strongly recommended to check that the output generated by this new option file is correct. If you are not sure in correctness of the output please contact Delcam Support.

Overview

A design engineer uses **CAD (Computer Aided Design)** software to create a part drawing, containing all features and dimensions required to manufacture a part.

Computer Aided Manufacturing (CAM) evolved from the need to manufature using numerically controlled machine tool. Once CAD drawing is complete, it can be imported into CAM software for further use. Using various functions withing the CAM software, the programmer selects various lines, circles, surfaces or other entities from the imported CAD drawing and then develops cutting sequences, or tool paths, that simulate actual machinig of those entities.

The most CAM applications, the tool path that is generated for each cutting sequence is placed into a file in a specific order and syntax, almost like a structured language. This file is called a **Cutter Location Data (CLDATA)** file. It contains various commands, as well as X, Y, Z coordinate values, and in the case of four and five axes machines, will also contain tool vector I, J, K values used to determine the angle of the tool in relation to the part.

Postprocessor reads each statement or command contained withing the CLDATA and postprocesses (translates) each of those statements or commands into CNC machine code. This CNC machine code (G-Codes) is created based on the content of the CLDATA and in combination with the settings in **option file**.

The **Delcam PostProcessor** application contains the following modules:

- The **Postprocessor** creates programs for NC/CNC machine tool, in the form of tapefiles.
- The **Editor** is used to create or edit the option file, which configures the **Postprocessor**.
- The **Console** creates output suitable for an NC/CNC machine tool using the commands and parameters specified in the **Editor**.
- The **PostComparer** enables you to compare the structures of different option files.

Introduction to the Postprocessor

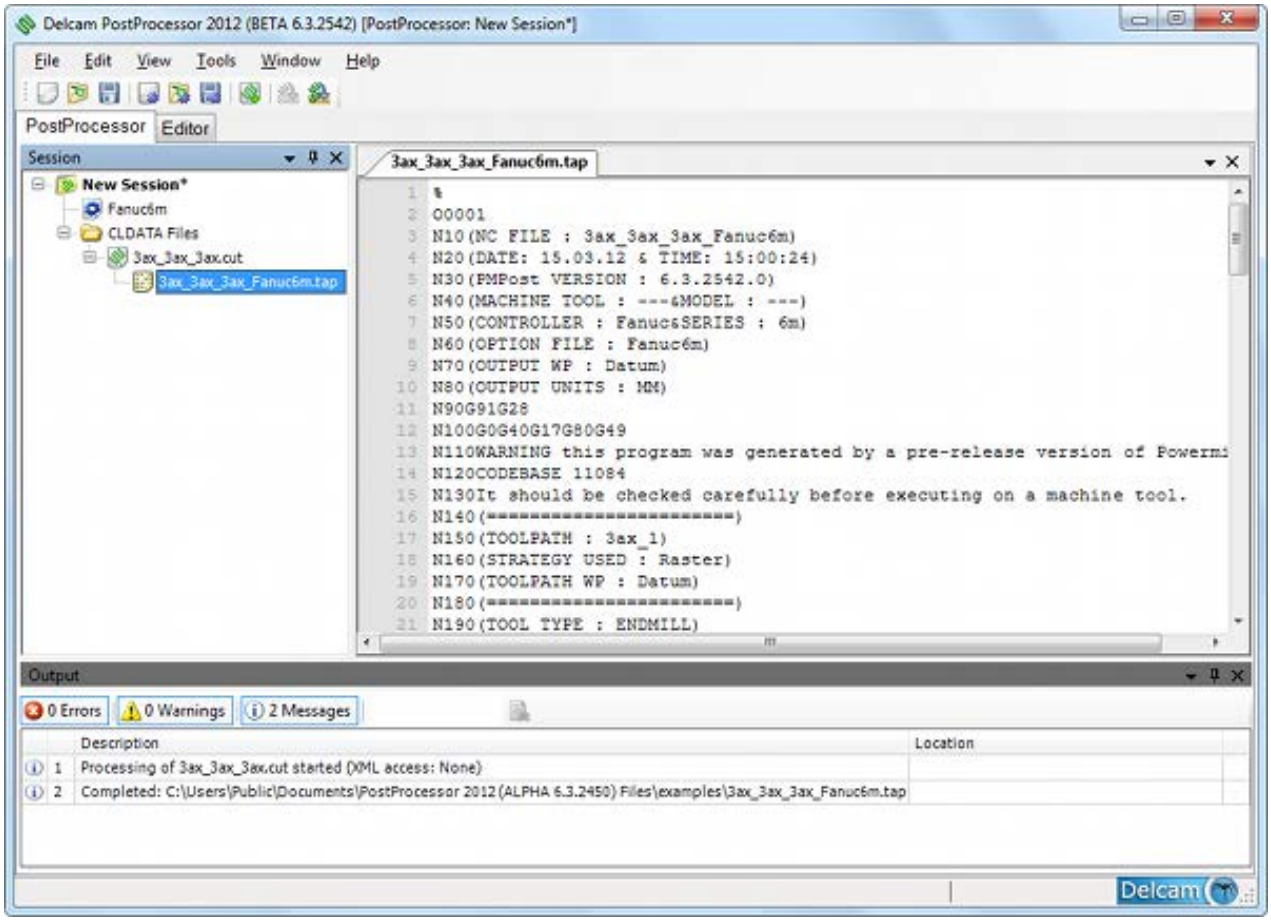
The **Postprocessor** is used to generate NC programs for a specific machine controllers from CLDATA files produced by CAM systems.

The **Postprocessor** is universal NC program generator, which uses custom option files. An option file describes all specifics and controls how the CLDATA files will be converted for a particular controller. Use **Editor** for create or edit option files.

The **Postprocessor** may work with full range of Delcam CLDATA formats:

- ASCII or Binary CLDATA. For example, a ***.cut** file generated by **PowerMILL**.
- XML-CLDATA. For example, a ***.cxm** file containing probe paths that was exported from **PowerINSPECT**.

On the image below you may see main window of Postprocessor. On the left, is located **Session Explorer**, that allows you to load option files, to load and postprocess CLDATA files. Result NC programs you may see on the right pane. The bottom window is needed for diplaying errors, warnings and other information about postprocessing.

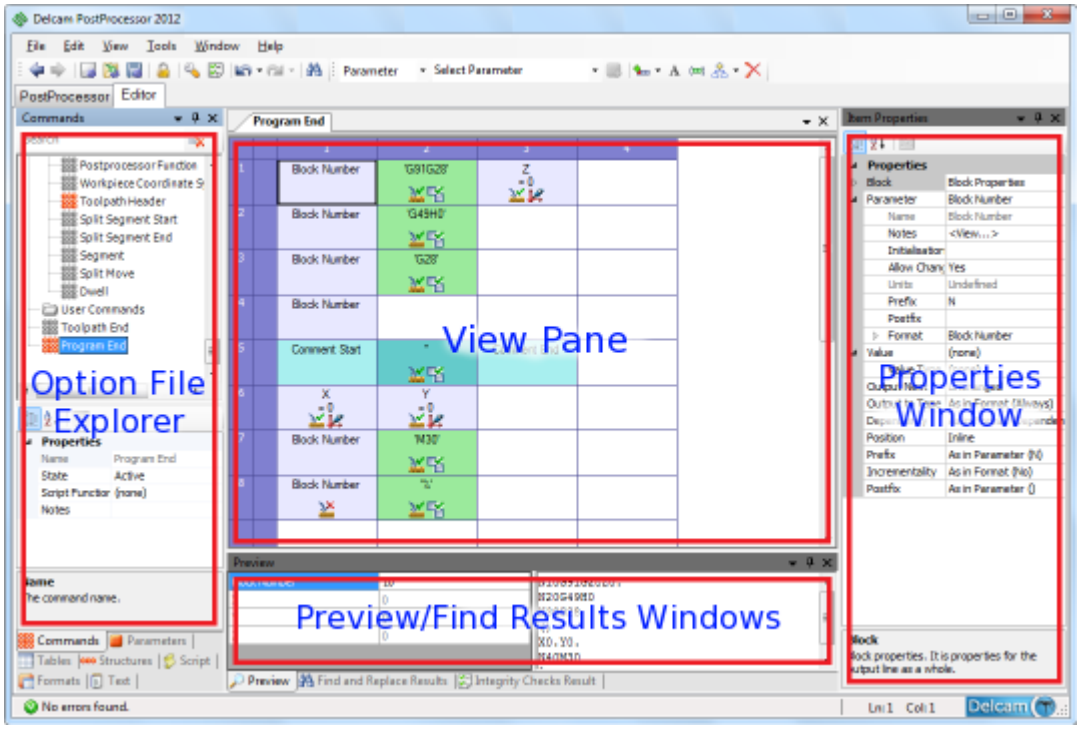


Introduction to the Editor

The **Editor** enables you to create and edit the controller-specific option files used in configuring the **Postprocessor** to create output suitable for the associated NC/CNC machine tools.

To open the **Editor**, click the **Editor** tab.

A set of standard, selectable elements are displayed in the **Option File Explorer**. When a command is selected in the explorer, its details are as a table in the **View** pane. For example:



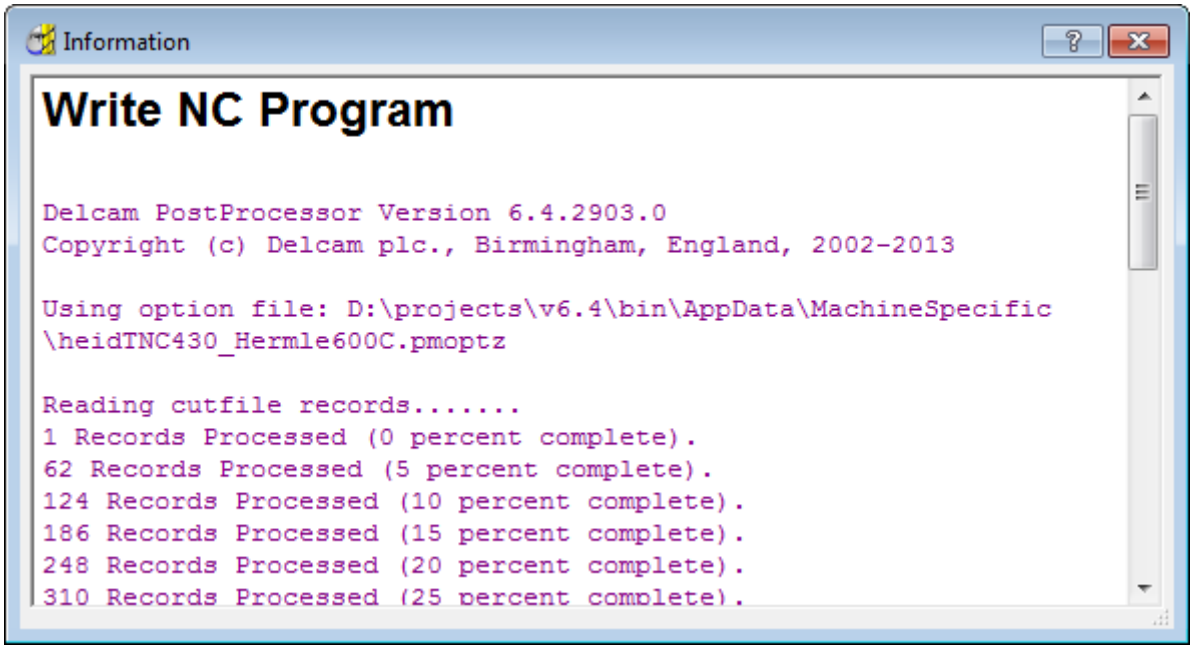
Each command table represents a program frame. It specifies the NC code that is output to the tapefile when that command is found in the cut file. For more details, refer to Command Interface.

You can use the **Preview** window at the bottom of the **PostProcessor** to view the NC code created for the selected command.

Console

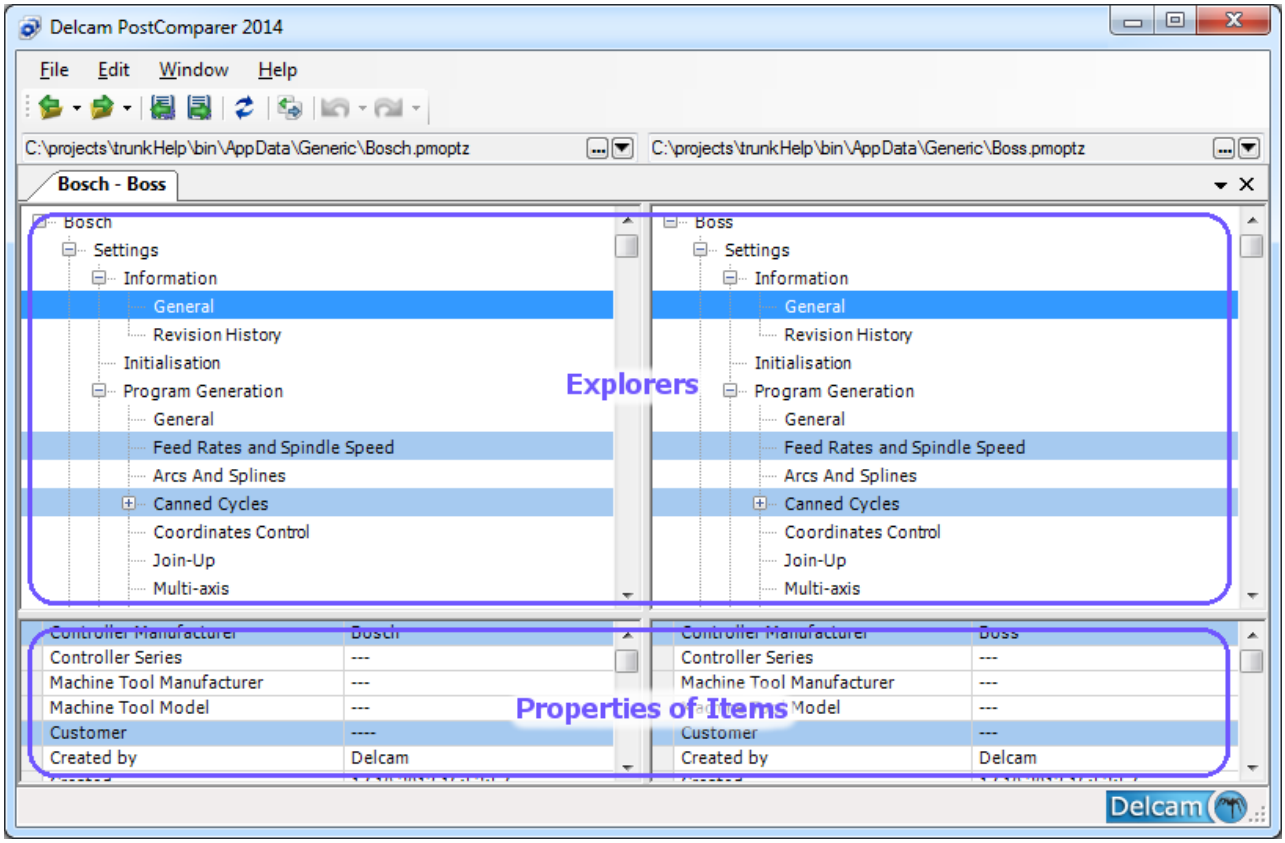
The **Console** module enables you to generate NC programs using a command prompt.

This GUI-less application usually executed by CAM systems in background. On the image below you may see console output in **Delcam PowerMILL**.



Introduction to the PostComparer

The **Delcam PostComparer** enables you to compare the structure of selected option files. Use it to view the differences between settings, commands, parameters, formats and so on. In addition, you may merge selected changes/features from one option file to another.



The **Explorer** windows display the different sections of the option files to be compared. The differences between the files are highlighted.

The **Properties** windows display the properties of the selected elements in the **Explorer** windows.

Using Delcam PostProcessor

The Delcam PostProcessor reference help includes the following sections:

Toolbar and menus

The toolbar and menus enable you to create, open and save option files; to specify the properties of option files; to password-protect option files; and to generate option file reports.

Toolbar

When the PostProcessor tab is selected:

Button	Description
	Creates a new session and closes the current one.
	Opens a session.
	Saves the current session to a file. When you save the session for the first time, the **Save Session As** dialog is displayed.
	Creates an option file.
	Opens an option file.
	Saves the current option file.
	Loads a CLDATA file, such as a ***.cut** or ***.cxm** file.
	Postprocesses the selected CLDATA files using the current option file.
	Postprocesses all CLDATA files using the current option file.

When the Editor tab is selected:

Button	Description
	Displays the previous page you selected.
	Displays the next page you selected.
	Creates an option file.
	Opens an option file.
	Saves the current option file.
	Password-protects the option file.
	Specifies the option file properties, such as machine kinematics, program generation setting, output format.
	Undoes the last option file edit.
	Reverses the action of the last **Undo**.
	Searches the option file for the text specified in the **Find** dialog. You can search the entire option file, or restrict the search to one type of element.

Menus

The menus are located at the top of the window. They contain commands that enable you to control and work with the Delcam PostProcessor.

- **File** menu
- **Edit** menu
- **View** menu
- **Tools** menu
- **Window** menu
- **Help** menu

File menu

When the PostProcessor tab is selected:

New >

- **Option File** - Creates an option file.
- **Session** - Creates a new session.

Open >

- **Option File** - Loads an option file.
- **Session** - Loads a session from file.
- **Session Folder** - Loads the files associated with a session from a session file.

Save Option File - Saves the option file to disk.

Save Option File As - Saves the option file under a new name and folder.

Save Session - Saves the session to file.

Save Session File As - Saves the session, together with the pointers to the associated session files (option file, CLDATA and output files), under a new name and folder.

Save Session Folder - Saves copies of all the files in the session to a new folder.

Session Properties - Sets session preferences.

Add CLDATA - Adds a CLDATA file to the session.

Add CLDATA Folder - Adds all the CLDATA files from a selected folder to the session. If the files are in sub-folders those sub-folders appear in the Session Tree.

Recent Session - Lists recently used sessions.

Recent Option File - Lists of recently used option files.

Recent CLDATA File - Lists recently used CLDATA file.

Exit - Closes **Delcam PostProcessor**.

When the Editor tab is selected:

New - Creates an option file.

Open - Loads an option file.

Save - Saves the option file.

Save as - Saves the option file under a new name.

Recent - Lists recently used option files.

Option File Properties - Specifies the option file properties, such as machine kinematics, program generation setting, output format.

Set Protection - Password-protects the option file.

Exit - Closes **Delcam PostProcessor**.

Edit menu

When the PostProcessor tab is selected:

Copy - Copies the selected text from the **Output** pane.

Select All - Selects all text in the **Output** pane.

Find - Displays the **Find** toolbar.

Find Next- Searches the **Output** pane for the next instance of the text specified in the **Find** toolbar.

When the Editor tab is selected:

Undo - Undoes your last edit action.

Redo - Cancels your last **Undo** action.

Cut - Copies the selected item to the clipboard and deletes it.

Copy - Copies the selected item from **View** pane to the clipboard.

Paste - Pastes the contents of the clipboard after the current selection.

When you paste items from another option file, the **Paste Wizard** is displayed so you can choose whether to paste settings associated with those items. For example, if you paste a command from another option file, you must also decide whether to paste the states and values associated within any parameters that already exist in the target option file.

Delete - Deletes the selected item.

Select All - Selects all items in the **View** pane.

Find - Displays the **Find** dialog.

Using the Paste wizard

When you copy items in Delcam PostProcessor, the Paste wizard copies the items and all their associated data using the same XML model as that used in the option file. For example, when you copy a parameter to the clipboard, the wizard also copies the format assigned to the parameter.

When you paste data from the clipboard, the wizard checks the data in the clipboard against the target option file and queries any differences. For example, if you paste a parameter that has the same name as a parameter in the target file, but the parameters use different formats, the wizard asks whether you want to paste the format from the clipboard or keep the format of the parameter in the target file.

If the wizard encounters a problem, an explanatory message is displayed and the paste is automatically cancelled.

Pasting formats

For each format on the clipboard, the wizard searches the target option file for a format with the same name as source format:

- If no match is found, the format is pasted from the clipboard to the target file.
- If a matching name is found and the formats are identical, the clipboard format is not pasted. The format of the target file is automatically assigned to any pasted parameters which use that format.
- If a matching name is found but the formats are different, a message is displayed asking whether you want to paste the format from the clipboard into the target file. The selected format is automatically applied to all original parameters and all pasted parameters.

Pasting parameters

Pasting parameters works in the same way as pasting formats. However, because parameters are linked to a specific format, you are asked about merging the prefix, postfix and group states when the format of the target parameter differs from that of the parameter being pasted. Any additional questions are displayed on separate dialogs within the wizard.

Parameters cannot be pasted when the target option file contains a parameter that has the same name as a parameter in the clipboard and the parameters have a different value type.

Pasting tables

Pasting tables works in the same way as pasting formats.

Pasting structures

Pasting structures works in the same way as pasting formats.

Pasting commands

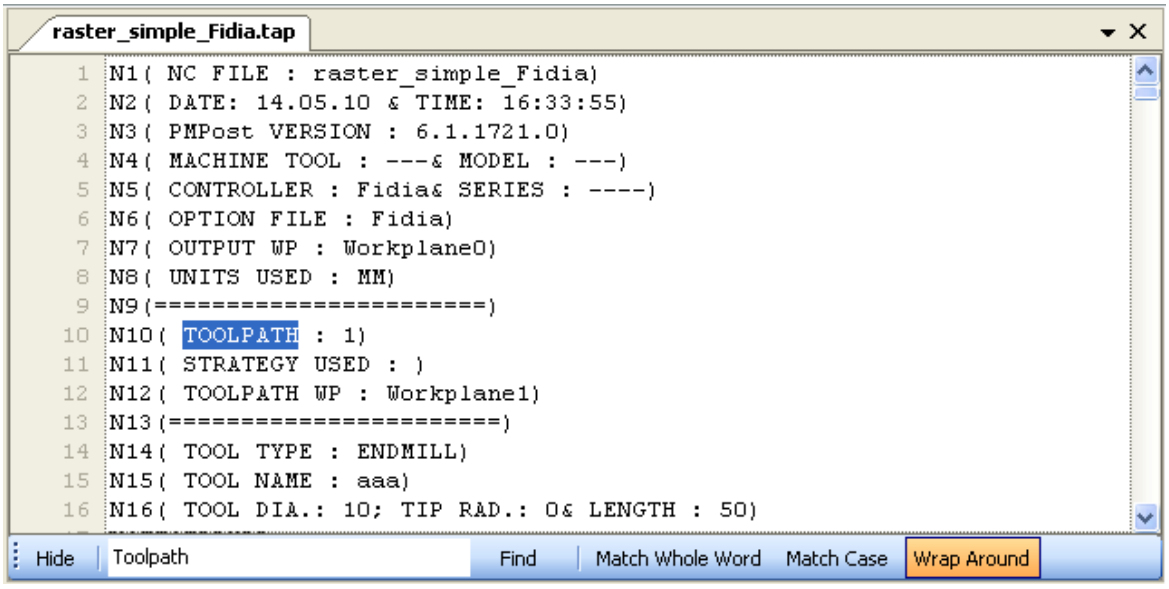
Pasting commands works in the same way as pasting formats.

Using the Find toolbar

The **Find** toolbar is used to search for strings in the **View** pane of the **PostProcessor** tab. Use the **Find** toolbar to search the active output file (displayed in the **View** pane of the **PostProcessor** tab) for text strings.

To search for a text string:

1. Select the **Edit > Find** menu option, or press **Ctrl+Shift+F**. The **Find** toolbar is displayed at the bottom of the **View** pane.



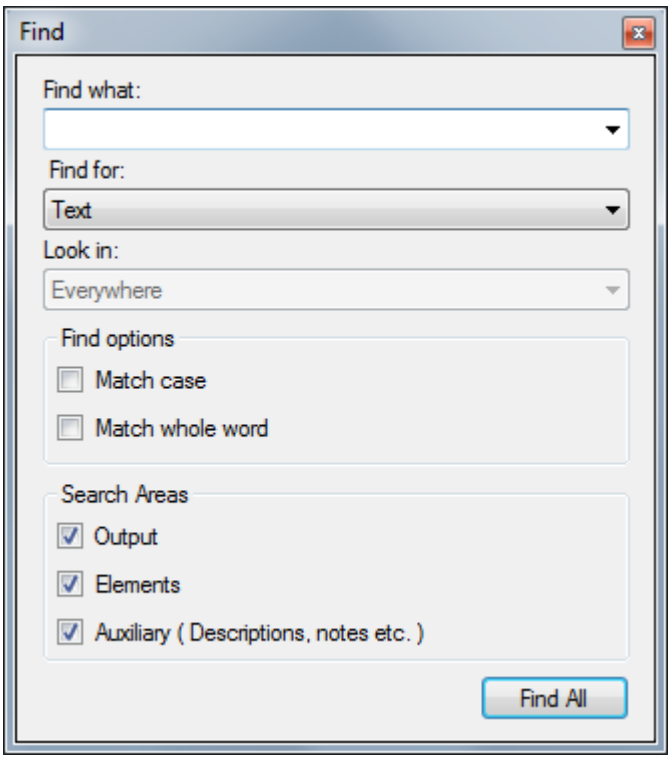
2. Enter the text you want to search for in the entry box.
3. If you want to prevent the search from finding instances of the string within words, select **Match Whole Word Only**.
4. If you want to restrict the search to names that match the case of the string you entered, select **Match Case**.
5. If you want the search to continue from the beginning of the file when the end is reached, select **Wrap Around**.
6. Select **Find**. The first instance of the string is selected and displayed in the **View** pane. Click **Find** again to search for the next instance of the string.
7. When you have finished searching, select **Hide** to close the toolbar.

Using the Find dialog

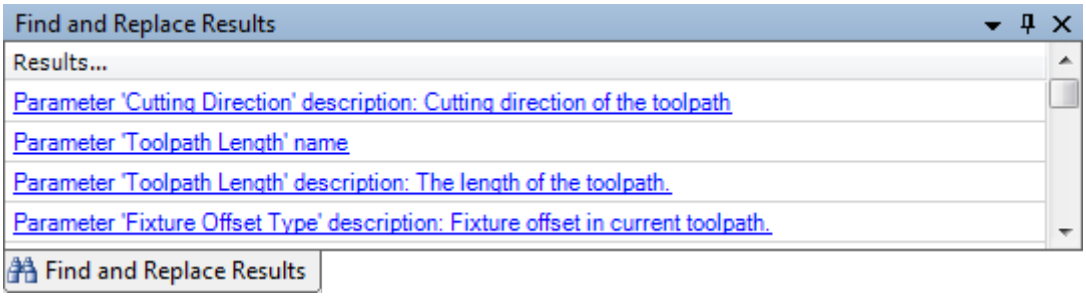
Use the **Find** dialog to search the option file for a specified text string.

To search the option file:

1. Click the **Find** button in the toolbar, or select the **Edit > Find** menu option. The **Find** dialog is displayed.



2. In the **Find What** box, enter the text you want to search for.
3. In the **Find For** field, select the type of search you want to make:
   - Select **Text** to search the option file for all instances of the specified **Find What** text.
   - Select an element to restrict the search to references to that type of element.
4. To restrict the search to text that matches the case of the **Find What** string, select **Match case**.
5. To prevent the search from finding instances of the string within words, select **Match whole word**.
6. Click **Find All**. The **Find Results** window lists all instances of the string found in the options file. Click a result to display the element in which the string was found.



View menu

When the PostProcessor tab is selected:

Switch to Editor - Displays the **Editor** tab.

Session Explorer - Displays the **Session Explorer**. Click the option again to hide the **Explorer** pane.

Output - Displays the **Output** pane. Click the option again to hide the pane.

When the Editor tab is selected:

Back - Displays the previous page you selected.

Forward - Displays the next page you selected.

Switch to PostProcessor - Displays the **PostProcessor** tab.

Commands - Displays the **Commands** tab in the **Explorer** pane. Click the option again to hide the tab.

Parameters - Displays the **Parameters** tab in the **Explorer** pane. Click the option again to hide the tab.

Structures - Displays the **Structures** tab in the **Explorer** pane. Click the option again to hide the tab.

Tables - Displays the **Tables** tab in the **Explorer** pane. Click the option again to hide the tab.

Formats - Displays the **Formats** tab in the **Explorer** pane. Click the option again to hide the tab.

Script - Displays the **Script** tab in the **Explorer** pane. Click the option again to hide the tab.

Item Properties - Display the **Item Properties**. Click the option again to hide the properties.

Item Reference Properties - Display the **Item Reference Properties**. Click the option again to hide the properties.

Preview Window - Display the **Preview** window. Click the option again to hide the window.

Find Results Window - Display the **Find Results** window. Click the option again to hide the window.

Tools menu

When the PostProcessor tab is selected:

Process - Creates NC programs for the selected CLDATA file.

Process All - Creates NC programs for all the CLDATA files.

Process as Debug - Creates a debugged NC program.

Option File Report - Creates an HTML report containing the file history, settings, commands and parameters.

PostProcessor Settings - Enables you to change your Delcam PostProcessor user name, and to specify your interface preferences.

When the Editor tab is selected:

Option File Report - Creates an HTML report containing the file history, settings, commands and parameters.

PostProcessor Settings - Enables you to change your Delcam PostProcessor user name, and to specify your interface preferences.

Window menu

Create Empty Window - Adds a new tab to the **View** pane. Select an element in the explorer to display it in the tab.

Different tabs enable you to display different elements of the session in the same window. Click a tab to display its contents.

Close Active - Closes the currently displayed tab.

Close All - Closes all tabs.

Close All But Active - Closes all tabs except the tab that is currently displayed.

Reset Layout - Returns the user interface to its default layout.

Help menu

Help Contents - Opens the reference help for Delcam PostProcessor.

Getting Started - Opens the introductory help for Delcam PostProcessor.

Check for Updates - Check the Delcam website for new versions of Delcam PostProcessor.

Subscribe to the PostProcessor Newsletter - Subscribe to the Delcam PostProcessor online newsletter.

Visit User Forum - Opens the User Forum page on the Delcam website.

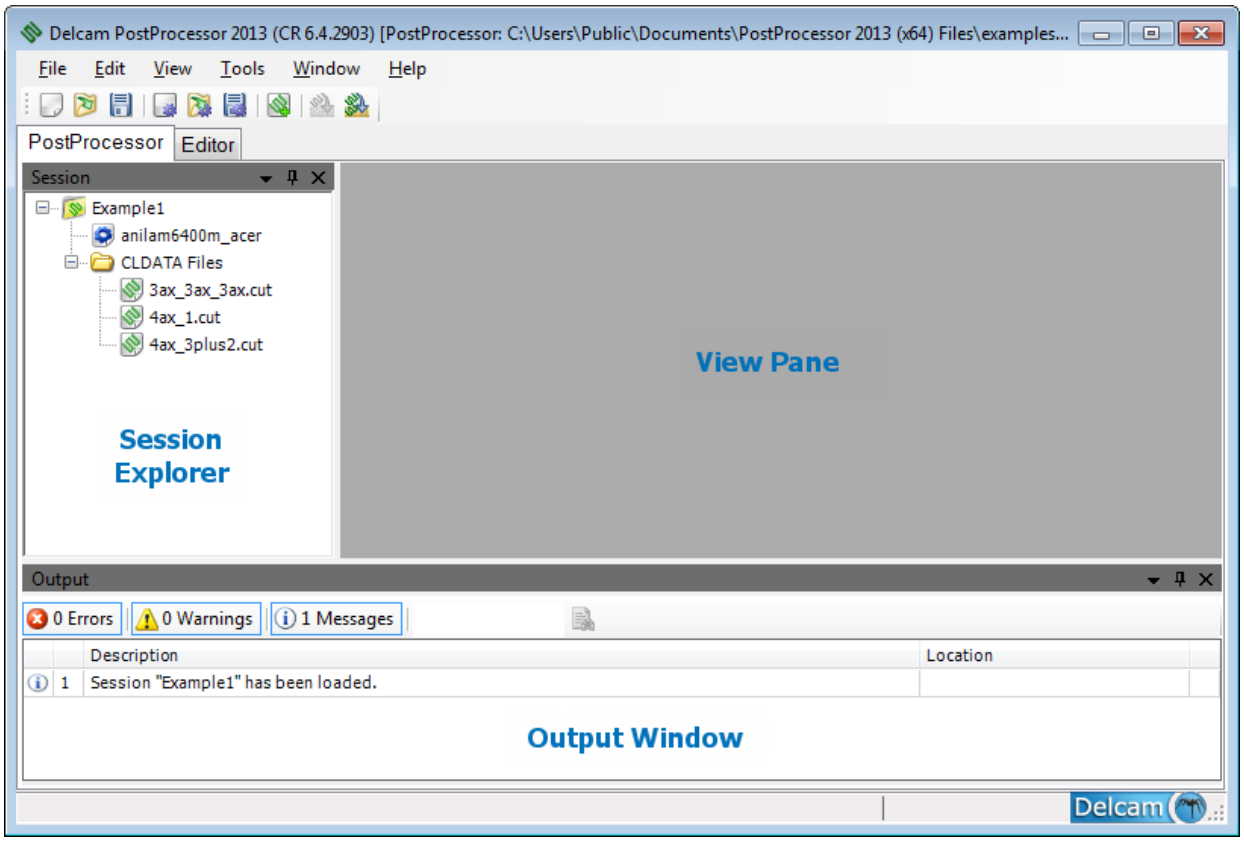
Delcam Home Page - Opens the Home page on the Delcam website.

About Delcam PostProcessor - Displays information about this version of Delcam PostProcessor including:

- copyright information;
- details of the files installed;
- licensing details.

Using the PostProcessor

The **PostProcessor** tab enables you to generate controller-specific tapefiles for NC/CNC machine tools from CLDATA files. It is opened by default when you start Delcam PostProcessor. You can also display it by clicking the **PostProcessor** tab, or by selecting the **View > Switch to Postprocessor** menu option.



The **Postprocessor** tab includes the following sections:

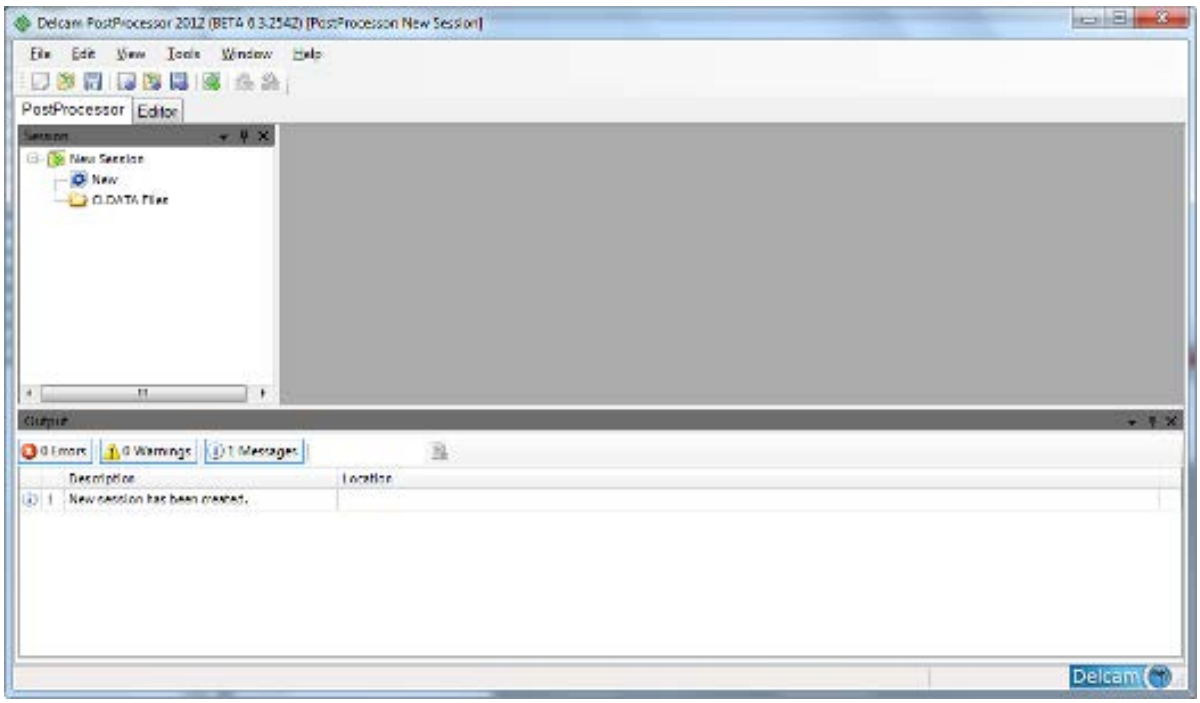
- **Main Toolbar** and **Menus** - These enable you to create, open and save projects; to create, load and save option files; to load and delete CLDATA files; to perform postprocessing; and to specify the postprocessing settings.
- **Session Explorer** - This lists the name of the current session, the option file used to postprocess the CLDATA files, and the CLDATA files to be converted using the specified option file. If the CLDATA files have been postprocessed in this session, it also lists the output (or tapefiles) that have been generated from them.
- **View Pane** - This displays the contents of the selected tapefile. When no tapefile is selected, the pane is empty.
- **Output Window** - This displays any error, warning and messages produced during postprocessing.

Starting Delcam PostProcessor

To start **Delcam PostProcessor**:

- Click the Windows **Start** button, select **All Programs**, and select the **Delcam PostProcessor** icon from its installation folder.
- Double-click the **Delcam PostProcessor** icon on your desktop.
- Double-click an option file or session file in Window Explorer.

A new session is opened:

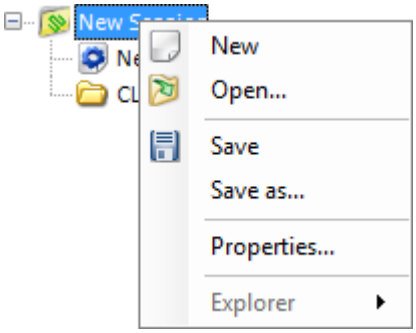


To resize the frames, position the cursor over a frame border and, when the cursor changes to a double-headed arrow, drag the border to a new position.

Context menus

Delcam PostProcessor contains several context menus that are displayed when you right-click an item in the user interface:

Session menu



New - Starts a new session.

Open - Opens an existing session.

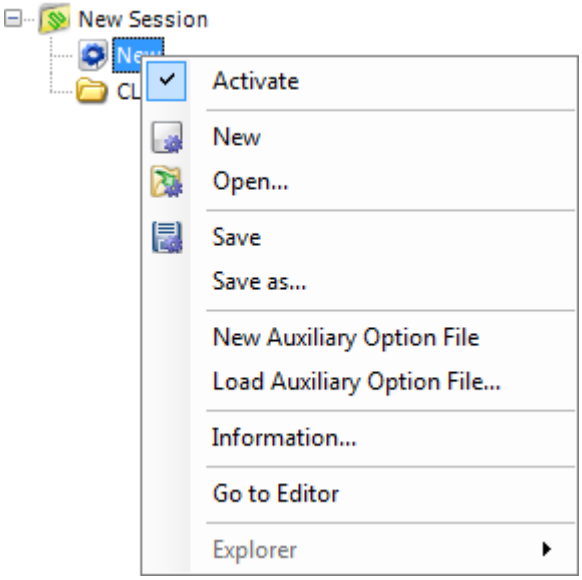
Save - Saves the current session.

Save As - Saves the current session under a different name.

Properties - Specifies the session properties.

Explorer - Displays shell context menu.

Option file menu



New - Creates an empty option file.

Open - Loads a different option file.

Save - Saves the current option file.

Save As - Saves the option file under a different name.

New Auxiliary Option File - Creates new auxiliary option file.

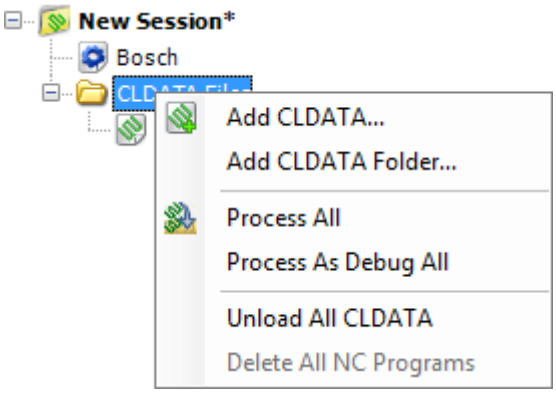
Load Auxiliary Option File... - Loads new auxiliary option file.

Information... - Display information about option file.

Go to Editor - Displays the Editor tab.

Explorer - Displays shell context menu.

CLDATA folder menu



Add CLDATA - Adds a CLDATA file to the session.

Add CLDATA Folder - Adds CLDATA files from the specified folder.

Process All - Postprocesses all CLDATA files in the session.

Process As Debug All - Postprocess all the CLDATA files with debug information.

Unload All CLDATA - Removes all CLDATA files from the session.

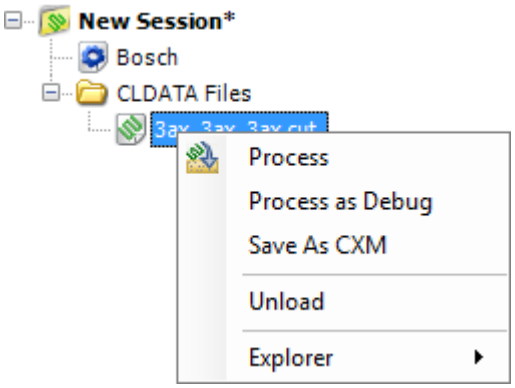
Delete All NC Programs - Removes all tapefiles from the session.

Create Folder - Creates a CLDATA folder. See CLDATA Folders.

Remove Folder - Removes the CLDATA folder selected. See CLDATA Folders.

Rename - Renames the CLDATA folder selected. See CLDATA Folders.

CLDATA file



Process - Postprocesses the selected file.

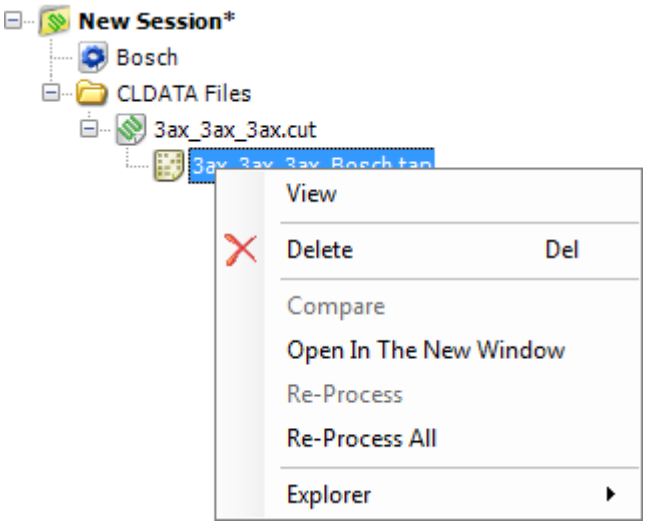
Process As Debug - Postprocesses the selected file in Debug mode.

Save As CXM - Saves the selected file as a ***.cxm** file.

Unload - Removes the cutfile from the session.

Explorer - Displays shell context menu.

Tapefile and debug file menu



View - Displays tapefile.

Delete - Removes tapefile from the session and hard drive.

Compare - Displays changes in the tapefile from previous postprocessing.

Open In The New Window - Displays tapefile in the new window.

Re-Process - Postprocess CLDATA again with modified option file.

Re-Process All - Postprocess all CLDATAs with modified option file.

Explorer - Displays shell context menu.

Session

Creating a new session

To create a new session, click the **New Session** button, or select the **File > New > Session** menu option.

If you have any unsaved changes, a message asks whether you want to save them. Click:

Yes to save your changes and close the session.

No to close the current session without saving your changes.

Cancel to cancel the close.

Opening an existing session

To open an existing session:

1. Select the **PostProcessor** tab.
2. Choose one of the following options:
   - Click the **Open File** button or select the **File > Open > Session File** menu option. In the **Open** dialog, select the ***.pmp** file and click **Open**.
   - Double-click the session file or folder in Windows Explorer.
   - Select the session file or folder in Windows Explorer, and drop it on the **Session Explorer**.

If you have any unsaved changes, a message asks whether you want to save them. Click:

Yes to save your changes and close the session.

No to close the current session without saving your changes.

Cancel to cancel the close.

Saving a session

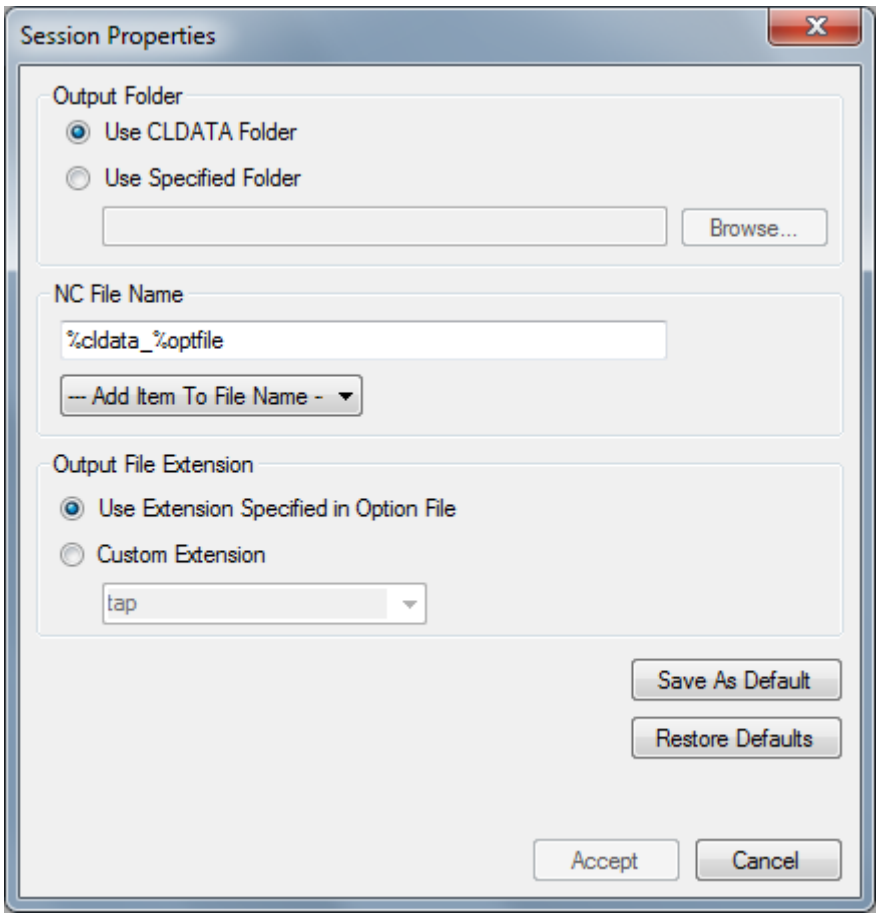
To save your session:

1. Click the **PostProcessor** tab.
2. Choose a save option:
   - To save the session under its current name, click the **Save Session** button on the toolbar, or select the **File > Save Session** menu option.
   - To save the session under another name, select the **File > Save Session as** menu option.
   - To save all session files (option files, CLDATA files and NC program files) to one folder, select the **File > Save Session Folder** menu option.

     The location and filename are displayed in the Title bar of the Delcam PostProcessor without an asterisk.

Session properties

Use the **Session Properties** dialog to specify the location and name of tapefiles created using this session.



To specify the session properties:

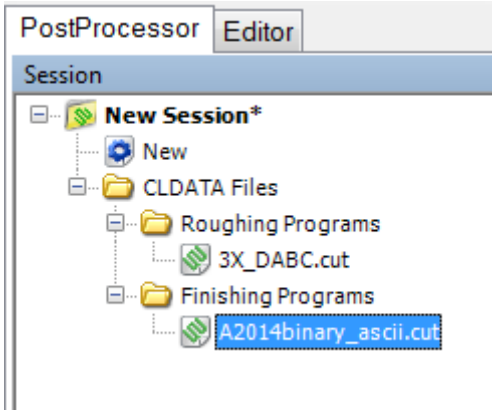
1. Select the **File > Session Properties** menu option, or select **Properties** from the **Session Explorer** context menu to open the **Session Properties** dialog.
2. Specify the location in which you want to save the postprocessed tapefiles. Select:
   - **Use CLDATA Folder** - to save tapefiles to the folder that contains the CLDATA files.
   - **Use Specified Folder** - to save tapefiles to the specified folder. Enter the folder path in the adjacent field, or click **Browse** to locate the folder.
3. In the **NC File Name** box, specify the names of the tapefiles to be created from this session. To create a unique name for each tapefile, include one or more entries from the **Add item to file name** list. Select:
   - **Option file name** to insert the name of the option file. The **%optfile** variable is added to the name box.

- **Cldata name** to insert the name of the CLDATA file. The **%cldata** variable is added to the name box.
- **Date** to insert the date in dd_mm_yy format. The **%date** variable is added to the name box.
- **Time** to insert the time in hh_mm_ss format. The **%time** variable is added to the name box.
- **Counter** to insert a unique number in the file name. The number is incremented each time you save the file. The **%counter** variable is added to the name box.

4. Specify the extension for the tapefiles created by this session. Select:
   - **Use extension specified in option file** to generate all output files using the extension specified in the option file.
   - **Custom Extension** to specify the extension in the adjacent box. The extension can contain any characters except **. , ? * : ; = + \ | / "**.
5. Click **Save as Default** if you want to use the current settings as the default session properties for new sessions.
6. Click **Restore Defaults** to if you want to reset the properties to the default settings.
7. Click **OK** to save the preferences in the session file.

CLDATA Folders

The CLDATA files of a session can be arranged in folders. Those folders may not exist on the disc but are stored in the session.



In order to create a folder do the following:

- Select **CLDATA Files** in the session tree.
- Click **Create Folder**.
- Press **F2** and type the name of the folder.
- Press **Enter**.

If you want to rename a folder select it, click **Rename** from the context menu (or press **F2**) and type its new name.

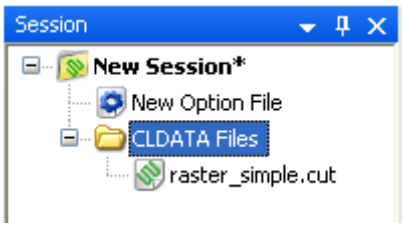
If you want to remove a folder select it, click **Remove** from the context menu (or press **Del**).

When you do **Save Session Folder** the directory tree is automatically created at destination according to CLDATA folders in the session.

Loading a CLDATA file

To load a cutfile or ***.cxm** file into a session:

1. Choose one of the following methods:
   - Click the **Add CLDATA Files** button on the toolbar.
   - Select the **File > Add CLDATA** menu option.
   - Right-click the **CLDATA files** folder in the **Session Explorer**, and select **Add CLDATA** from the context menu.
2. In the **Open** dialog, select the CLDATA file and click **Open**. The file is displayed in the **CLDATA files** folder of the **Session Explorer**. For example:



You can also drag CLDATA files from Windows Explorer and drop them on the **PostProcessor** tab.

To load several CLDATA files from a folder, select the **File > Add CLDATA Folder** menu option. In the **Browse for Folder** dialog, select the folder and click **OK** to load its files into the **Session Explorer**.

To create a **.cxm** file from a cutfile, in the same folder and with the same prefix, right-click the file in the **Session Explorer** and select **Save as CXM** from the context menu.

Postprocessing CLDATA files

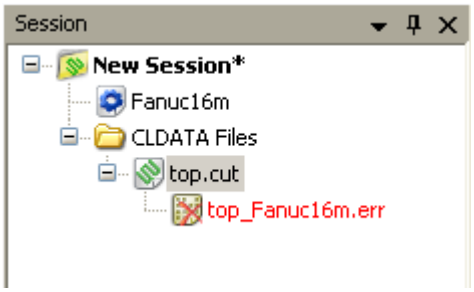
To generate a tapefile from a CLDATA file:

1. Select the **PostProcessor** tab.
2. Load and edit the option file for the machine controller for which you want to generate a tapefile.
3. Right-click the **CLDATA files** folder in the **Session Explorer** and select **Add CLDATA** from the context menu. In the Open dialog, select the files you want to postprocess.
4. Choose one of the following options:
   - Click the **Process** button.
   - Select the **Tools > Process** or menu option.
   - Right-click the cutfiles and select **Process** from the context menu.

To postprocess all the CLDATA files you have loaded , click the ***Process All*** *button or select the* ***Tools - Process All*** *menu option.*

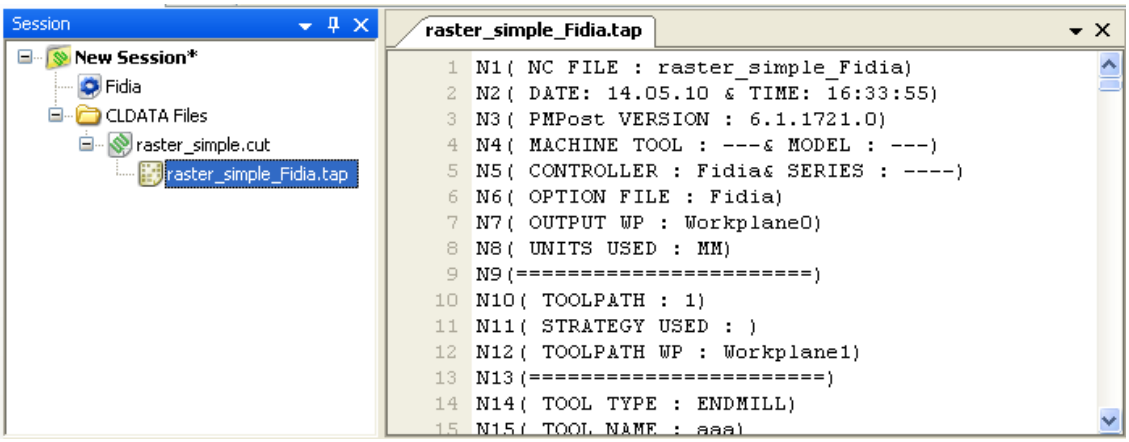
When postprocessing is complete, the tapefile is added to the **CLDATA files** folder of **Explorer**:

- If no errors were found during processing, Delcam PostProcessor creates a tapefile with an extension of **.tap**.
- If Delcam PostProcessor finds any errors during processing, the tapefile is saved with an extension of **.err**, and the name is displayed in red. For example:

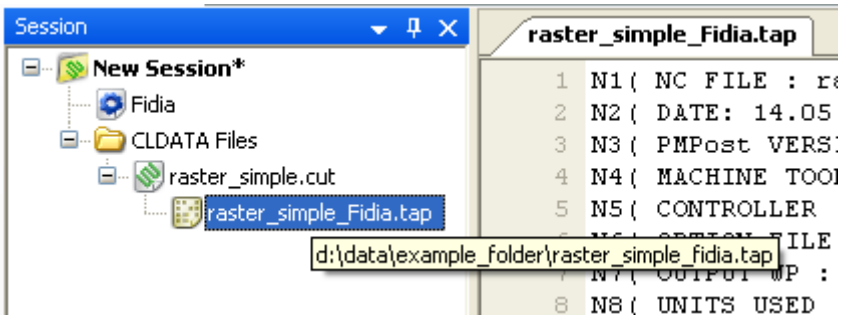


If the processed part cannot be retrieved, no files are generated.

5. Click the tapefile to display the generated NC program. For example:



6. If you want to display the location of the tapefile, position the mouse cursor over the file's entry in the **Session Explorer**. For example:



Changes to an option file are used during postprocessing even if they have not been saved.

If you change an option file after postprocessing, a red cross is displayed against the output file to indicate that it is no longer current.

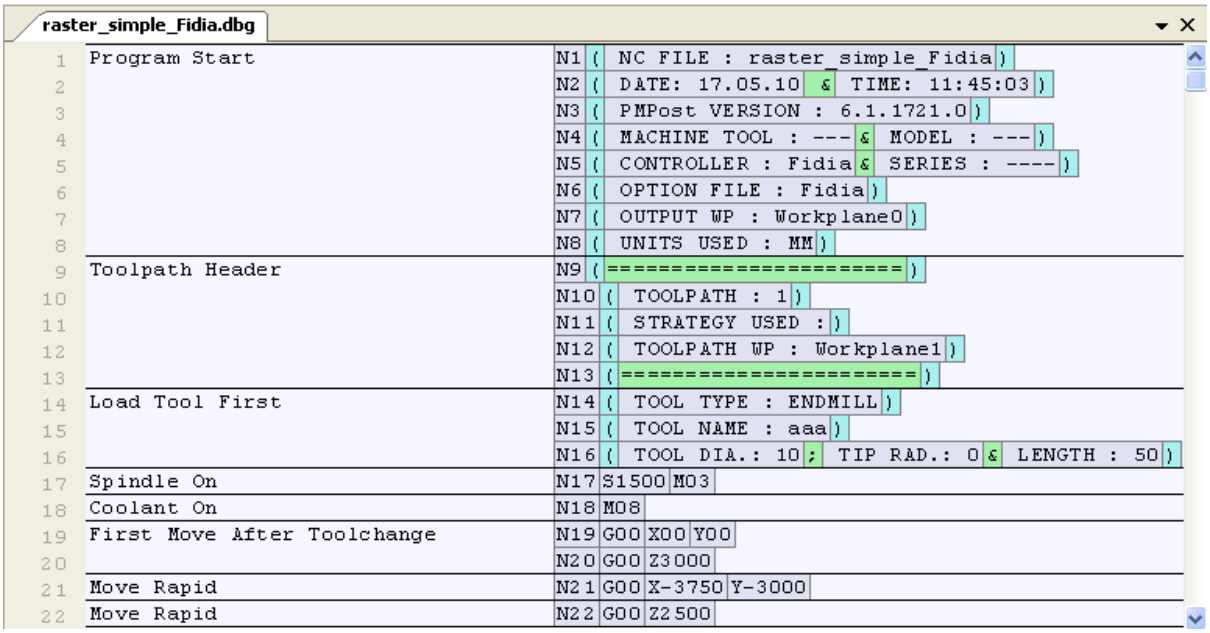
Debug mode

Debug mode produces an NC Program file with additional information, so you can see the extents and names of the blocks, and the parameter settings used. They cannot be used for machining.

Debug postprocessing is not recommended for large source files.

To generate a debug file, select **Process As Debug** from the cutfile's context menu in the **Session Explorer**. Debug files have an extension of ***.dbg** and are indicated by the  icon.

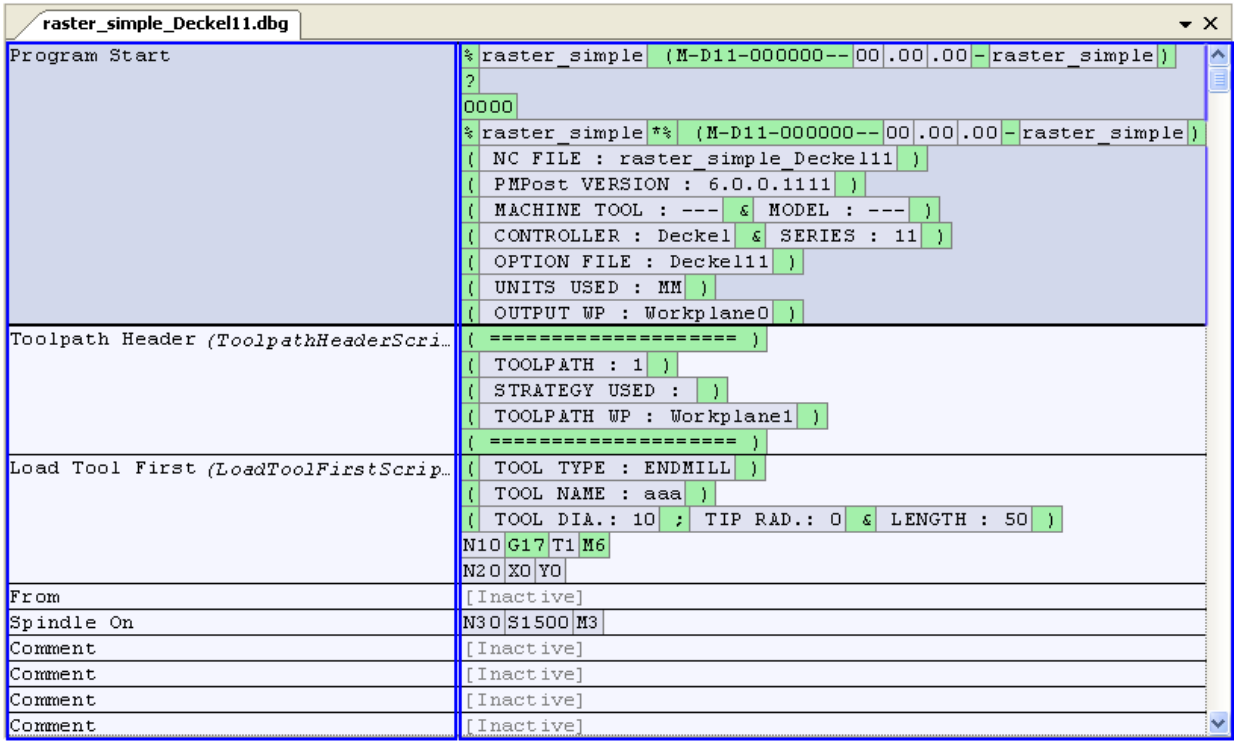
To display the debug file, click its entry in the **Session Explorer**.



The file has two interdependent representation of processed file:

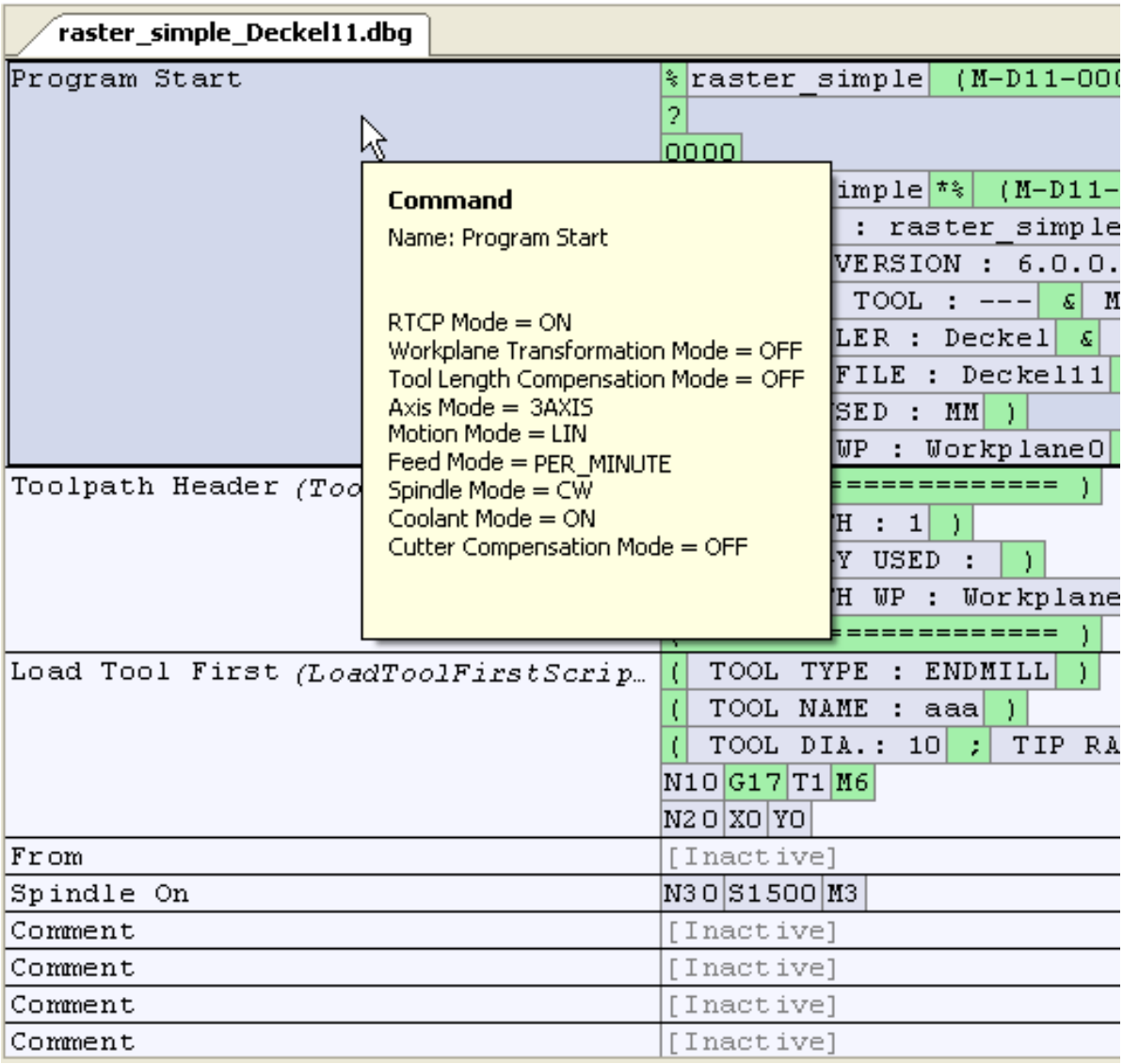
- **Responses order** - the order is defined by sequence of commands responses during postprocessing. The order involves all commands with scripts functions executed.
- **NC code order** - the order is simple generated NC code.

The view allows to trace links between response and its output (NC code).

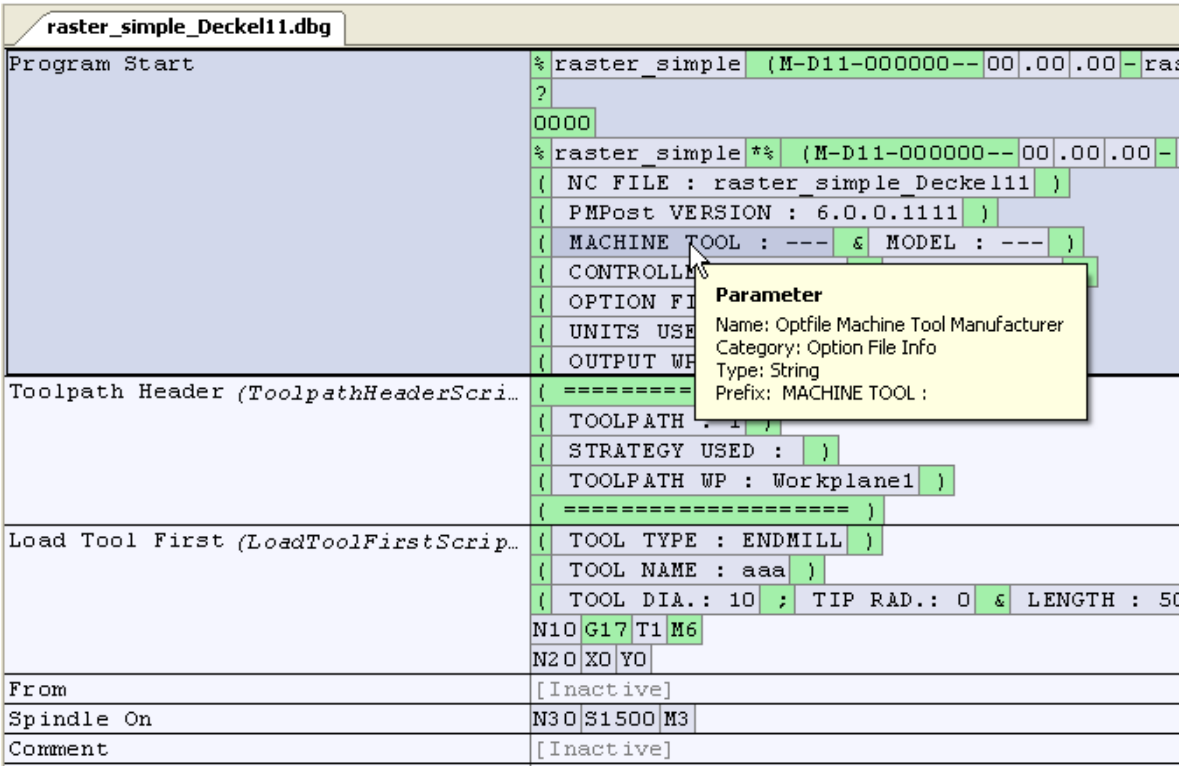


Responses order NC code order

To display the name and **Controller Switches** of the command that produced a section of the code, move the mouse over **Response order** section of the debug file. Double-click the command to display the command .



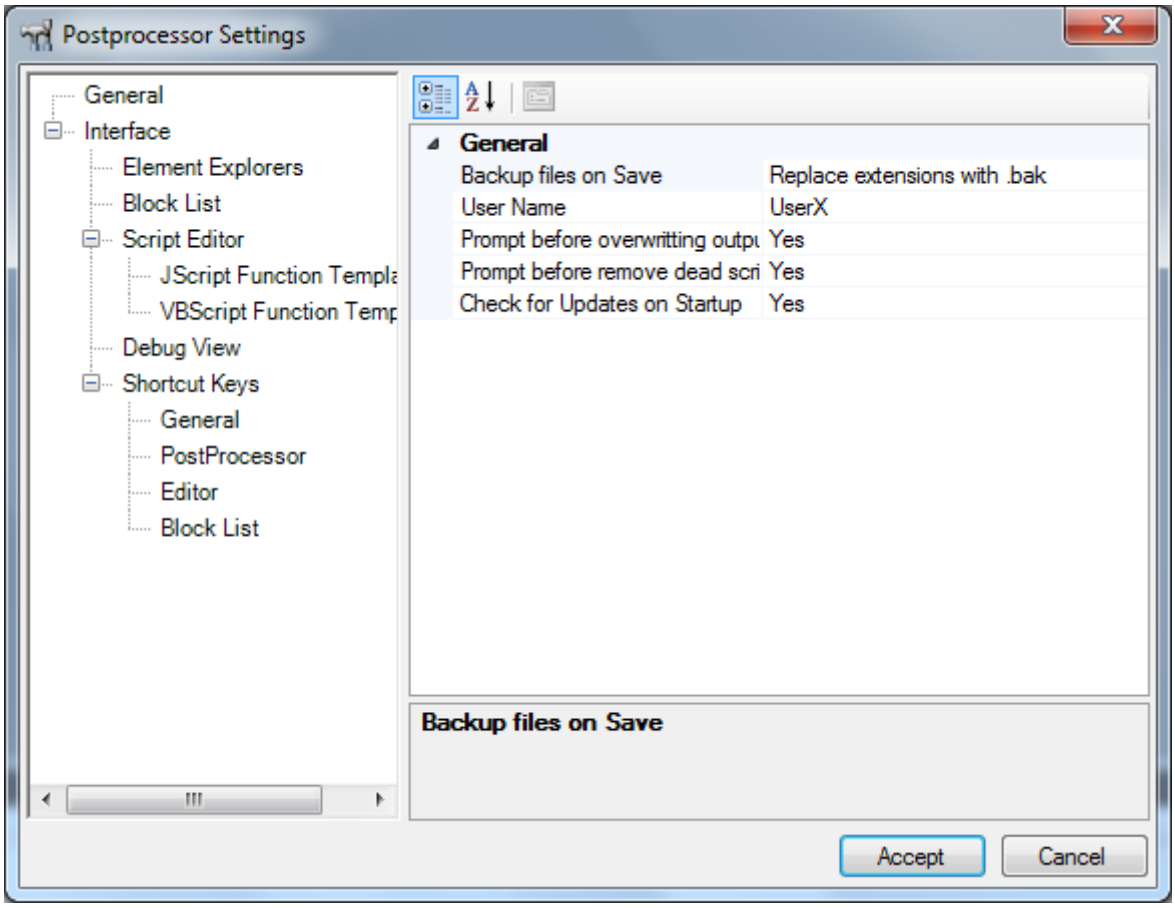
To display the **name and attributes of the parameter** that included into selected command, move the mouse over **NC code order** section of the debug file. Double-click the parameter to display selected command and select block item associated with the parameter.



PostProcessor Settings

Use the **PostProcessor Settings** dialog to control the behaviour of the user interface. To open the dialog, select the **Tools > PostProcessor Settings** menu option.

General settings



Backup Files on Save - Specifies what happens when you save an option file. Select:

- **No Backup** to overwrite the original file with the new version.
- **Replace extension with .bak** to rename the original option file with an extension .bak and save the option file under the original name.

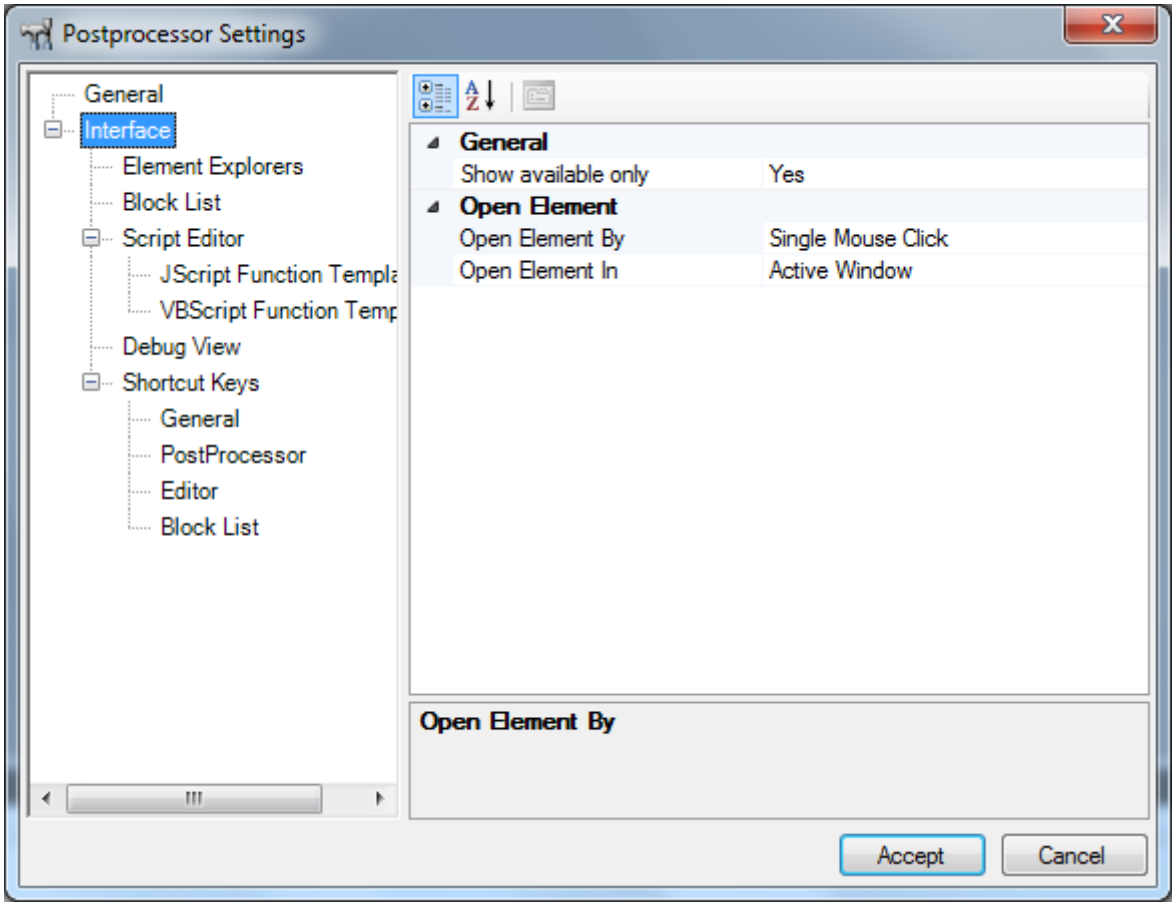
Users Name - Specifies the user name used to track changes in the **Editor**. Select:

- **Login Name** to use the user name of the person who is currently logged in.
- to specify the user name in the adjacent field.

Prompt before overwriting output files - Specifies whether you want Delcam PostProcessor to ask for confirmation before overwriting an existing cutfile file. Select:

- **Yes** to prompt you for confirmation.
- **No** to overwrite cutfiles without confirmation.

Interface settings



Open Element specifies your preferences for opening an element:

- **Show available only** - Whether to filter elements according to option file properties. If No, all existing elements are shown.
- **Open element by** - Specifies the number of mouse clicks required to open an element. Select:

  Single mouse-click to open elements using a single mouse-click.

  Double mouse-click to open elements using a double mouse-click.
- **Open element to** - Specifies where an element is opened. Select:

  Open an element in active window to open each element in the active window. Any previously displayed element is replaced.

  Open an element in new window to open each element in a new window.

Script Editor specifies your script-editing preferences:

- **Word Wrap** - Specifies how long lines are treated in the script window. Select:

  Yes to display any text that extends beyond the right edge of the window on the next line.

  No to display any text that extends beyond the right edge of the window by scrolling.
- **Tab Size** - Specifies the size of tab character in spaces.

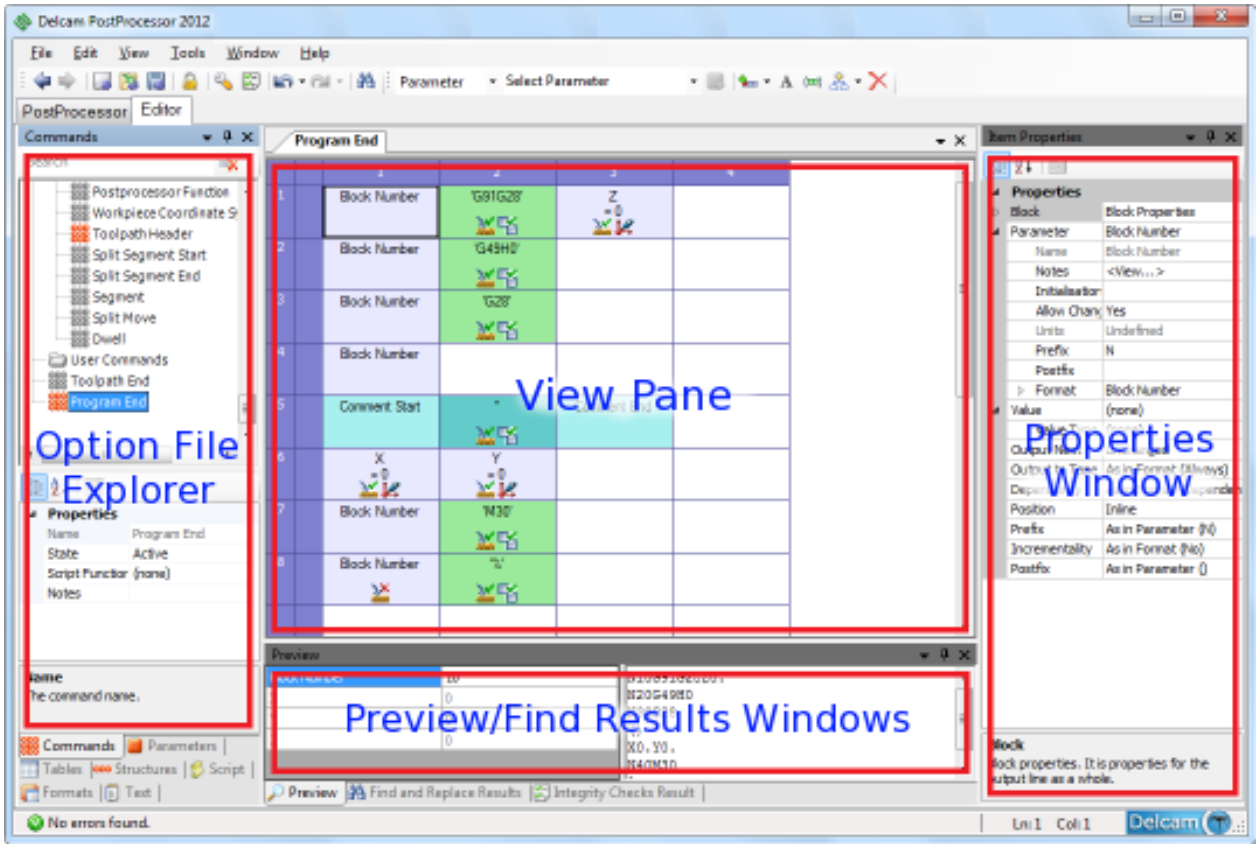
- **Indent** - Specifies how the Tab key works in the Script Editor. Select:

  Keep Tabs to insert a tab character each time the Tab key is pressed. The **Tab Size** field specifies the length of each tab.

  Insert Spaces to insert space characters each time the Tab key is pressed. The **Tab Size** field specifies how many spaces are inserted by each key press.

Using the Editor

The **Delcam PostProcessor Editor** enables you to create and edit the option files that convert CLDATA files to controller-specific programs. To display the **Editor**, click the **Editor** tab, or select the **View > Switch to Editor** menu option.



The **Editor** tab contains the following sections:

- **Main Toolbar** and **Menus** - These enable you to create, open and save option files; to specify option file properties; to protect option files; and to generate option file reports.
- **Option File Explorer** - This displays the elements that can be used in an option file. It contains the following tabs:

Commands

Parameters

Tables

Structures

Scripts

Formats

A toolbar is displayed at the top of each tab to enable you to add and move items. A **Properties** section at the bottom of each tab displays the attributes of the selected item in the list.

*Use the options in the **View** toolbar to hide or display tabs in the Option File Explorer.*

- **View Pane** - This displays contents of selected element in Option File Explorer.
- **Properties Window** - This displays properties of the selected item in the **View** Pane.
- **Preview window** - This shows a preview of the selected command.
- **Find Results window** - This lists the results of your last **Find** operation. Click a link to display the element in which the search string has been found.

Creating an option file

The easiest way to create an option files is to:

- load the option file for the required machine controller, edit the file, and then save it under the same name.
- load an option file that is similar to what you require, then customize it and save it under a different name.

Before creating an option file, you require the following information:

Machine Tool Manual

- Coordinate system and direction movement
- Axis travel limits and (optional) home positions
- Axis feed rate limits
- Head limits/ranges.

Machine Control and the Programmer's manual

- Machine preparatory and auxiliary codes
- Address register formats/limits
- G-functions or equivalent
- Head limits/ranges.

To create a new option file after starting or opening a **Delcam PostProcessor** session:

- Select the **File > New > Option File** menu option.
- Click the **New Option File**  button on the toolbar.

  If the current session contains any unsaved changes, you are prompted to save them.

The new option file is displayed in the **Session Explorer**. If you have created a new file (as opposed to opening a template), it is called **New1**, or **New2** if it is the second new file in the session, and so on. You can rename it when you first save it.

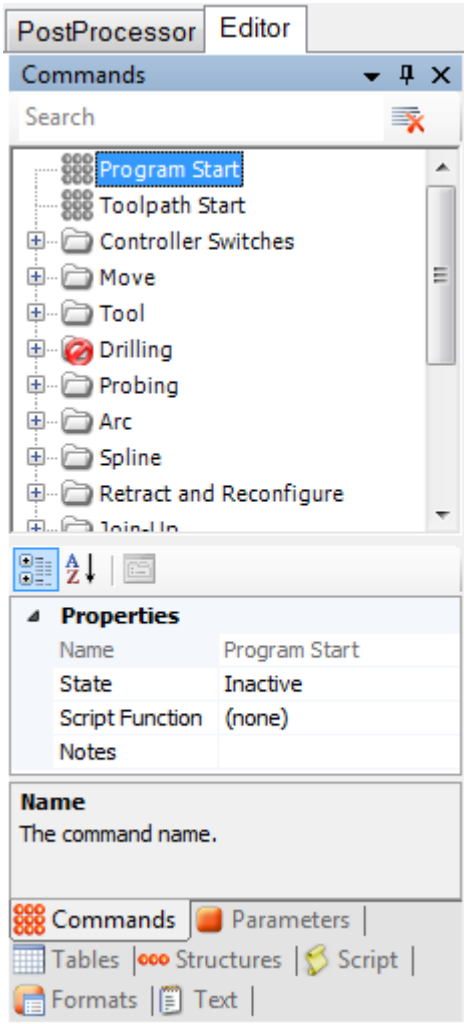
Click the **Editor** tab and edit the option file. If you are using an existing file as a template, rename it by selecting the **File > Save Option File As** menu option, or clicking  on the toolbar. This displays the **Save option file as** dialog.

Option File Explorer

The **Option File Explorer** pane in the **Editor** tab enables you to select the different parts of the option file. It enables you to navigate to and work with the elements that make up the option file:

- Commands
- Parameters
- Tables
- Structures
- Script
- Formats

Select a tab at the bottom of the **Explorer** to choose the element you want to work with.



Use the View menu options to show and hide the tabs displayed in the Explorer.

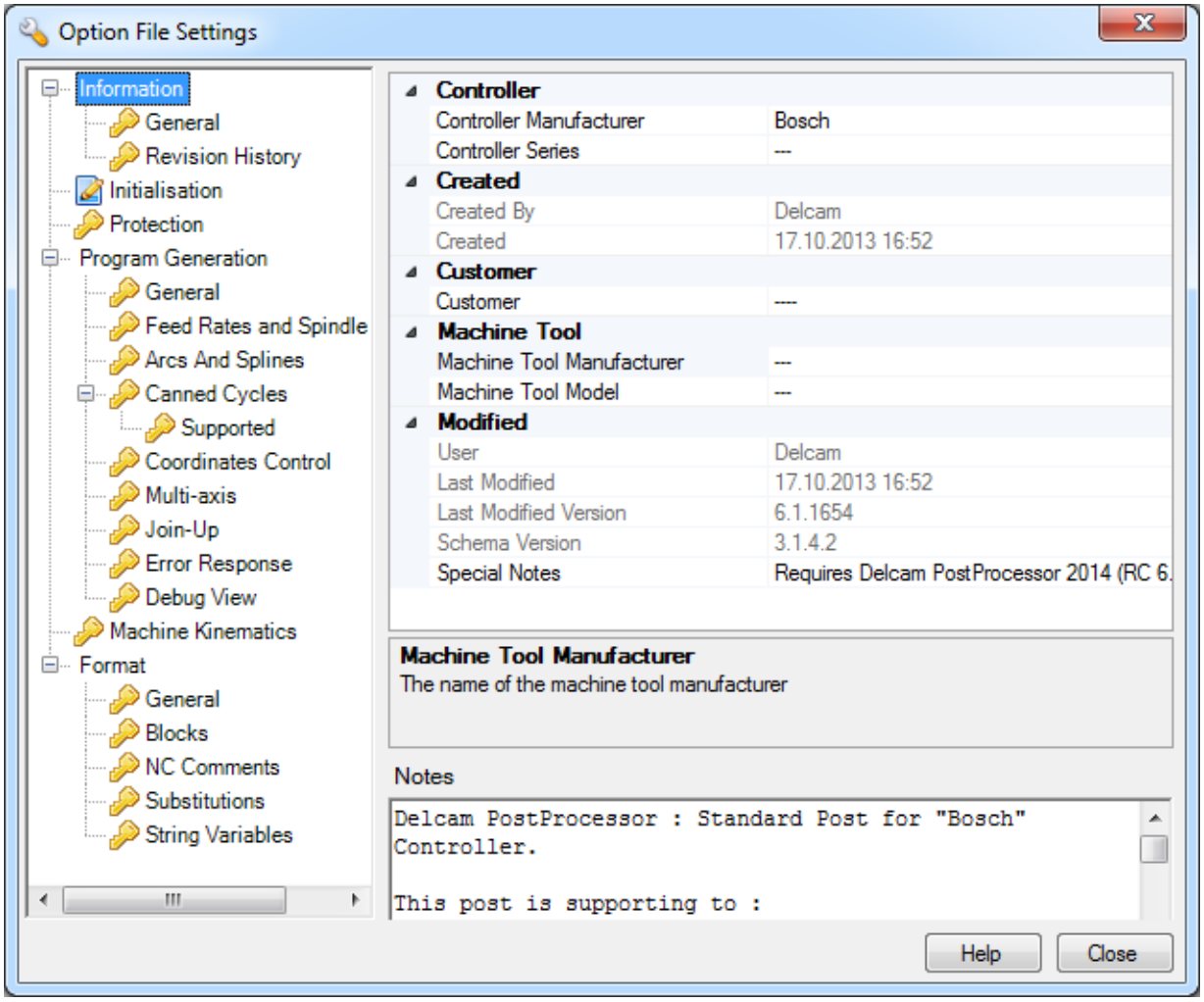
The pane contains following sections:

- **Search toolbar** - allows to filter by text in Elements Explorer.
- **Element Explorer** - lists the items available for the currently selected element.
- **Properties** - allows to view and edit properties for selected object in elements Explorer.

Click any of objects listed in the Explorer to display more information in the **Properties** window and the **View** pane.

Using the Option File Properties dialog

Use the **Option File Settings** dialog to specify the controller-specific parameter settings for the option file you are working with. To open the dialog, click the **Option File Properties** button in the toolbar, or select the **File > Option File Properties** menu option.



Click an option in the left panel to view and edit the parameters.

Informationdisplays common information about the option file and summary of changes made during option file creation and editing: Select:

- **General**to display common information about the option file.
- **Revision History**to displays summary of changes made during option file creation and editing.

Initialisation specifies the settings used at the start of postprocessing.

Program Generation provides overall control of generation settings for current controller. Select:

- **General** to specify general program generation settings.
- **Feed Rates** to set the minimum and maximum allowable feed rates for a controller.

- **Arcs And Splines**to enable arc or spline interpolation if it is supported by the controller. This page also enables you to set conditions when arc linearisation (when a sequence of small linear moves is written out instead of the arc) is required in an NC program.
- **Drilling** to specify the drilling cycle parameters. This page also includes settings for expanding helical and threading cycles if they are not supported by the controller.
- **Coordinates Control** to enable and disable **Automatic Coordinate Control**. When it is enabled, you can select machine profiles and presets to define the associated **Toolpath Axis Modes (3-Axis**, **3+2** or **Multi-Axis)**.
- **Multiaxis Configuration** to specify multi-axis attributes. This option is available only when **4-Axis** or **5-Axis** is selected in **Machine Kinematics**.
- **Error Response** to specify the actions to the **PostProcessor** responses.

Machine Kinematics enables you to specify the number of axes in the machine tool (3-Axis, 4-Axis and 5-Axis) and the number, type and location of the machine's tilting axes. There are no tilting axes for a 3-Axis machine, one tilting axis for a 4-Axis machine and two tilting axes for a 5-Axis machine.

Format controls the output format for NC programs. Select:

- **General** to specify general formatting options.
- **Blocks** to specify the format for the output of blocks to NC programs.
- **NC Comments** to specify the format for the output of comments to NC programs.
- **Substitutions** to automatically replace characters in the output file.
- **String Variables** to automatically replace strings in the output file.

By default most settings is locked  to edit in protected option files. But you may allow to edit  any settings using context menu or just clicking on node icon.

Information properties

The section contains information about the option file itself and lists the changes that have been made to the option file:

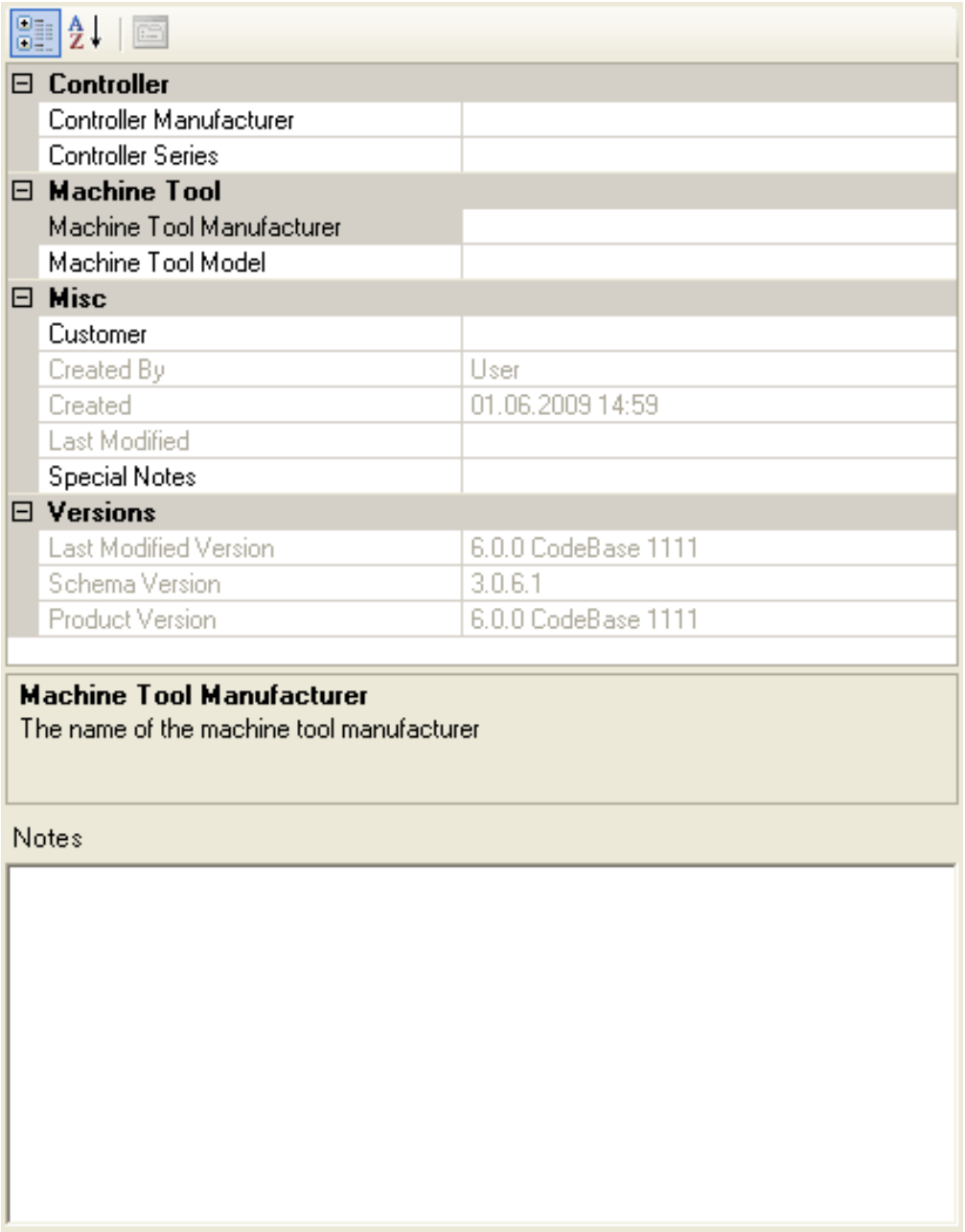
- **General**
- **Revision History**
- **Information - General**
- **Information - Revision history**

Information – General

Use the **General** page to record general information about the option file and how it is used.

To open the **General** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.
2. In the **Option File Properties** dialog, select **Information - General**.



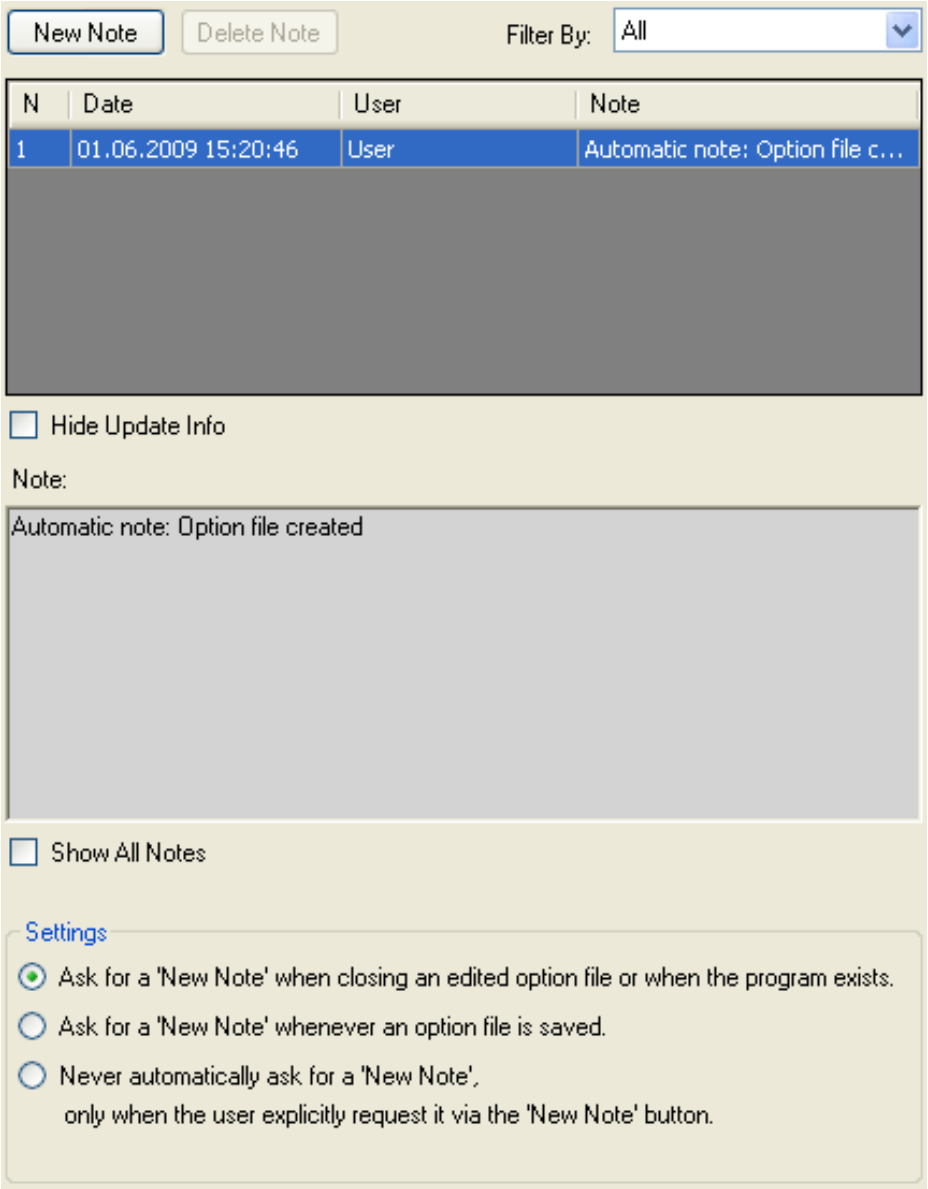
Parameter	Description
Controller Manufacturer	Specifies the name of the manufacturer of the machine controller.
Controller Series	Specifies the machine controller series and version.
Machine Tool Manufacturer	Specifies the name of the machine tool manufacturer.
Machine Tool Model	Specifies the machine tool model and version.
Customer	Specifies the customer name.
Created By	Specifies the author's name. This field is completed automatically.
Created	Specifies the creation date of the option file. This field is completed automatically.
Last Modified	Specifies the date of the last change. This field is completed automatically.
Special Notes	Use this box to record extra information about the option file, such as when it should be used.
Last Modified Version	Specifies the version of **Delcam PostProcessor** used when the option file was last modified. This field is completed automatically.
Schema Version	Specifies the current schema version of the option file. This field is completed automatically.
Product Version	Specifies the current version of **Delcam PostProcessor**. This field is completed automatically.

Information - Revision history

The **Revision History** page summarizes the changes made to the option file. Use it to record information about the option file, such as changes made to the file and the reasons for the changes.

To open the **Revision History** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.
2. In the **Option File Properties** dialog, select **Information - Revision History**.



To add a note to the revision history:

1. Click the **New Note** button.
2. In the **New Note** dialog, enter the information you want to record, and click **Accept**. The note is added to the option file history. To view the details of the note, select its entry in the list.

Configuration options

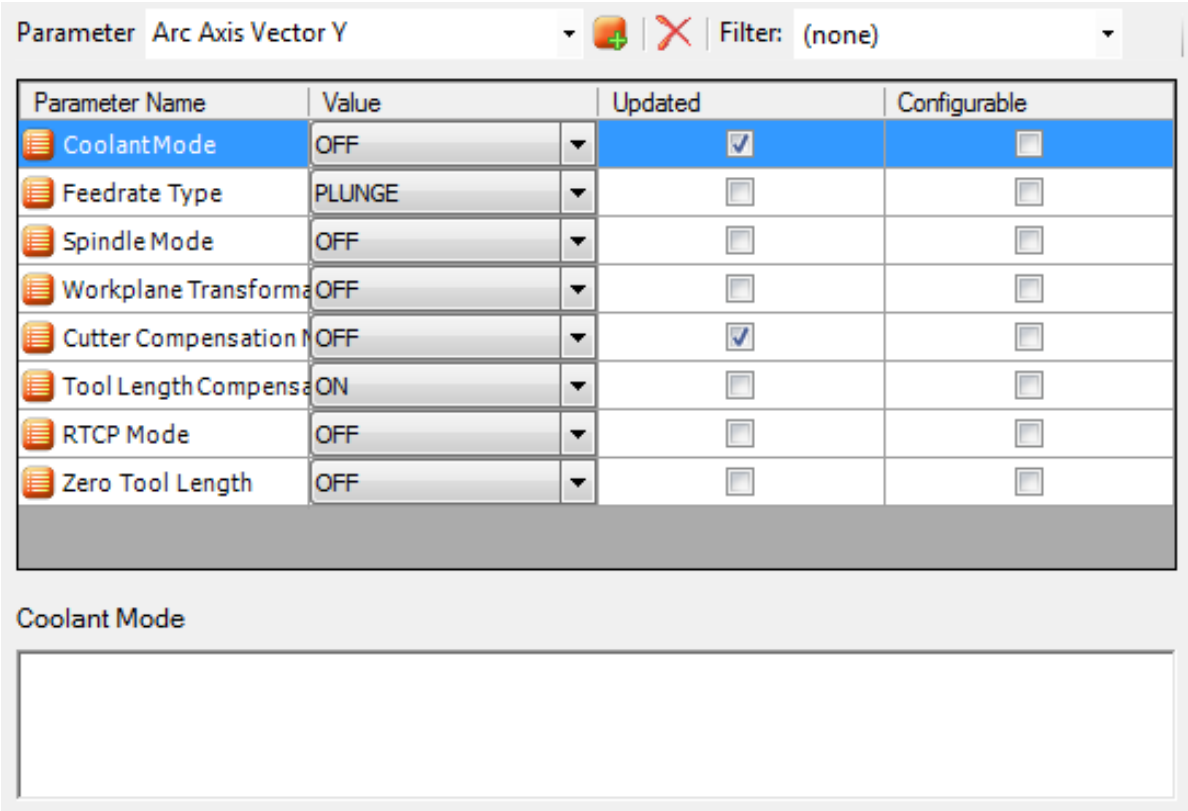
Hide Update Info Select this check box if you do not want to list information about automatic updates.

Set Show All Notes Select this check box to list all history entries in the **Note** box.

Settings Select an option to specify when the user is asked to update the history.

Initialisation

Use the **Initialisation** page to specify start values for parameters that must be set at the start of postprocessing, such as the **Coolant Mode** and **Cutter Compensation Mode**. The list of parameters to be initialised depends on the machine controller used.



To specify an initial value for a parameter:

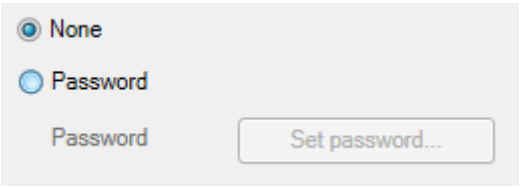
1. Select the parameter in the **Parameter** drop-down list and click the **Add** button. The following columns are displayed:
   - **Parameter Name** - specifies the name of the parameter.
   - **Value** - specifies the start value of the parameter.
   - **Updated** - controls the initial output of modal parameters
   - **Configurable** - controls visibility in protected option file
2. Enter the initial value for the parameter:
   - To change a **Real**, **Integer** or **String** parameter, overtype the **Value** box.
   - To change a **Group** parameter, click the down arrow and select an entry from the drop-down list.
3. If the parameter is modal set **Updated** property to specify its initial output.:
   - **Check** to output the value the first time the parameter is invoked (subject to the other settings in the format or command block). The status is reset to **No** following output of the value.
   - Leave **Unchecked** to prevent the value from being output the first time the parameter is invoked.

4. If you want to record extra information about the parameter, enter the information in the **Notes** box.

5. To remove a parameter from the list, select the parameter and click the **Remove** button.

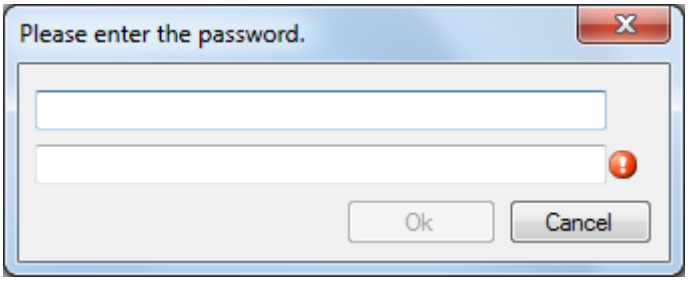
Protection

Use the **Protection** page to protect the option file from editing. When the option file is protected, you can postprocess CLDATA files as usual, but you can't open it in the **Editor** without entering a password it was protected with.



To protect the option file with a password:

1. Select the **Password** radio button.
2. Enter a password for the option file and conirm it in dialogue:



3. Save the option file. The file is password protected when it is next opened.

To unprotect the option file:

1. Select the **None** radio button.
2. Save the option file.

By default, only **Initialisation** page is available in a protected file. It's possible to enable any other page of option files settings.

1. Select a page you want to mark as editable in protected mode.
2. Right click on it.
3. Select **Editable**

Program Generation properties

The section provides overall control of generation settings for current controller:

- **Program Generation - General**
- **Program Generation - Feed Rates**
- **Program Generation - Arcs and splines**
- **Program Generation - Drilling**
- **Program Generation - Coordinates control**
- **Program Generation - Multi-axis configuration: general**
- **Program Generation - Multi-axis configuration: retract and reconfigure**
- **Program Generation - Error Response**
- **Angle Styles**

Program Generation – General

Use the **Program Generation - General** page to define the parameters that provide overall control while the **Postprocessor** is running.

To open the **General** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.
2. In the **Option File Properties** dialog, select **Program Generation - General**.

General	
NC Program Tolerance	Use CLDATA Tolerance
Output File Extension	tap
Error File Header	[This file has been processed with errors!!!]
Probing Support	No
Allow Repeated Tool Change	Yes
NC File Name	
Output File Name Template	%cldata_%optfile
Performance	
Access to XML Data	None
Splitting And Segmentation	
Split Criterion	None
Split Mode	Split NC Program
Split Segment Length	1
Split Part Postfix	_%part
Field Width of Part Number	1
Use Split Part Postfix for the first file	No
Units	
Option File Linear Units	Metric
Output Linear Units	Metric
Option File Angular Units	Degrees
Output Angular Units	Degrees

Parameter	Description
NC Program Tolerance	Specifies the overall tolerance used in trajectory calculations. This tolerance has the greatest impact on multi-axis linearisation but is also important if linearisation is switched off. Select: 💡 **Use CLDATA Tolerance** to use the tolerance from the CLDATA file as the overall tolerance. 💡 to specify the overall tolerance in the adjacent field. The final tolerance of the NC trajectory is the PowerMILL tolerance plus the selected **NC Program Tolerance**. So, if the PowerMILL tolerance is 0.025 and the **NC Program Tolerance** is set to **Use CLDATA Tolerance**, the final tolerance is 0.050.
Output File Extension	Specifies the suffix appended to output files (for example, **tap**). It cannot include any of the following characters: .,?*:;=+|/".
Error File Header	Specifies the header text of an erroneous output file.
Probing Support	Specifies whether the probing commands and parameters are visible. This option is for GUI only and has no direct impact on the output. It has no effect if **Show available ony** is **No** in **PostProcessor Settings**.
Allow Repeated Tool Change	Specifies whether the **Load Tool** command is invoked when a repeated tool change is encountered.
Option File Linear Units	Specifies the units used to input linear parameter values in the **Editor**. You can select: **Metric**, **Inches**, or **As Output Units**.
Output Linear Units	Specifies the units used to output linear parameters. You can select: **Metric**, **Inches**, or **As Input** (CLDATA) units.
Output Angular Units	Specifies the units used to output angular parameters. You can select: **Degrees** or **Radians**.

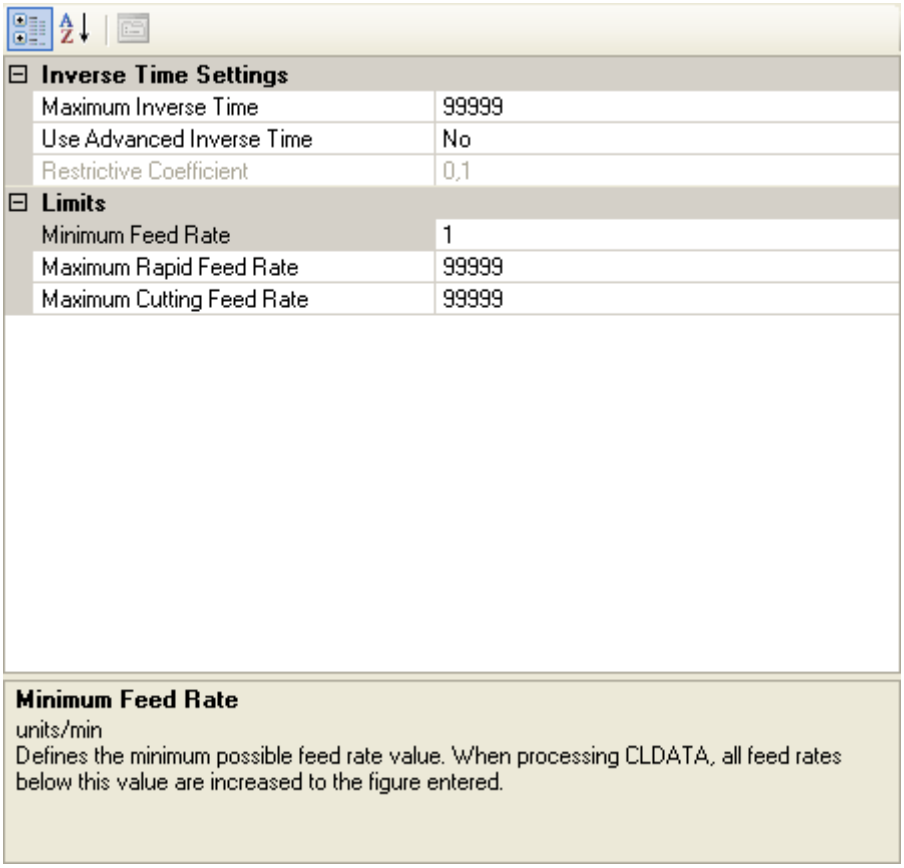
Parameter	Description
Access to XML Data	Specifies the type of access to the XML model. This option is only required for backward compatibility with old option files using **xml** and **node** objects in the script. This option must be set to **None** or **Automatic** for the new option files. • **None** to disallow XML access. Recommended. • **Automatic** to allow the **Postprocessor** to select the appropriate XML access according to its analysis of the option file. • **Partial** to allow the **Postprocessor** to use simplified access to XML. The simplified model reduces memory usage by including all commands except moves. This mode is recommended when processing a large CLDATA file with option file using the "xml" object in the script. • **All Data** to access whole XML model including move commands, such as **Move Linear** and **Move Rapid**. This mode is required only when the "node" object is accessed in the script. • **All Data** mode can result in significant memory usage. You should avoid using this mode when processing files of several megabytes.

Program Generation - Feed Rates

Use the **Program Generation - Feed Rates** page to set the minimum and maximum allowable feed rates for the controller.

To open the **Feed Rates** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.
2. In the **Option File Properties** dialog, select **Program Generation - Feed Rates**.



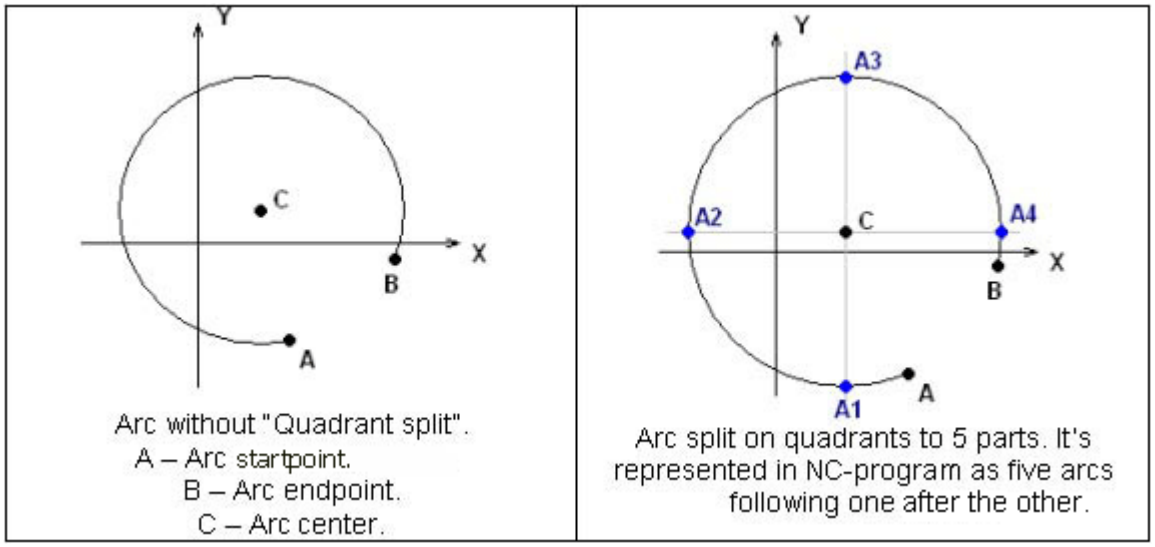
Parameter	Description
Maximum Inverse Time	Specifies the maximum feed rate value for cutting moves when using a **Feed Mode** of **INVERSE_TIME**.
Use Advanced Inverse Time	Select **Yes** to correct the feed rate when the tool rotates around the tool tip.
Restrictive Coefficient	Enter a value to control the maximum feed rate when **Use Advanced Inverse Time** is selected. The value represents the offset of a secondary point on the tool for calculating the feed rate when the tool axis changes, but the tip does not move.
Minimum Feed Rate	Specifies the minimum possible feed rate value. When processing CLDATA, all feed rates below this value are increased to this rate.
Maximum Rapid Feed Rate	Specifies the maximum feed rate value for rapid moves. When processing CLDATA, all feed rates above this value are decreased to this rate.
Maximum Cutting Feed Rate	Specifies the maximum feed rate value for cutting moves. When processing CLDATA, all cutting feed rates above this value are decreased this rate.

Program Generation - Arcs and splines

Use the **Program Generation - Arcs and Splines** page to specify the arc interpolation settings for the option file.

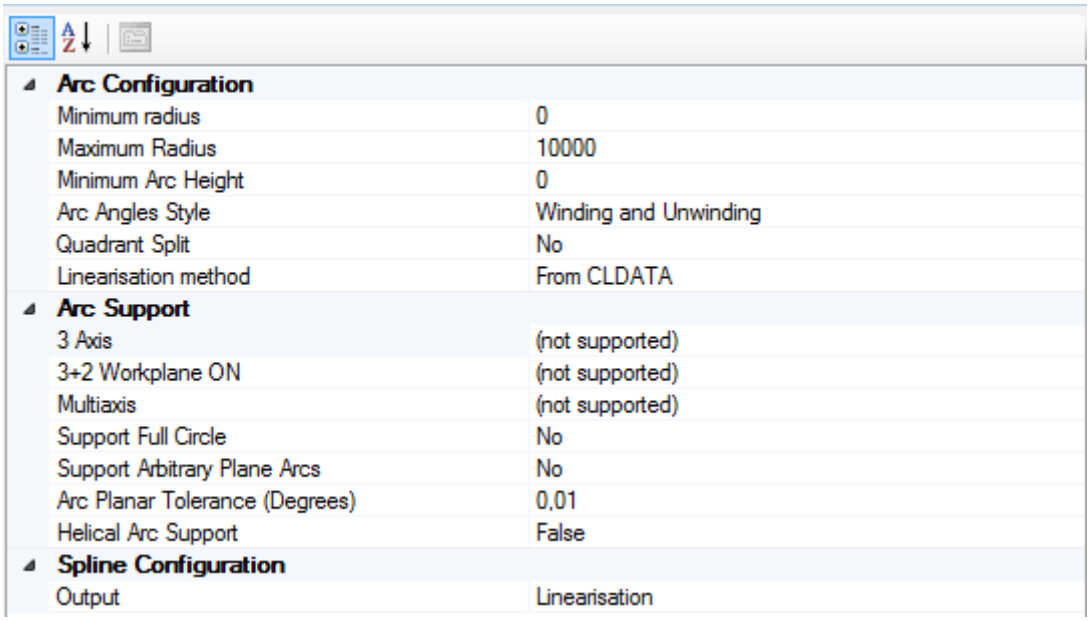
To open the **Arcs and Splines** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.

Parameter	Description
Minimum Radius	Specifies the minimum possible radius value for an arc in the NC program. An arc with a smaller radius is written out in a linearised form. This option is available only when you have selected one or more options in **Arcs support**.
Maximum Radius	Specifies the maximum possible radius value for an arc in the NC program. An arc with a greater radius is written out in a linearised form. This option is available only when you have selected one or more options in **Arcs support**.
Minimum Arc Height	Specifies the minimum **bulge factor** supported by the machine controller. An arc with height below this value are written out as a series of linear segments approximating the arc. This option is available only when you have selected one or more options in **Arcs support**.  The bulge factor is the maximum distance between the arc and its chord.
Arc Angles Style	Specifies angle style for **Arc Start Angle** and **Arc End Angle**. For more information see Angle Styles. Note, **Arc Start Angle** is used as previous value for **Arc End Angle** if *Relative* or *Winding and Unwinding* styles are used.
Quadrant Split	Specifies whether arcs are split into quadrants in the NC program. Some controllers require arcs to be split in this way, so select **Yes**. This option is available only when you have selected one or more options in **Arcs support**. 
Linearisation Method	Specifies how an arc is linearised: 💡 **From CLDATA** - linearisation is carried out by the CAM system and is contained in the processed CLDATA file. 💡 **Calculation** - linearisation is carried out by **Delcam PostProcessor** when the CLDATA file is postprocessed. Any linearisation carried out by the CAM system is ignored. This means that **Delcam PostProcessor** can define the arc linearisation with a tolerance different from the CLDATA tolerance. 💡 For the setting of the **Tolerance**, refer to **Linearisation**.

Arcs Support	Specifies the method used to create arcs in each base plane. For each 3-axis and 3+2-axis toolpath option, select the check box if you want to write out arc movement commands to the NC program file; deselect the check box if you want to write out arcs as a series of linear segments approximating the arc.
Helical Arc Support	Specifies how a helical-like arcs is output: • **Yes** - controller is able handle such arcs and will be used **Helical Move XY** (**XZ**, **YZ** depending on plane) command. • **No** - default value, will be used **Linearisation**.
Spline Configuration	Because **PowerMILL** cannot generate spline records, this option assumes you have used the **SplineMILL** utility to add splines to your CLDATA file. The postprocessed **Output** can take either of the following forms: • **Linearisation** - polygonises the spline. • **Polynomial** - generates the curve as a polynomial, using the output from **SplineMILL**. • **B-spline** - generates the Bezier curve, using the output from **SplineMILL**.

2. In the **Option File Properties** dialog, select **Program Generation - Arcs and Splines**.



Arc interpolation

When **PowerMILL** saves CLDATA file with arc interpolation, it also saves arc linearisation.

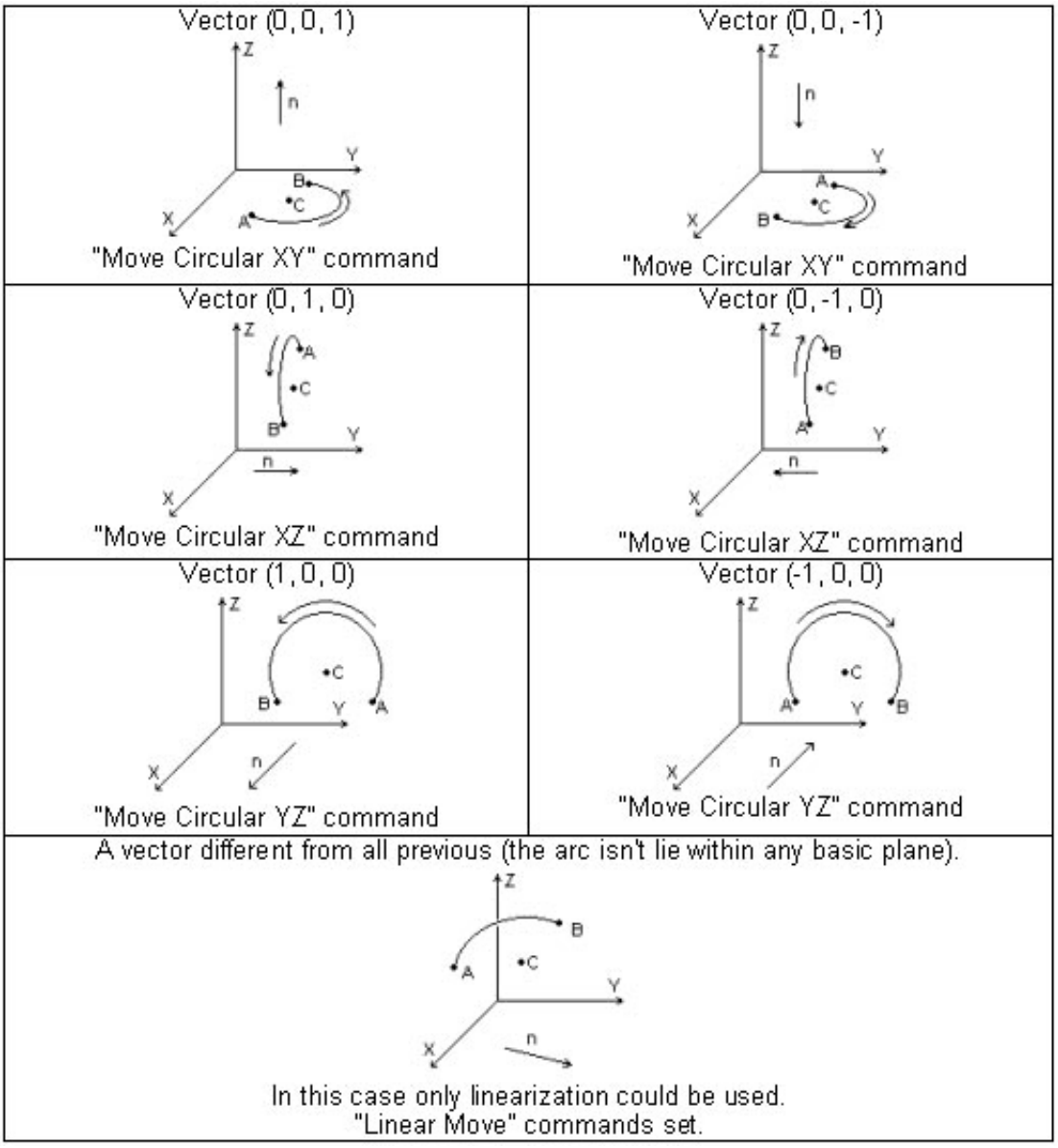
Since the CLDATA contains not only the arc attributes, but also the linearisation sequence, you can choose what to output in the NC program.

The vector of a normal to an arc's plane plays the main role in arc definition. Depending on this, the vector arc can take different positions in space and different commands are used for its output to an NC program.

An arc is defined by the following:

- **A** - Arc start point;
- **B** - Arc end point;
- **C** - Arc centre;
- **n** - Vector of the normal to the arc.

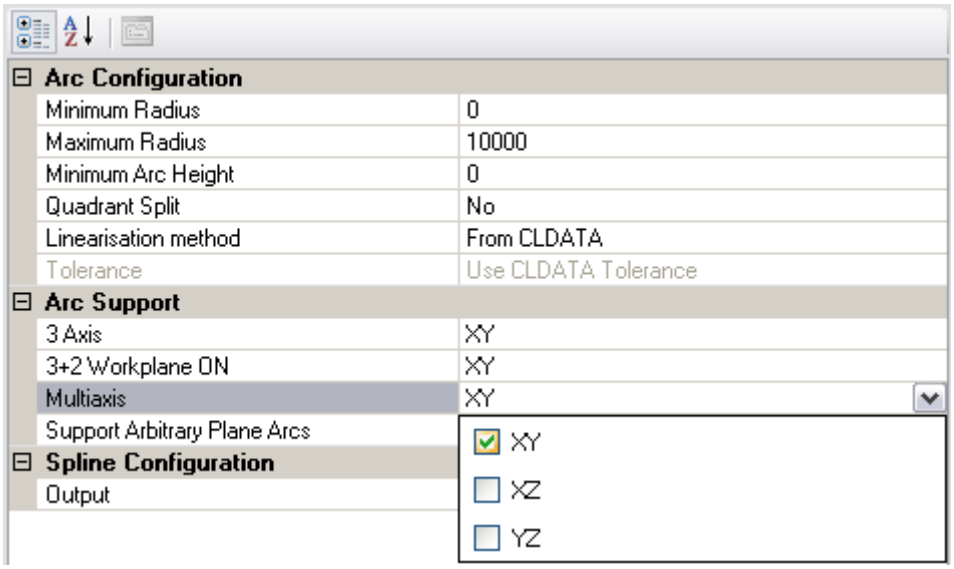
The vector **n** defines the location of the arc, whereas the other options define the arc itself. The following illustration shows the effects of changing the vector **n** of an arc:



Arc interpolation is supported

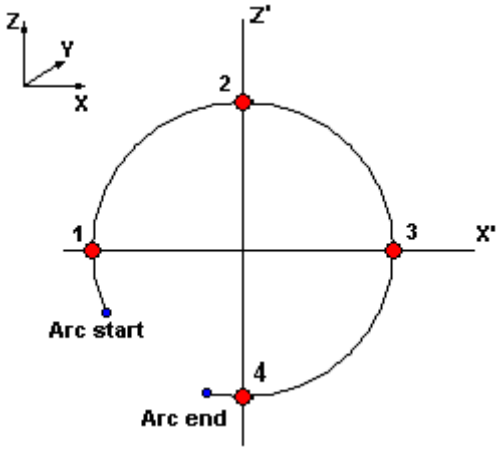
If arc interpolation is supported by a controller, you must set **Output Arc** to **Arcs** so that the **Postprocessor** outputs any circular moves using Circle Processing.

In the following example, all the **XY** arcs contained in the CLDATA files are written out by the **Postprocessor** as a set of arc moves:



In this case, the **Minimum Radius**, **Maximum Radius** and **Quadrant Split** fields are enabled.

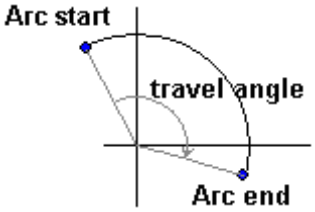
- **Minimum Radius** - enter the minimum radius for the arcs that are generated.
- **Maximum Radius** - enter the maximum radius for the arcs that are generated. If the **Postprocessor** receives a radius in excess of this value, the arcs are polygonised.
- **Quadrant Split** - if this is set to **Yes**, then, during arc output, an arc is divided into quadrants if it crosses the coordinate axis of the local two-dimensional coordinates:



The above illustration shows the arc in the XZ plane. Red points denote the points where the arc traverses the quadrant boundaries. The location of the axes of the local coordinates match the corresponding axes of the global coordinates and so the arc is output in the NC program as five successive circular moves (**Arc start-1**, **1-2**, **2-3**, **3-4**, and **4-Arc end)**.

Arc travel angle

Some machines use a travel angle, which is a relative angle characterizing the angle moves on an arc:



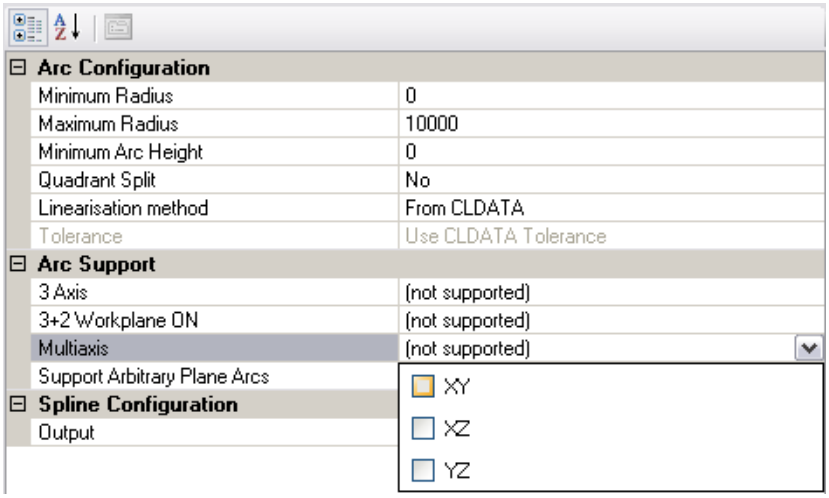
If a trajectory is clockwise, the travel angle is negative. If the trajectory is counter-clockwise, the travel angle is positive.

The **Arc Travel Angle** is available as both a parameter and format in **Delcam PostProcessor**. This is a real value which, when used as a parameter in a command block, must be set to the difference between the absolute start angle (typically, zero) and the end arc angle (this should be no more than 360 degrees).

Linearisation

If arc interpolation is not supported by the controller, you must set **Output Arc** to **Linearisation** so the **Postprocessor** outputs linear approximations for the arcs, which are defined by CLDATA or generated by the **Postprocessor** (**Delcam PostProcessor** uses a linear approximation when the arc radius is greater than the maximum radius for the machine).

In the following example, all arcs contained in the CLDATA files are written out by the **Postprocessor** as a set of standard linear moves:

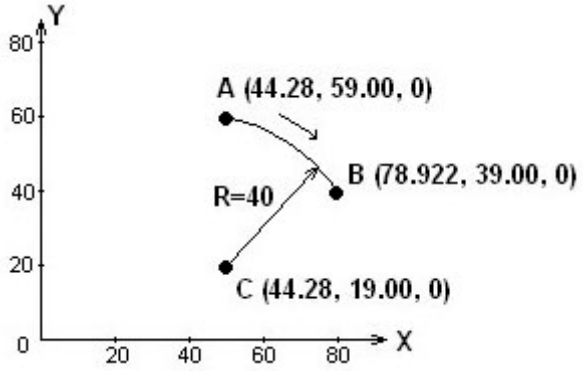


If you have an **Arc Output** of **Linearisation**, you must specify the **Linearisation Method**. A setting of **From CLDATA** uses the CLDATA file settings, whilst the alternative setting of **Calculation** allows you to specify your own **Tolerance** setting in the **Tolerance** field. This field defines the maximum deviation between the original arc and a series of straight lines used to represent the curve, and you can set it to either **Use CLDATA Tolerance** (in which case the deviation is taken from the CLDATA file) or **Specify Tolerance**, in which case you can specify your own deviation in the adjacent field.

> Linearisation is represented in an NC program as a sequence of **Move Linear** commands.

Comparing arc and linearised output

There are three variants of NC code by which an arc can be represented in an NC program (based on the option file for a Fanuc controller):

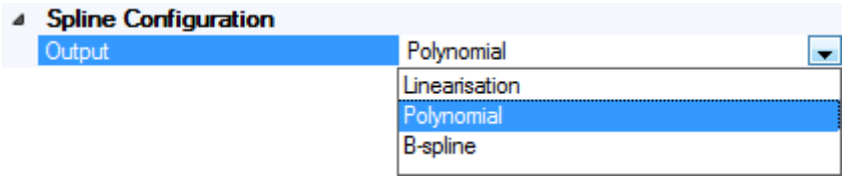


Linearization is not used	CLDATA linearization	PM-Post linearization with tolerance of 0.09
N410G2G17X78.922Y39.I0J-40. Here: G17 – indicates XY plane, G2 – clockwise direction, X, Y – circle end point, I, J – coordinates of the arc center (relative to the start point).	N400X44.281 N410X49.346Y58.678 N420X54.457Y57.684 N430X59.482Y55.999 N440X64.281Y53.641 N450X68.723Y50.664 N460X72.694Y47.155 N470X76.111Y43.225 N480X78.922Y39.	N400X44.281 N410X46.897Y58.914 N420X49.502Y58.658 N430X52.085Y58.231 N440X54.634Y57.637 N450X57.139Y56.877 N460X59.588Y55.955 N470X61.973Y54.875 N480X64.281Y53.641 N490X66.504Y52.259 N500X68.631Y50.734 N510X70.655Y49.074 N520X72.565Y47.284 N530X74.355Y45.374 N540X76.015Y43.35 N550X77.54Y41.223 N560X78.922Y39.

Spline configuration

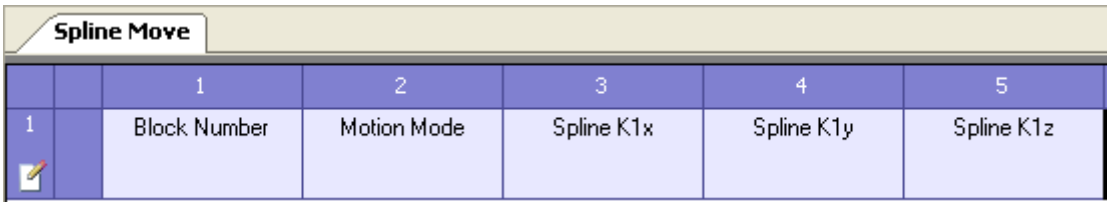
To enable splines in your option file:

1. Ensure your CLDATA files were generated by SplineMILL. (CLDATA files from **PowerMill** do not contain spline data.)
   - Only binary CLDATA can be accepted by **SplineMILL**.
2. Go to the **Option File Properties** dialog in the **Editor** and set **Spline Configuration - Output** to **Polynomial** or **B-spline**:



3. Open the Spline folder in the Commands tree, and activate the **Spline Move** command.

4. Complete the **Spline Move** block with the appropriate spline parameters from the controller's manual.



5. Spline interpolation automatically creates a curve that smoothly traces specified points. This enables high-speed and high-accuracy machining for free shapes along a smoothly curved toolpath.

Delcam PostProcessor supports set of splines, for example **Cubic Polynomial Splines** can be represented as:

C(t) = K0 + K1 * t + K2 * t ^ 2 + K3 * t ^ 3, t = (0;1)

For most controllers, K0 may be omitted.

Delcam PostProcessor has the special parameters "Spline K0x", "Spline K0y", "Spline K0z", ... "Spline K3x", "Spline K3y", "Spline K3z". These parameters are spline coefficients.

For example, Heidenhain MillPlus accepts splines in the following form:

G6 X51=.. Y51=.. Z51=.. X52=.. Y52=.. Z52=.. X53=.. Y53=.. Z53=..

MillPlus	**Delcam PostProcessor**
X51	Spline K1x
Y51	Spline K1y
Z51	Spline K1z
X52	Spline K2x
Y52	Spline K2y
Z52	Spline K2z
X53	Spline K3x
Y53	Spline K3y
Z53	Spline K3z

Program Generation - Drilling

The **Drilling** page enables you to define the parameters and settings that control the drilling cycle.

To open the **Drilling** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.
2. In the **Option File Properties** dialog, select **Program Generation - Drilling**.

Parameter	Description
Drilling Hole Top	Specifies the distance between the Z position of the tool and workpiece surface.
Drilling Feed Rate	Specifies the traversing speed of the tool during drilling.
Drilling Clear Plane	Specifies the coordinate in the tool axis at which no collision can occur between the tool and workpiece (including the clamping devices).
Drilling Peck Depth	Specifies the depth of the pecking move.
Drilling Number of Depths	Specifies the total number of pecking moves.
Drilling Total Depth	Specifies the total depth of the hole.
Drilling Dwell	Specifies the time in seconds that the tool remains at the hole bottom.
Drilling First Depth	Specifies the depth of the first drilling move.
Drilling Hole Diameter	Specifies the diameter of the hole.
Drilling Retract 2nd Height	Specifies the height of retract position.

To invert a parameter's value, select the corresponding check box in the **Negative** column.

To specify the drilling cycle origins for the controller, choose an entry in the **Origin Positions** list for each parameter. Select:

- **Last Z** to output depths measured from the current Z value reached by the toolpath.
- **Clear Plane** to output depths relative to the **Drilling Clear Plane** setting.
- **Hole Top** to output depths relative to the hole top setting in the CLDATA file.
- **Hole Bottom** to output depths relative to the hole bottom setting in the CLDATA file.
- **Z Zero** to output depths relative to the origin of theNC Program

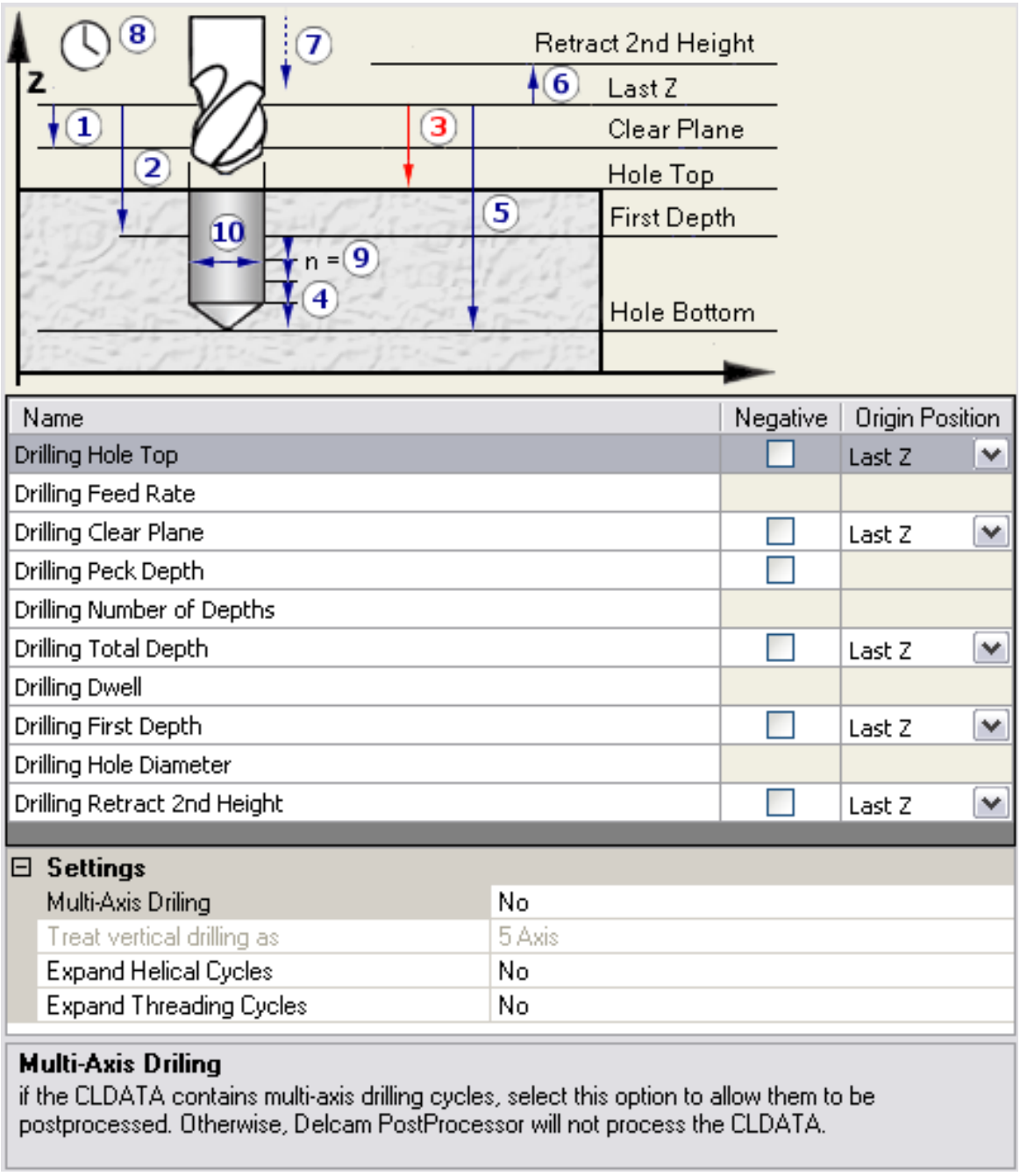
Settings

Multiaxis drilling - if the CLDATA contains multi-axis drilling cycles, select this option to allow them to be postprocessed. Otherwise, Delcam PostProcessor will not process the CLDATA.

Treat vertical drilling as - Delcam PostProcessor uses Automatic Coordinate Control to handle multi-axis drilling cycles. It generates the commands that transfer a multi-axis cycle into the 3+2 part of the trajectory as Delcam PostProcessor defines and uses a local workplane for each multi-axis cycle that it processes. However, in some cases, a multi-axis cycle can be processed without being transformed to a local workplane, such as when the direction of the cycle is parallel to the machine's Z axis (for example, for machines with rotary axes on the table). If you select this option, Delcam PostProcessor avoids using ACC wherever possible, giving simpler results. If the option is not selected, Delcam PostProcessor always uses ACC for drilling cycles.

This option is unavailable if **Multiaxis drilling** is not selected, or if Automatic Coordinate Control is not enabled.

In this example, all the depths are measured from the last Z position of the tool. **Drilling First Depth** is equal to **Drilling Total Depth** for single pecking drilling cycles.



Some parameters may be measured from different levels to meet controller requirements. Delcam PostProcessor allows any of the parameters to be recalculated with scripts. For example, on Heidenhain TNC530s, the clear plane value is the distance between the start position of the tool tip and the workpiece surface. For Delcam PostProcessor it can be recalculated in the following way:

Drilling Clear Plane = Drilling Hole Top - Drilling Clear Plane

The corresponding JScript code for use in Scripting is as follows:

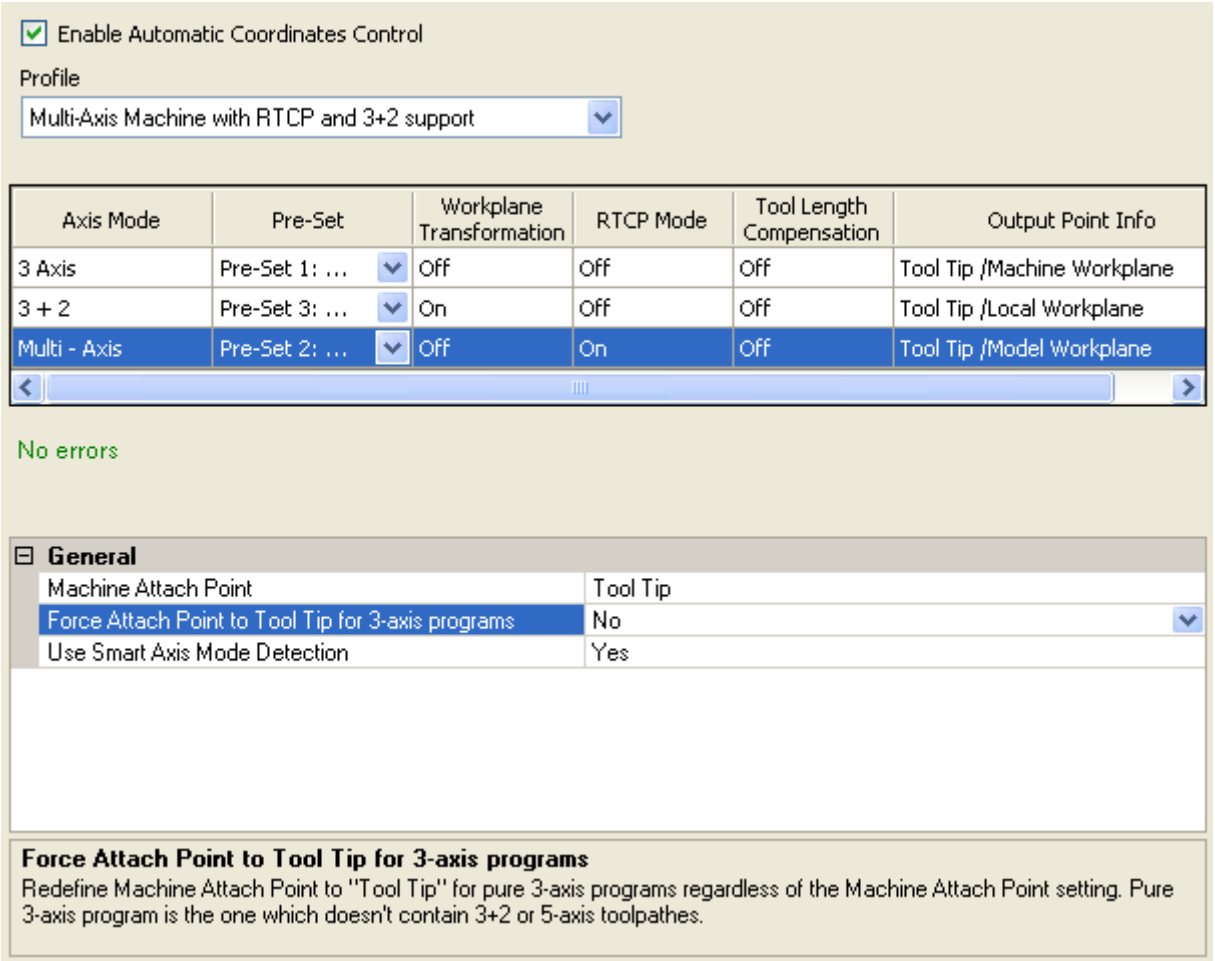
```
SetParam("%p(Drilling Clear Plane)%", GetParam("%p(Drilling Hole Top)%") -
GetParam("%p(Drilling Clear Plane)%"));
```

Expand Helical Cycles - if the controller does not support standard helical cycles, select this option to replace the cycles with a series of helices in the NC program file.

Expand Threading Cycles - if the controller does not support standard threading cycles, select this option to replace the cycles with a series of helices in the NC program file.

Program Generation - Coordinates control

Use the **Coordinates Control** view to configure the ways in which the **X**, **Y** and **Z** coordinates are calculated.



Enable Automatic Coordinate Control (ACC) Select the check box to automatically control the output of coordinates according to axis mode.

Profiles Select an entry in this list to specify the axis modes supported by the target controller. These profiles specify the presets that control the coordinates output using commands and parameters.

- You are advised to set **Automatic Coordinate Control** to **On** to gain maximum multi-axis and workplane control, although you should pay special attention to this when Drilling.

Machine Attach Point This determines the **X**, **Y** and **Z** coordinates of the machine's drive point. The available options are:

- **Tool Tip** (used in the majority of cases),
- **Gauge Face**
- **Pivot**
- Many calculations inside **Delcam PostProcessor** depend on the **Machine Attach Point** setting.

Force Attach Point to Tool Tip for 3-axis Select **Yes** to force a **Machine Attach Point** of **Tool Tip** for 3-axis cases regardless of the **Machine Attach Point** setting.

Use Smart Axis Mode Detection Sets traditional or new method to define **Toolpath Axis Mode**:

- **Yes** (default). Toolpath Axis Mode is set dependently on behavior of real moves contained in a toolpath.
- **No**. Old-style Toolpath Axis Mode definition based on special CLDATA records.

Commands affected by ACC

- You are strongly advised to set **Automatic Coordinate Control** to **On** to maximize multi-axis and workplane control. However, you should pay special attention when Drilling.

The associated commands are given in the following table:

> Command	Description
Set Workplane On **Set Workplane Off**	These commands are used to turn the local workplane **On** and **Off**. They are used only when **Automatic Coordinate Control** is **On** and the controller includes 3+2 support. The **Set Workplane On** command is invoked for each local workplane in the toolpath. Following the processing of a toolpath in the local workplane using **the Set Workplane On** parameters, the **Set Workplane Off** command is invoked prior to the start of the next toolpath, unless both toolpaths consist of 3+2 moves using the same tool and the same local workplane, and the next toolpath starts where the previous one ended (so that there are no connection moves). These commands can be used to control the **Workplane Transformation** parameter without scripting.
Set Multi-Axis On **Set Multi-Axis Off**	**Set Multi-Axis On** is selected by a 5-axis toolpath in CLDATA, and by a 3+2 toolpath when **Set Workplane** is **Off**. You must set **RTCP (Rotate Tool Centre Point)** to **On** in the **Set Multi-Axis On** command (where appropriate). **Set Multi-Axis Off** is automatically selected by a 3+2 toolpath when **Set Workplane** is **On**. You must set **RTCP (Rotate Tool Centre Point)** to **Off** in the **Set Multi-Axis Off** command (where appropriate) - see below for an example.
Workplane Setting	This command must be activated only if **ACC** is **Off**. It is the basic command for working with local workplanes that are derived from CLDATA. If **ACC** is **On**, and 3+2 is supported, you are advised to use the **Set Workplane On** and **Off** commands instead to control local coordinates.

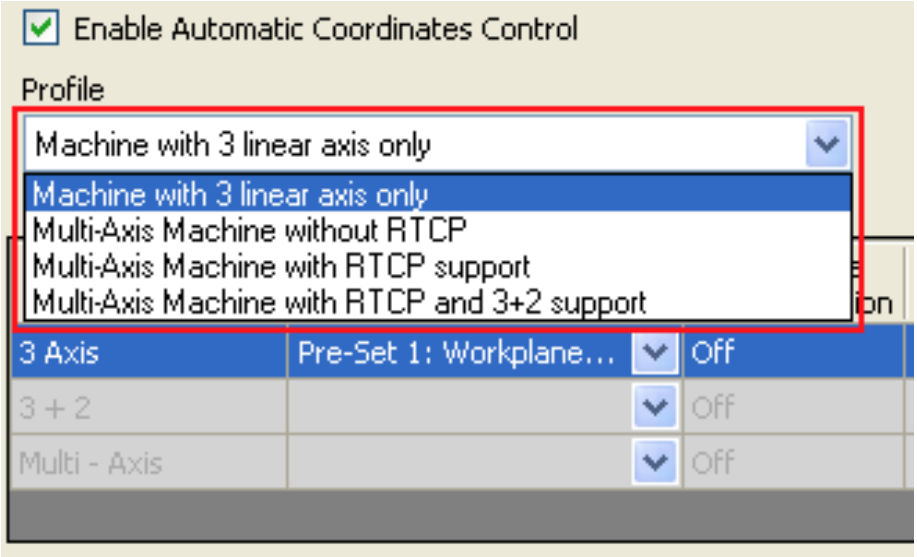
Parameters affected by ACC

Parameter	Possible States	Used in Commands	Description
Toolpath Axis Mode	**3-Axis** **3+2** **Multi-Axis**		This parameter contains the current axis mode. The states are never mixed in one program - 3-Axis trajectories can be present only in pure 3-Axis programs; otherwise they are always considered as 3+2 (or if **3+2** is not supported, as **Multi-Axis**). This parameter is controlled from CLDATA and cannot be changed. It can help when working with scripting.
Workplane Transformation	**On / Off**	**Set Multi-Axis On** **Set Multi-Axis Off** **Workplane Setting** when **ACC** is **Off**. **Set Workplane On/Off** when **ACC** is **On**.	Controls local workplane activation. If the setting is **On**, **Delcam PostProcessor** tries to transfer all the coordinates into the local workplane specified in CLDATA (alternatively, you can change the workplane using scripts). This control is not possible for continuous multi-axis motion sequences used for 3+2 trajectories. After transfer into the local workplane, the path assumes the properties of a 3-Axis trajectory, including support of arc interpolation, drilling cycles, and so on. The output point in the local workplane is **Tool Tip** or **Gauge Face**, depending on the current state of the **Tool Length Compensation Mode** parameter (see below). If **Workplane Transformation** is **Off**, there is no corresponding transformation of coordinates. The transformation of coordinates into the local workplane cannot be carried out when **RTCP Mode** is **On**.
RTCP (Rotation Tool Compensation Point) Mode	**On / Off**	**Set Multi-Axis On / Off**	This parameter can be used for continuous **Multi-Axis** move sequences and **3+2** sequences where the local workplane is not used. When a machine makes rotations about its rotary axes, a tool changes its direction and can also change its position dependent on the workpiece (being dictated by the linear coordinates). These changes must be

Parameter	Possible States	Used in Commands	Description
			compensated for to keep the tool within the workpiece's coordinate system. If **RTCP Mode** is **On**, **PM- Post** assumes a controller can compensate for these rotations itself, and the output point in this case is always **Tool Tip** (by default, it is assumed that a controller can compensate for tool length as well as rotations. Otherwise, if the state is **Off**, **Delcam PostProcessor** calculates the compensations and outputs a **Pivot** point to the program.
Tool Length Compensation Mode	**On / Off**	**First Move After Toolchange** **Load Tool First** **Load Tool**	Most controllers can lift the head along the machine's **Z** axis by the length of the tool after the tool is loaded. There is no need to apply this parameter to controllers that do this by default. The state of this parameter becomes important when the **Tool Tip** or **Gauge Face** option is defined as the output point. If **Tool Length Compensation Mode** is **On**, **Delcam PostProcessor** assumes the controller can compensate for the tool length itself and outputs a **Tool Tip** point. Otherwise (if the state is **Off**), **Delcam PostProcessor** compensates for the tool length by outputting a **Gauge Face** point. However, this depends on other parameters. For example, **Tool Length Compensation Mode** is ignored when **RTCP Mode** is **On** (as **RTCP** includes **Tool Length Compensation Mode**); also, **Tool Length Compensation Mode** cannot be applied when the current tool direction is different from the Z axis direction (but it works in **3+2** trajectories when the local workplane is used).
Output Point Info	**Tool Tip** **Gauge Face** **Pivot**		This parameter shows the point for each workplane currently defined by the **X**, **Y** and **Z** coordinates. It cannot be changed.

Profiles and presets

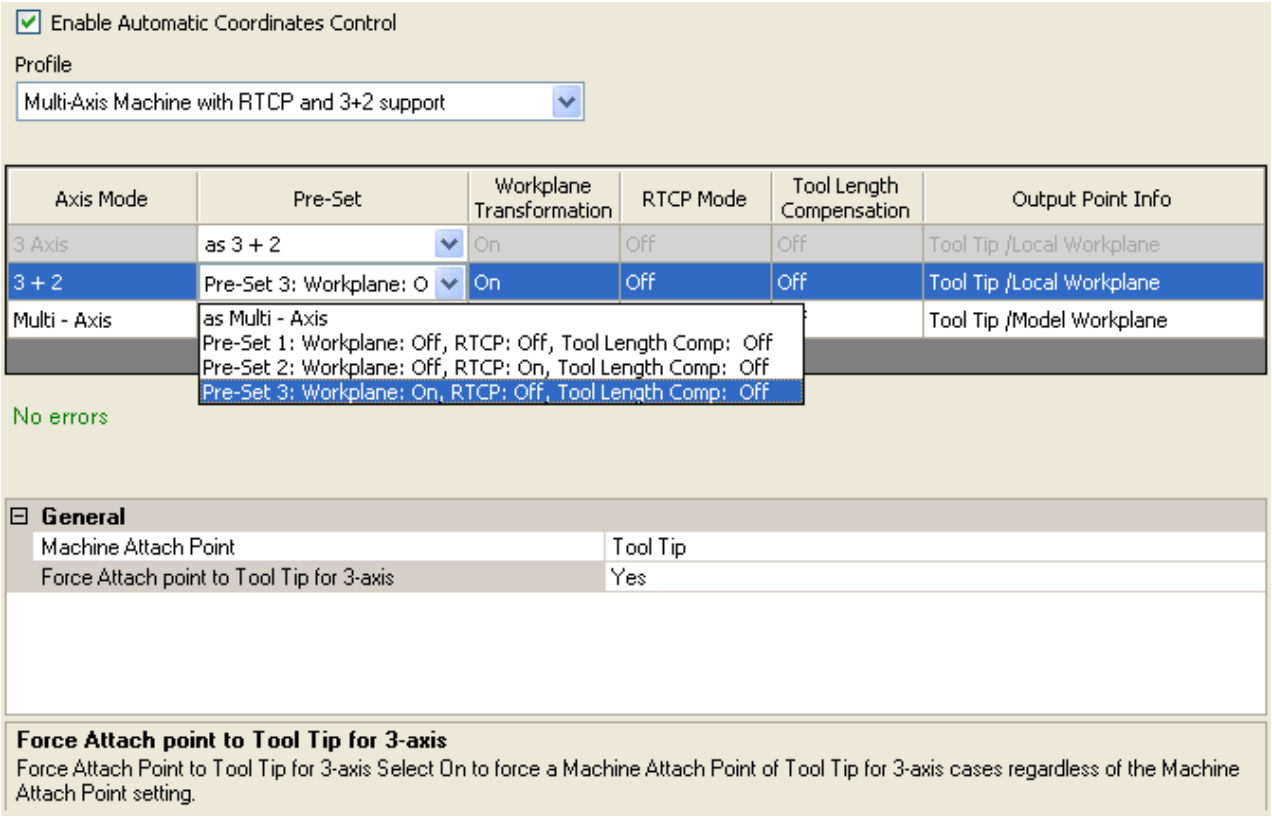
It is important that you select a profile that is appropriate for the machine controller on which the associated tapefiles will be used:



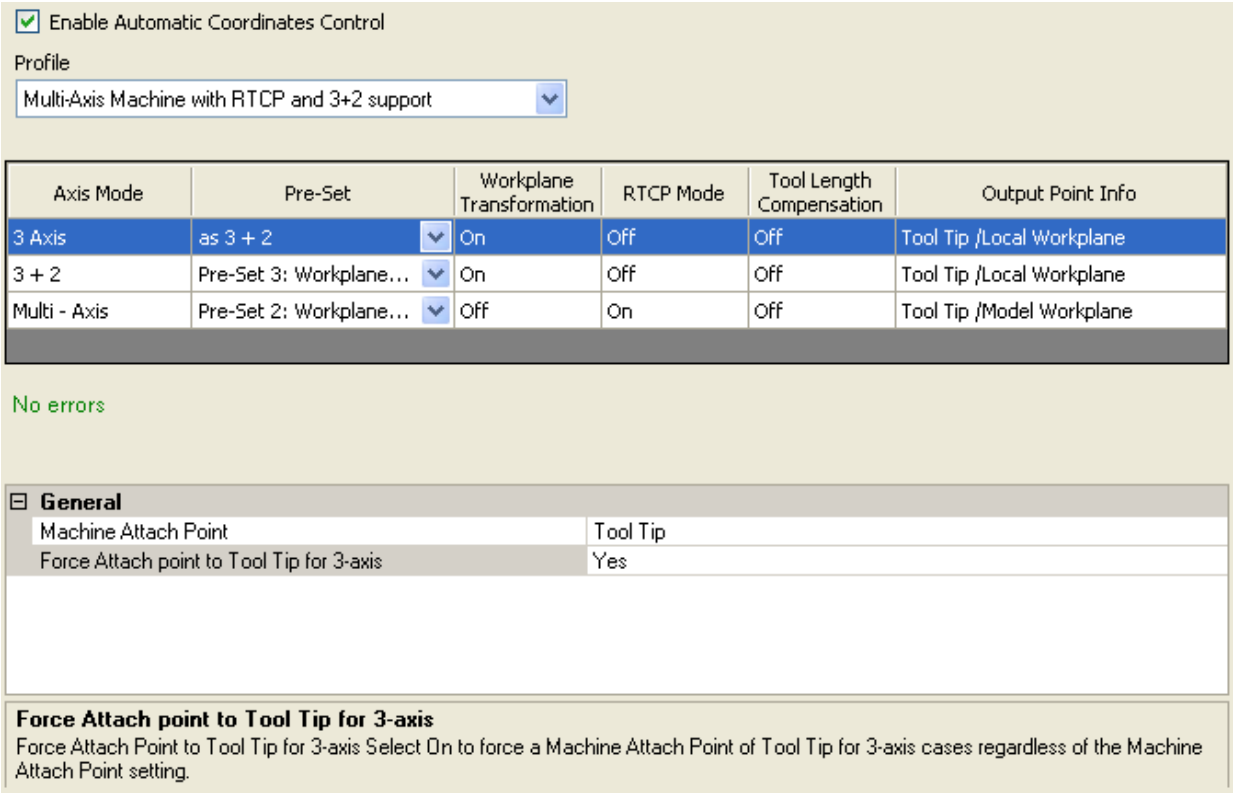
When the profile is selected, you may need to adjust the preset for each axis mode.

> A **preset** is a set of states which **Delcam PostProcessor** uses to control coordinate parameters when a program is in the corresponding axis mode (**3-Axis**, **3+2** or **Multi-Axis**). The number of available presets depends on the selected profile.

To select a preset, click the down arrow next to the **Pre-Set** field for the required **Toolpath Axis Mode**, and select the preset in the drop-down list:



Following selection of the presets, the table in the **View** pane shows the parameter states that **Delcam PostProcessor** uses for each axis mode during postprocessing. For example:



For each preset, the combination of the states of the **Workplane Transformation**, **RTCP Mode** and **Tool Length Compensation Mode** parameters defines the **Output Point Info**, which shows what the **X**, **Y** and **Z** coordinates mean. The information consists of two parts: a point and the coordinate system in which the point is represented. The possible points are: **Tool Tip**, **Gauge Face**, **Pivot**, or **Undefined** (which means that the **XYZ** values do not correspond to any of the points on the machine). The possible coordinate systems are: **Model**, **Machine** or **Local**.

> For **Machine Attach Point** and **Force Attach Point to Tool Tip for 3-axis**, refer to Coordinate Control.

Machine with three linear axes only

This is the most restricted configuration as it only allows you to control **Tool Length Compensation Mode** (the other parameters, **RTCP Mode** and **Workplane Transformation**, are not used by **Delcam PostProcessor** with this machine configuration).

The only available **Toolpath Axis Mode** available for this profile is **3-Axis**:

Available Presets	Default Output Point Mode	Notes
Preset 1: Workplane: Off, RTCP: Off, Tool Length Comp: Off	**Gauge Face / Machine Workplane**	This preset can be used if **Preset 2** with its **Tool Length Compensation Mode** cannot be applied and you need to have **Gauge Face** coordinates in a 3-axis program.
Preset 2: Workplane: Off, RTCP: Off, Tool Length Comp: On	**Tool Tip / Machine Workplane**	This preset represents the default configuration for 3-axis programs. For some controllers it is not necessary to enable **Tool Length Compensation Mode** in a program (usually using the **G43** code) because compensation is automatic. For more information, see Parameters defining output coordinates.

Multi-axis machine without RTCP support

This configuration is intended for older multi-axis machines which have rotary axes that cannot compensate for rotations around them (that is, machines that do not support the **Rotate Tool Centre Point** setting).

The available presets for this profile for the different **Toolpath Axis Modes** are:

Axis Mode: 3-Axis

Available Presets	Default Output Point Mode	Notes
Preset 1: Workplane: Off, RTCP: Off, Tool Length Comp: Off	**Gauge Face / Machine Workplane**	This preset can be used if **Preset 2** with its **Tool Length Compensation Mode** cannot be applied and you need to have **Gauge Face** coordinates in a 3-axis program.
Preset 2: Workplane: Off, RTCP: Off, Tool Length Comp: On	**Tool Tip / Machine Workplane**	This preset represents the default configuration for 3-axis programs. For some controllers it is not necessary to enable **Tool Length Compensation Mode** in a program (usually using the **G43** code) because compensation is automatic - refer to Parameters defining output coordinates.
as 3+2	As for **Axis Mode: 3+2**	This preset must be selected if 3-axis trajectories are be represented as 3+2. It then makes no difference whether 3-axis or a similar 3+2 trajectory (3-axis program saved in **Multi-axis** mode) is processed. 3-axis fields in the table are disabled if this preset is selected. All states of the 3-axis attributes are set to be the same as the corresponding 3+2 states

Axis Mode: 3+2

Available Presets	Default Output Point Mode	Notes
Preset 1: Workplane: Off, RTCP: Off, Tool Length Comp: Off.	**Gauge Face / Machine Workplane**	This preset can be used for all kinematics because there is no restriction on **Tool Length Compensation Mode**. This preset is suitable for older machines where "everything is off".
Preset 2: Workplane: Off, RTCP: Off, Tool Length Comp: On.	**Tool Tip / Machine Workplane**	This preset (with **Tool Length Compensation Mode** enabled) can be applied only in the case of kinematics without head rotary axes. Because Workplane Transformation is **Off**, Delcam PostProcessor assumes the 3+2 trajectory is continuous multi-axis and all multi-axis rules are therefore applied. Because **RTCP** is also **Off**, a controller does not compensate machine rotations. In this condition, if there is at least one rotary axis on the head, the tool direction vector is variable in space and **Tool Length Compensation Mode** along the machine's Z axis cannot be applied. This preset is still available for selection for such kinematics, but **Delcam PostProcessor** stops processing with an error message if there is an attempt to process a 3+2 trajectory. In conclusion, this preset is used for kinematics such as a 4- axis or 5-axis table, where a tool has constant direction in space and **Tool Length Compensation Mode** can be applied without problems.
as Multi-Axis	As for **Axis Mode: Multi-Axis**	Selecting this preset removes the difference betwen 3+2 and pure multi-axis trajectories. **Delcam PostProcessor** starts to process all non-3-axis moves as multi-axis. All 3+2 fields in the table are therefore disabled. All states of the 3+2 attributes are set to the same as the corresponding multi-axis states. If this preset is used with the **as 3+2** preset for **3-Axis** mode, **Delcam PostProcessor** assumes all trajectories to be multi-axis.

Axis Mode: Multi-Axis

Available Presets	Default Output Point Mode	Notes
Preset 1: Workplane: Off, RTCP: Off, Tool Length Comp: Off	**Gauge Face / Machine Workplane**	This preset can be used for all kinematics because there is no restriction on **Tool Length Compensation Mode**. This preset is suitable for older machines with poor capabilities where "everything is off".
Preset 2: Workplane: Off, RTCP: Off, Tool Length Comp: On	**Tool Tip / Machine Workplane**	**This preset works only for kinematics without head rotary axes**. For continuous multi-axis trajectories, **Workplane Transformation** cannot be used in **Delcam PostProcessor.** As **RTCP** is also **Off**, **Tool Length Compensation Mode** becomes important, but it only applies to machines with rotary axes on the table. **Delcam PostProcessor** stops processing with an error message if there is an attempt to process a multi-axis trajectory using kinematics with head rotary axes.

Multi-axis machine with RTCP support

This profile supports the **Rotate Tool Centre Point (RTCP)** setting where appropriate.

The available presets for **3-Axis Mode** are as for Multi-Axis Machine without RTCP Support.

The presets for the other modes are:

Axis Mode: 3+2

Available Presets	Default Output Point Mode	Notes
Preset 1: Workplane: Off, RTCP: Off, Tool Length Comp: Off	**Gauge Face / Machine Workplane**	This preset can be used for all kinematics because there is no restriction on **Tool Length Compensation Mode**. This preset is suitable for older machines with poor capabilities where "everything is off".
Preset 2: Workplane: Off, RTCP: Off, Tool Length Comp: On	**Tool Tip / Machine Workplane**	This preset (with **Tool Length Compensation** enabled) can be applied only in the case of kinematics without head rotary axes. Because Workplane Transformation is **Off**, the 3+2 trajectory is considered by **Delcam PostProcessor** as continuous multi-axis and all multi-axis rules are therefore applied here. Because **RTCP** is also **Off**, a controller does not compensate machine rotations. In this condition, if there is at least one rotary axis on the head, the tool direction vector is variable in space and **Tool Length Compensation Mode** along the machine's Z axis cannot be applied. This preset is still available for selection for such kinematics, but **Delcam PostProcessor** stops processing with an error message if there is an attempt to process a 3+2 trajectory. In conclusion, this preset is used for kinematics such as a 4-axis or 5-axis table, where a tool has constant direction in space and **Tool Length Compensation Mode** can be applied without problems.
Preset 3: Workplane: Off, RTCP: On, Tool Length Comp: On	**Tool Tip / Model Workplane**	This preset implies that 3+2 moves are considered as continuous multi-axis moves. **RTCP** is used here, and so the **Tool Length Compensation Mode** state is not important because the **Output Point** is always the **Tool Tip**. Make sure there is no superfluous output of **Tool Length Compensation Mode** codes in a program (for 3+2 trajectories). You must also set the **RTCP On/Off** codes in the commands **Set Multi-Axis On** and **Set Multi-Axis Off.**
as Multi-Axis	As for **Axis Mode: Multi-Axis**	Selecting this preset removes the difference between 3+2 and pure multi-axis trajectories. **Delcam PostProcessor** starts to process all non-3-axis moves as multi-axis. All 3+2 fields in the table are therefore disabled. All the states of the 3+2 attributes are set to be the same as the corresponding multi-axis states. If this preset is used together with the **as 3+2** preset for **3-Axis Mode**, then all trajectories without exception are considered by **Delcam PostProcessor** as multi-axis.

Axis Mode: Multi-Axis

Available Presets	Default Output Point Mode	Notes
Preset 1: Workplane: Off, RTCP: Off, Tool Length Comp: Off.	**Gauge Face / Machine Workplane**	This preset can be used for all kinematics because there is no restriction on **Tool Length Compensation Mode**. This preset is suitable for older machines with poor capabilities where "everything is off".
Preset 2: Workplane: Off, RTCP: Off, Tool Length Comp: On.	**Tool Tip / Machine Workplane**	**This preset works only for kinematics without head rotary axes**. For continuous multi-axis trajectories, **Workplane Transformation** cannot be used in **Delcam PostProcessor.** As **RTCP** is also **Off**, **Tool Length Compensation Mode** becomes important but it can only be applied to machines with rotary axes on the table. **Delcam PostProcessor** stops processing with an error message if there is an attempt to process a multi-axis trajectory using kinematics with head rotary axes.
Preset 3: Workplane: Off, RTCP: On, Tool Length Comp: Off.	**Tool Tip / Model Workplane**	This is the default preset when **RTCP** is available. The **Tool Length Compensation Mode** state is not important because the **Output Point** is always the **Tool Tip**. **Workplane Transformation** is not available for continuous multi-axis trajectories in **Delcam PostProcessor**. You must set the **RTCP On / Off** codes in the commands **Set Multi-Axis On** and **Set Multi-Axis Off**.

Multi-axis machine with RTCP and 3+2 support

This profile supports the **Rotate Tool Centre Point (RTCP)** setting where appropriate, and also allows you to use the local workplane to represent coordinates. This profile gives you maximum flexibility with regard to coordinate control. For example, you can switch the local **Workplane** to **On** for 3+2 toolpaths, but avoid the use of **RTCP** for multi-axis moves.

The available presets for **3-Axis Mode** are as for Multi-Axis Machine without RTCP Support.

The available presets for **Multi-Axis Mode** are as for Multi-Axis Machine with RTCP Support.

The presets for **Axis Mode: 3+2** are as follows:

Available Presets	Default Output Point Mode	Notes
Preset 1: Workplane: Off, RTCP: Off, Tool Length Comp: Off	**Gauge Face / Machine Workplane**	This preset can be used for all kinematics because there is no restriction on **Tool Length Compensation Mode**. This preset is suitable for older machines with poor capabilities where "everything is off".
Preset 2: Workplane: Off, RTCP: Off, Tool Length Comp: On	**Tool Tip / Machine Workplane**	This preset (with **Tool Length Compensation Mode** enabled) can be applied only in the case of kinematics without head rotary axes. Because **Workplane Transformation** is **Off**, Delcam PostProcessor assumes the 3+2 trajectory to be continuous multi-axis, and all multi-axis rules are applied. Because **RTCP** is also **Off**, a controller does not compensate machine rotations. In this condition, if there is at least one rotary axis on the head, the tool direction vector is variable in space, and **Tool Length Compensation Mode** along the machine's Z axis cannot be applied. This preset is still available for selection for such kinematics, but **Delcam PostProcessor** stops processing with an error message if there is an attempt to process a 3+2 trajectory. In conclusion, this preset is used for kinematics such as a 4- axis or 5-axis table, where a tool has constant direction in space and **Tool Length Compensation Mode** can be applied without problems.
Preset 3: Workplane: Off, RTCP: On, Tool Length Comp: On	**Tool Tip / Model Workplane**	This preset implies that 3+2 moves are considered as continuous multi-axis moves. **RTCP** is used here, and so the **Tool Length Compensation Mode** state is not important because the **Output Point** is always the **Tool Tip**. Make sure there is no superfluous output of **Tool Length Compensation Mode** codes in a program (for 3+2 trajectories). You must also set the **RTCP On/Off** codes in the commands **Set Multi-Axis On** and **Set Multi-Axis Off**.
Preset 4: Workplane:	**Gauge Face /**	This preset allows you to use the local workplane for the output of 3+2 trajectories. **Delcam**

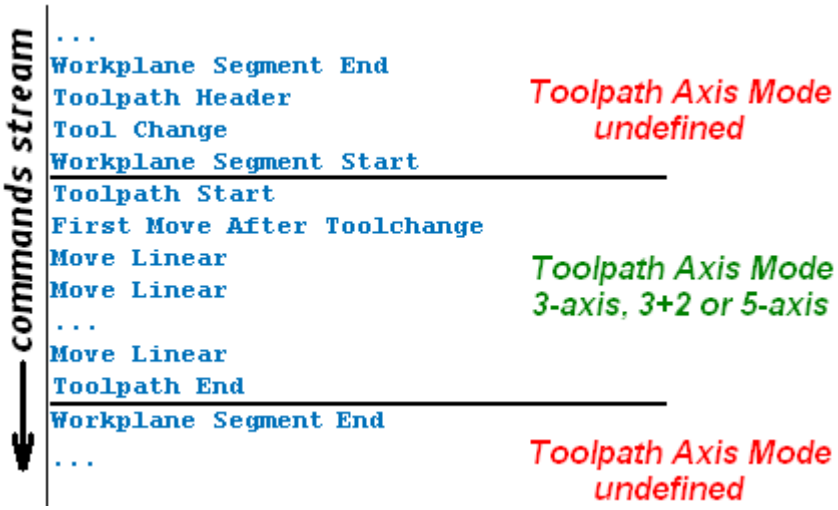
Available Presets	Default Output Point Mode	Notes
On, RTCP: Off, Tool Length Comp: Off	**Local Workplane**	**PostProcessor** automatically switches the local workplane **On** and **Off** for such trajectories, but you must configure the commands **Set Workplane On** and **Set Workplane Off**.
as Multi-Axis	As for **Axis Mode: Multi-Axis**	Selecting this preset removes the difference between 3+2 and pure multi-axis trajectories. **Delcam PostProcessor** starts to process all non-3-axis moves as multi-axis. All 3+2 fields in the table are therefore disabled. All the states of the 3+2 attributes are set to be the same as the corresponding multi-axis states. If this preset is used together with the **as 3+2** preset for **3-Axis Mode**, **Delcam PostProcessor** considers all trajectories as multi-axis.

Toolpath Axis Mode Definition

Each program consists of toolpaths and each toolpath has its **Toolpath Axis Mode**. It could be **3-axis**, **3+2** or **5-axis**. Machine controllers do not require axis mode to process, so, this term is invented to make option file writing process easier and there are more that one way to define **Toolpath Axis Mode**. The postprocessor supports two suggestions about **Toolpath Axis Mode** definition - **Smart Axis Mode Definition** (which is used by default) and **traditional** which can be set for backward compatibility of older option files.

Smart Toolpath Axis Mode Definition

This is a method of **Toolpath Axis Mode** definition using properties of toolpath moves. A structure of a program shown below demonstrates areas where **Toolpath Axis Mode** is defined and where it is undefined:



- Value of **Toolpath Axis Mode** parameter between toolpaths could contain axis mode of a previous toolpath. But it is recommended to count it as **undefined** and use the parameter within toolpath only.

Each toolpath (i.e. each program segment between **Toolpath Start** and **Toolpath End** commands contains a sequence of moves. **Toolpath Axis Mode** for each toolpath is defined using following rules:

Toolpath Axis Mode	Satisfied conditions
3-axis	- **Tool Direction** of all moves inside the toolpath is the same and **vertical** i.e. equal to **(0,0,1)**. - **Tool Orientation** of all moves inside the toolpath is the same and equal to **(1,0,0)**. - **Tool Direction** and **Tool Orientation** are absent in the program. In this case they are assumed to correspond to **3-axis** case.
3+2	- **Tool Direction** of all moves inside the toolpath is the same. - **Tool Orientation** of all moves inside the toolpath is the same.
5-axis	- **Tool Direction** or **Tool Orientation** is changed within the toolpath.

- It could be seen that **3-axis** is a particular case of **3+2** and **3+2** is a particular case of **5-axis**.

Traditional Toolpath Axis Mode Definition

This method analyzes CLDATA records to define **Toolpath Axis Mode**. There are no toolpath borders in this case and this parameter is always defined. The rules used are:

Toolpath Axis Mode	Satisfied conditions
3-axis	**5000/3 (FROM)** record does not contain **Tool Direction**. This is the very basic property of **pure 3-axis** CLDATA: I 8I 22I 5000I 3I 0I 0 F+0000000000000000+0000F+0000000000000000+0000F+1000000000000000+0003;
3+2	**5000/3 (FROM)** record contains **Tool Direction**. This CLDATA property always means that the program is **multi-axis**: I 11I 40I 5000I 3I 0I 0 F+3587393569946289+0002F+4037593078613281+0002F+2000000000000000+0002 F+3406729159749328+0000F+2000349495862240+0000F+9186555302534097+0000; **9000/2/1 (MULTAX ON)** record was cancelled by **9000/2/0 (MULTAX OFF)** before current toolpath is started: I 4I 38I 9000I 2I 1; ... I 4I 40I 9000I 2I 0; ... I 11I 82I 5000I 5I 0I 0 F+4885429776957986+0002F+2350000000000000+0002F+1583375060258813+0002 F+4497201209304200+0002F+2350000000000000+0002F+1344863208252107+0001;
5-axis	**5000/3 (FROM)** record contains **Tool Direction**: I 11I 40I 5000I 3I 0I 0 F+3587393569946289+0002F+4037593078613281+0002F+2000000000000000+0002 F+3406729159749328+0000F+2000349495862240+0000F+9186555302534097+0000; **9000/2/1 (MULTAX ON)** record was NOT cancelled by **9000/2/0 (MULTAX OFF)** before current toolpath is started: I 4I 38I 9000I 2I 1; ... I 17I 71I 5000I 5I 0I 0 F+1005960559844971+0002F-5279238510131836+0002F+1225378616333008+0003 F-8924321993516674-0003F+6409995740022847+0000F+7675407153328475+0000 F+9940394401550293+0001F-

	5279238510131836+0002F+1225378616333008+0003 F+8924321993516674- 0003F+6409995740022847+0000F+7675407153328475+0000;

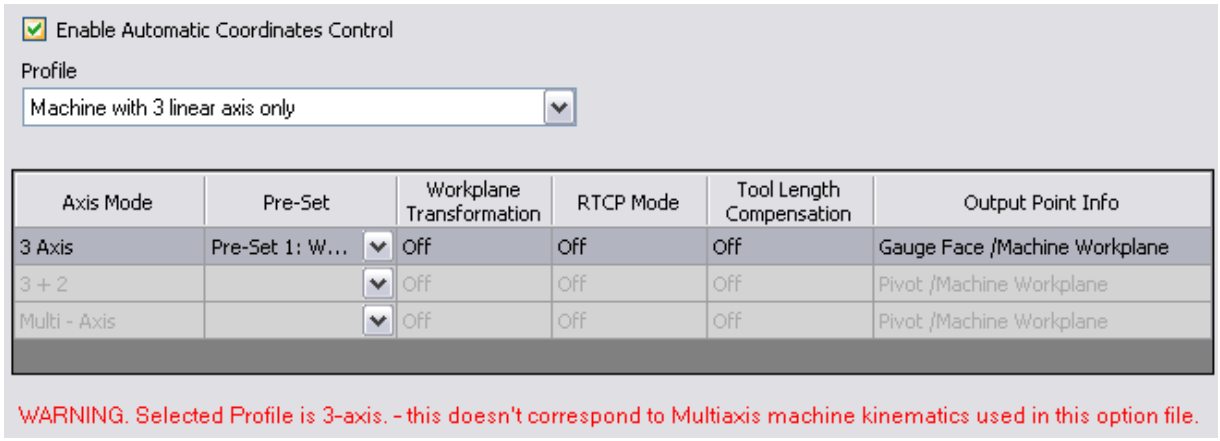
Simply speaking, in **multi-axis** CLDATA everything between **9000/2/1 (MULTAX ON)** and **9000/2/0 (MULTAX OFF)** is **5-axis**. Everything else is **3+2**.

Status and warnings

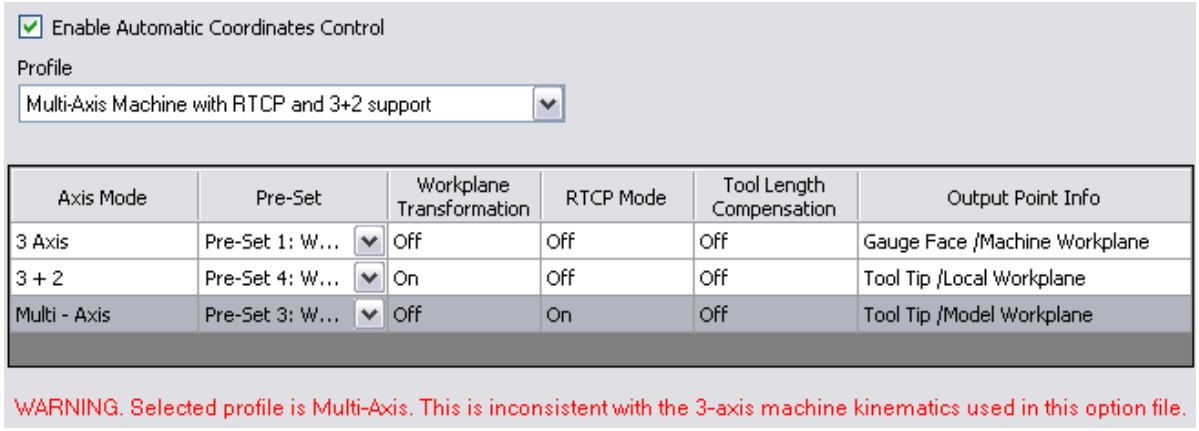
The **Status** field at the bottom of the ACC table shows whether the configuration selected is appropriate. If there are no problems, it displays **OK**. Otherwise, it displays **Warnings**.

Delcam PostProcessor can postprocess configurations for which warnings are displayed, but the output may give unexpected results.

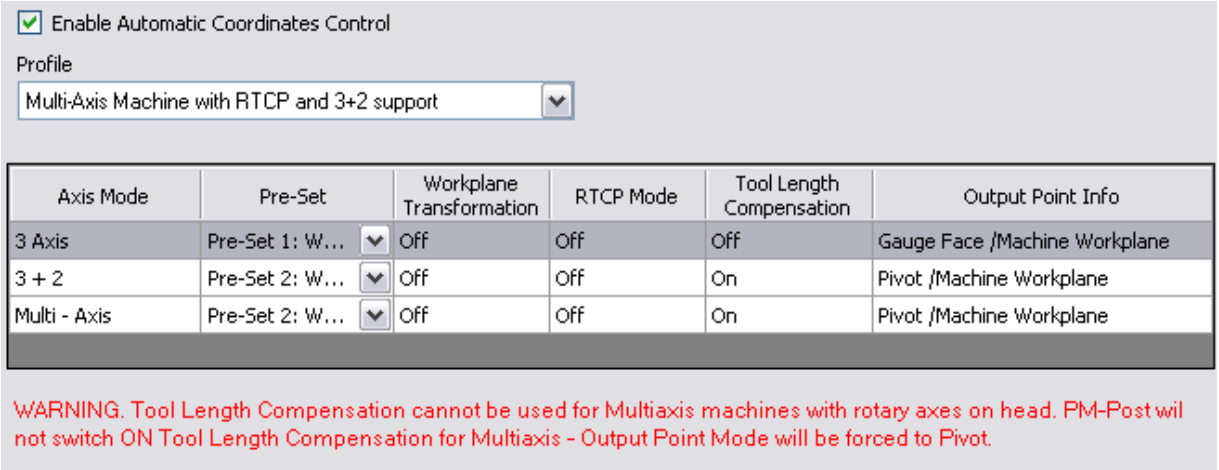
Information on how to correct a configuration **Warning** is displayed automatically under the **Profile** table:



This warning shows that the profile selected does not correspond to the machine kinematics used for this option file. In the above example, **Delcam PostProcessor** cannot use the rotary axes of the machine despite the **Multi-axis** kinematics. To fix this, either select another (multi-axis) profile, or change the machine kinematics to **3-axis**. There is a similar warning for 3-axis kinematics used with a multi-axis profile:

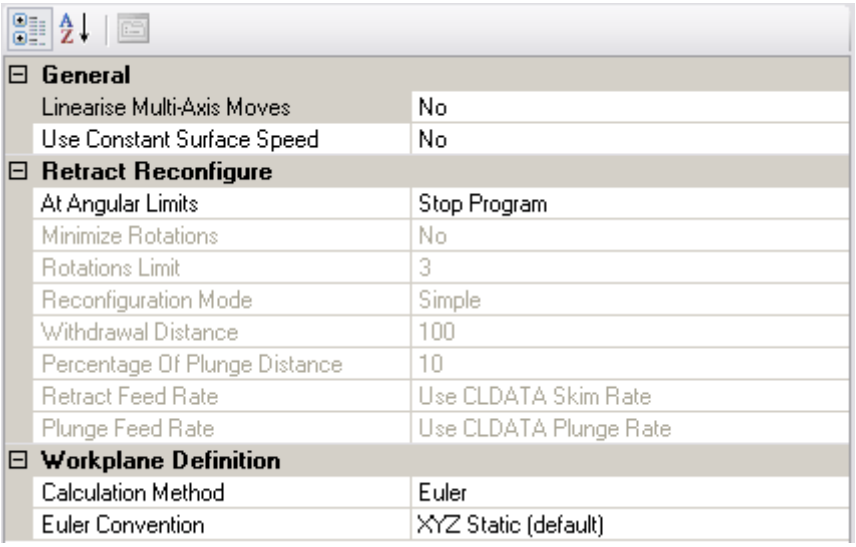


The following warning is displayed because **Tool Length Compensation Mode** cannot be applied when the tool direction is not parallel to the machine's Z axis (this is the compensation that is usually activated by the **G43** code and involves the lifting of a head by the tool length). As the tool direction is changed relative to the Z axis on machines with rotary axes on the head, **Delcam PostProcessor** does not apply **Tool Length Compensation Mode** for such machines. The **Warning** suggests how to rectify the problem.



Program Generation - Multi-axis configuration: general

Use the **Multi-Axis Configuration** view to define the settings for multi-axis processing.



Linearise Multi-Axis Moves - Use this option to specify how moves that exceed controller tolerance are treated:

- Select **Yes** to verify that all multi-axis moves are within the tolerance specified in **General** settings. When a move exceeds tolerance, **Delcam PostProcessor** splits it into two smaller moves.
- Select **No** to leave the original moves in CLDATA unchanged.

Euler Angles Style - Specifies angle style for **Workplane Euler A**,**Workplane Euler B** and **Workplane Euler C**. For more information see Angle Styles.

Machine Angles Style - Specifies angle style for **WP Machine A**, **WP Machine B**, **WP Machine C**, **Machine A**, **Machine B** and **Machine C**. For more information see Angle Styles.

Reset Axes. When a toolpath follows a 3+2 or 5-axis toolpath the machine rotary axes may not be in initial state. For example, a rotary table may be in the position corresponding to last machine state of previous toolpath. In order to force rotary axes to reset set this option.

- **Never** means not resetting axes at all (default).
- **Before Pure 3-axis** to reset only before a 3-axis toolpaths.
- **Before Any Toolpath** to reset before a toolpaths regardless of its axis mode.

Use Constant Surface Speed Feed Rate - In multi-axis moves where the pivot point is different from the tool tip point, the tool tip can move with a different speed to the feed rate specified in the CLDATA. Use this option to specify how you want to treat the surface speed rate:

- Select **Yes** to apply a constant feed rate at the contact point.
- Select **No** to apply no correction.

Calculation Method - specifies the method used to orient the tool for 3+2 processing when the machine controller supports working in a local workplane. Select:

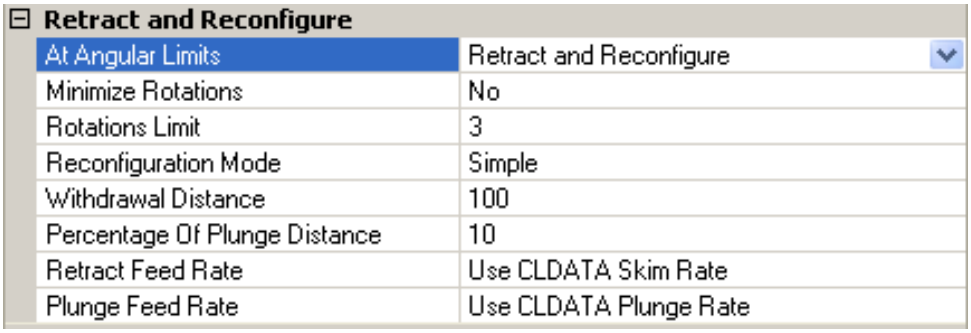
- **Machine** to specify the tool location using azimuth and elevation.
- **Euler** to specify the tool location relative to the local workplane's X, Y and Z values. If you select **Euler**, you must also select the **Euler Convention** used by the machine controller. For information on the **Calculation Method** and **Euler Convention** used by your controller, refer to the controller's manual.

Workplane Source - specifies the data used for workplane definition for a 3+2 toolpath. Actually, the workplane used in PostProcessor may differ from the original one commanded from the CAM system. Select:

- **Z of the Original Workplane** to specify that Z of the new workplane is aligned with Z of the original workplane. Other axes are defined according to the **Calculation Method**. If **Calculation Method** is **Euler** the new workplane is exactly the same as the original one.
- **Original Workplane** to specify the new workplane is exactly the same as the original one. In this case workplane X does not coincide with the orientation vector the toolpath appears as 5-axis rather than 3+2.
- **Tool Vector and Orientation** to specify the new workplane constricted from the tool vector and orientation vector. This option is preferrable is the program uses orientaion vectors. Nevertheless, the local workplane may differ from what was used in the CAM system.
- To enable 3+2 support, you must also set the **Profile** to **Multi-Axis machine with RTCP and 3+2 support** in the **Coordinates Control** settings, and activate the **Set Workplane On** and **Set Workplane Off** commands.

Ignore Toolpath Workplane Shift - specifies whether to use the linear part of the toolpath workplane. In other words, if set **Yes**, **Workplane Origin X**, **Workplane Origin Y** and **Workplane Origin Z** are always zeros.

Program Generation - Multi-axis configuration: retract and reconfigure



Retract and Reconfigure	
At Angular Limits	Retract and Reconfigure
Minimize Rotations	No
Rotations Limit	3
Reconfiguration Mode	Simple
Withdrawal Distance	100
Percentage Of Plunge Distance	10
Retract Feed Rate	Use CLDATA Skim Rate
Plunge Feed Rate	Use CLDATA Plunge Rate

The **Retract and Reconfigure** options define the behaviour of **Delcam PostProcessor** when a machine reaches its rotary axis limit during the processing of a trajectory.

These limits are defined in **Machine Kinematics** for standard multi-axis templates, or in the MTD file for MTD-based kinematics.

- Only angular limits are taken into account. If the linear axis limit is exceeded, **Delcam PostProcessor** always outputs an error and stops processing.

At Angular Limits - sets the behaviour of **Delcam PostProcessor** when an angular limit is reached. Select:

- **Stop Program** to stop processing and display an error message when an angular limit intersection is met. No other **Retract and Reconfigure** options are selectable.
- **Reset Angles Only** to reset the current value of the table axis by one or more full rotations. This option can be applied if a controller uses numerical limits for rotary axes. For example, you can use this option if table rotations on a machine are unlimited but the installed NC controller only accepts a table angle between -720 and 720. In this case, when a value of 720 is reached, the current value of the table axis can be reset to 360 or 0 without retracting the tool and reconfiguring the machine.

  Only full rotations are applied. If it is not possible to make a full rotation about the axis without exceeding the controller's limit, **Delcam PostProcessor** stops processing and displays an error message.

  Activation of this option also activates the **Rotations Limit** field:

  - A controller must support the resetting of angles for this option to be used in the option file.
- **Retract and Reconfigure** to activate the **Retract and Reconfigure** procedure when an angular limit is reached. The procedure uses the parameters specified.

  During the procedure, a tool retracts along its current direction by the **Withdrawal Distance**, using the **Retract Feed Rate**. Rotary angles are not changed during the retract. **Delcam PostProcessor** then recalculates the coordinates of the retracted position. It tries to find another solution for which all angles satisfy the limits. If successful, **Delcam PostProcessor** inserts the reconfigured position into the NC output and starts to move the tool back to the position from which the retract started using the **Retract Feed Rate**. Only the last part of the move uses the **Plunge Feed Rate**; the

length is defined by the **Percentage Of Plunge Distance**. After the tool returns to its initial position, it continues the move that was interrupted by the **Retract and Reconfigure** procedure.

Minimize Rotations - this parameter specifies the type of reconfiguration:

- **No** - before searching for closer solutions, **Delcam PostProcessor** tries to change the angle that exceeds its limit by making as many full circular revolutions as required (up to the maximum specified in the **Rotations Limit** field) towards the centre of the interval between the minimum and maximum limits.

  Example - if angle A has minimum and maximum limits [-720;+720], and A reaches value 720, the reconfigured position sets A equal to 0:

  - Position after retract - N110 X10 Y20 Z130 A720 B50
  - Position after reconfiguration - N120 X10 Y20 Z130 A0 B50
  - If it is not possible to make one full rotation, **Delcam PostProcessor** tries to find a solution using smaller values (see **Yes**).

- **Yes -** when reconfiguring, **Delcam PostProcessor** tries to find smaller new angle values for the reconfigured position rather than generate full 360 degrees rotations.

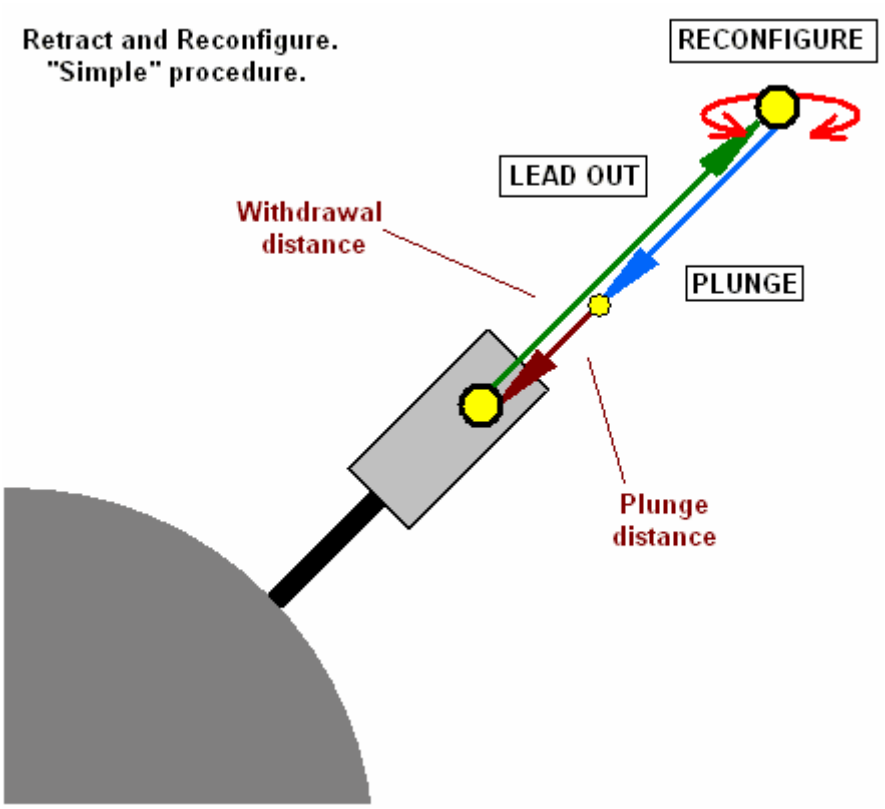
  Example - if angle A has minimum and maximum limits [-720;+720], and A reaches value 720, the reconfigured position may cause angles A and B to be rotated through 180 degrees:

  - Position after retract - N110 X10 Y20 Z130 A720 B50
  - Position after reconfiguration - N120 X10 Y20 Z130 A540 B230

Rotations Limit - this field is activated when **Minimize Rotations** is set to **No**. It limits the number of full circular revolutions towards the centre of the interval that **Delcam PostProcessor** can make when reconfiguring its position. **Delcam PostProcessor** never exceeds the limit specified and rotations are stopped when the middle of the interval or the limit is reached.

Reconfiguration Mode specifies the behaviour when the rotary axis limit is reached:

- **Simple** sets the simplest possible behaviour for **Delcam PostProcessor**, for example causing the reconfigured position to be inserted into a numbered block directly after the **Position after retract** (as in the examples above). Also, if linearisation is used in multi-axis moves, the reconfigured move is not linearised as it is assumed a machine can handle this move without problems.



Sample command order for simple mode:

Move Linear (cutting move)

Move Linear (cutting move)

Retract and Reconfigure Start (start of the reconfiguration motion)

Move Linear (retract to withdrawal distance)

Move Linear (reconfiguration/turning)

Move Linear (approach)

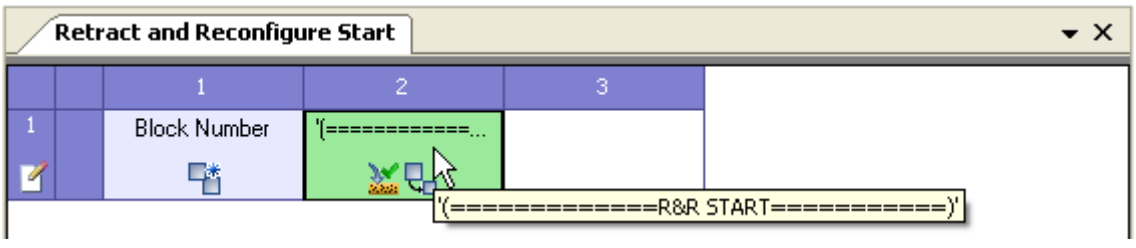
Move Linear (plunging move)

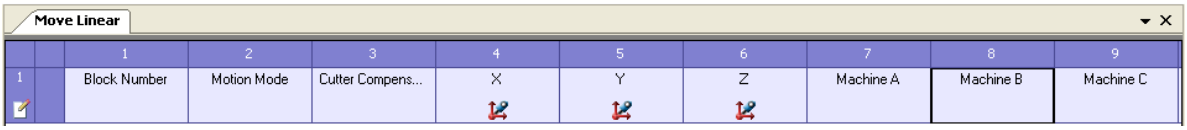
Retract and Reconfigure End (end of the reconfiguration motion)

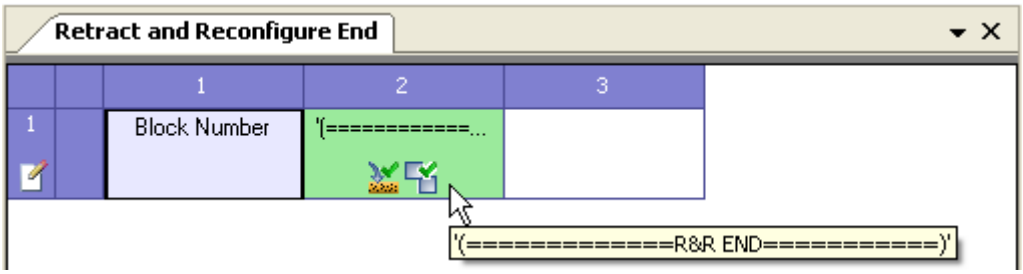
Move Linear (cutting move)

Move Linear (cutting move)

Example commands:





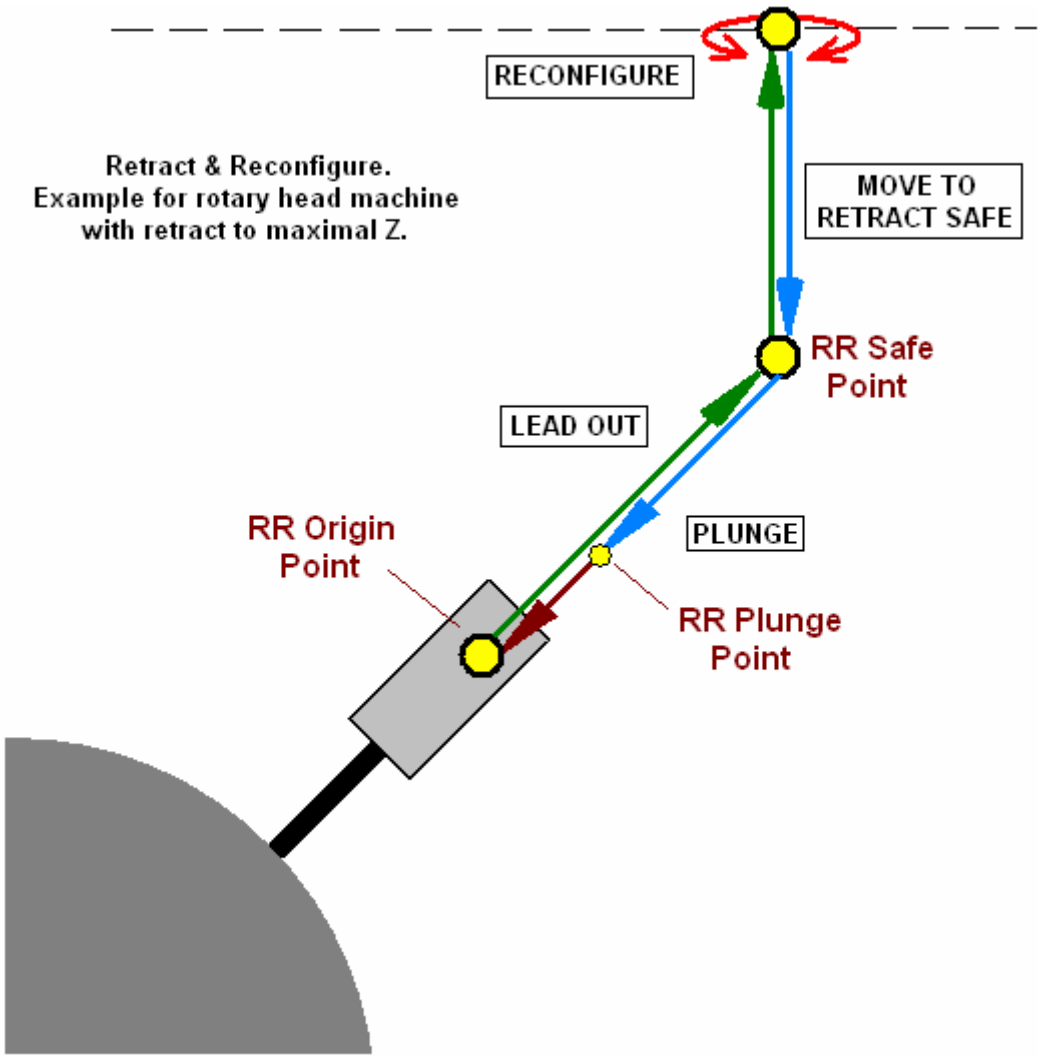


Retract and Reconfigure Start and **Retract and Reconfigure End** commands are used for entering some comments. Inner moves are realized by a standard **Move Linear** command. **Delcam PostProcessor** generates the sequence of moves internally using specified settings (see the setting page above). The feed rate is set automatically.

Example results:

N271 X-.002 Z29.382 A.000 B180.000
N272 (========== R&R START ==========)
N273 Z129.382 A.000 B180.000F9999
N274 X.002 Y-179.891 A180.000 B.000
N275 Z39.382 A180.000 B.000
N276 Z29.382 A180.000 B.000F750
N277 (========== R&R END ==========)
N278 X0 Z29.626 A180.000 B.262F2412

Manual allows manual customization of all moves in current reconfiguration motion. **Delcam PostProcessor** generates automated output for the Reconfigure for Manual Mode command. The user defines all moves for retracting and approaching by hand inside the Retract &Reconfigure Start and Retract&Reconfigure End commands.



Sample command order for manual mode:

Move Linear (cutting move)

Move Linear (cutting move)

Retract and Reconfigure Start (start of the reconfiguration motion, retract moves)

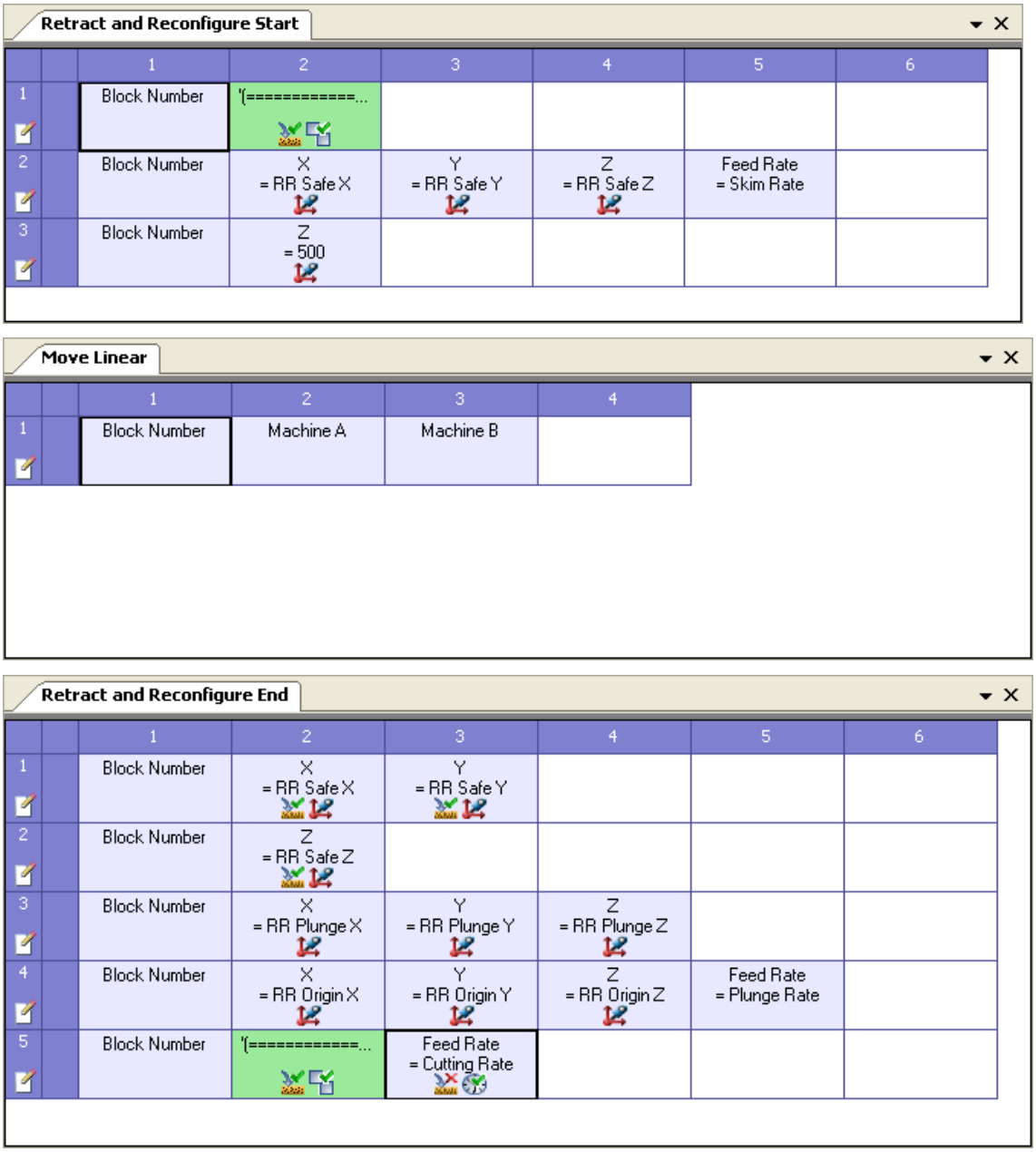
Reconfigure for Manual RR (reconfiguration)

Retract and Reconfigure End (end of the reconfiguration motion, approach moves)

Move Linear (cutting move)

Move Linear (cutting move)

Example commands:



Example results:

N165 X-120.592 Z-44.912 B-43.952
N166 X-122.712 Z-47.286 B-45.000
N167 (========== R&R START ==========)
N168 X-193.423 Z23.425 F9999
N169 Z500
N170 B315.000
N171 X-193.423 Y179.891
N172 Z23.425
N173 X-129.783 Z-40.215
N174 X-122.712 Z-47.286 F250
N175 (========== R&R END ==========)
N176 X-123.662 Z-48.378 B314.523F500
N177 X-126.661 Z-51.924 B312.991
N178 X-129.586 Z-55.548 B311.451

Withdrawal Distance - the distance of the retract move where a tool moves up out of a workpiece along its direction vector. You must enter a positive value that provides as safe a reconfiguration as possible. The default value is 100 units (so, it could be 100 millimetres or 100 inches, depending on which units are active).

Percentage Of Plunge Distance - the **Retract and Reconfigure** procedure uses the **Retract Feed Rate** specified in the next field. However, when a tool goes back to touch a workpiece (that is, when it makes a plunge), its speed must be reduced. This percentage determines the length of the plunge - for example, if the **Withdrawal Distance** is equal to **150** and the **Percentage Of Plunge Distance** is **10**, the plunge has a length of **15**.

Retract Feed Rate - the speed of a tool during retracts. The possible states are:

- **Use CLDATA Skim Rate** - the value of the skim rate from CLDATA is used for the **Retract and Reconfigure** procedure.
- **Specify Feed Rate** - this option opens the edit box where you can enter the value of the feed rate you want to use for the **Retract and Reconfigure** procedure.

Plunge Feed Rate - the speed of the tool when it is approaching the workpiece after reconfiguration. The possible states are:

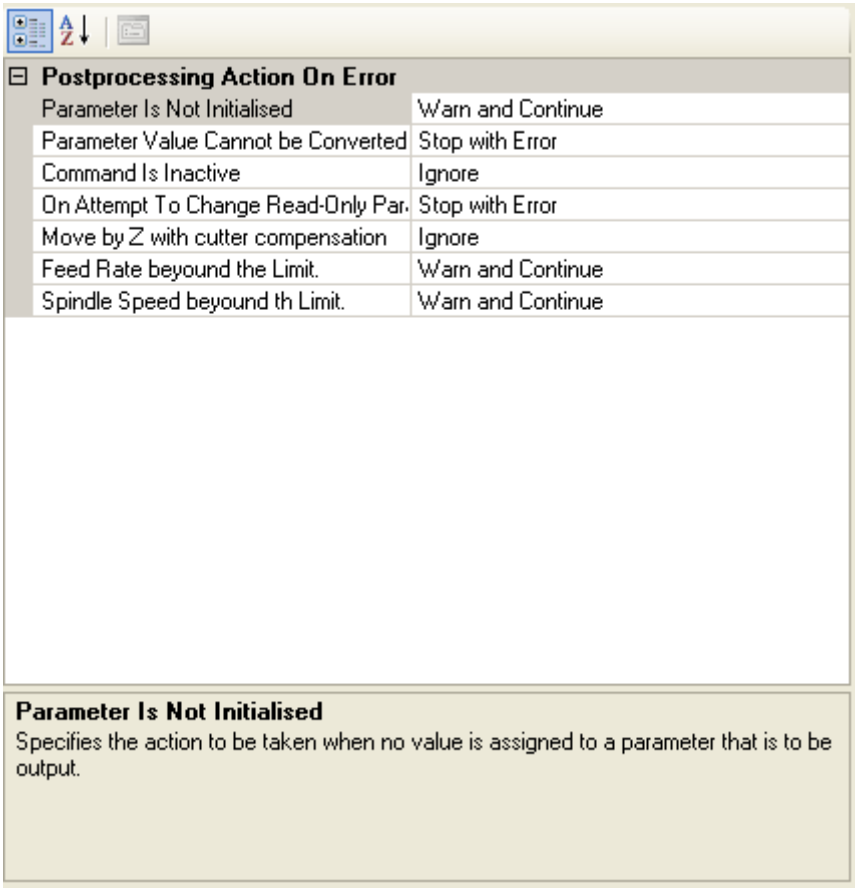
- **Use CLDATA Plunge Rate** - the value of the plunge rate from CLDATA is used for this move.
- **Specify Feed Rate** - this option opens the edit box where you can enter the value of the feed rate that you want to use for the plunge.

Program Generation - Error Response

The **Error Response** page enables you to specify the actions to be taken when the Postprocessor encounters an error.

To open the **Error Response** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.
2. In the **Option File Properties** dialog, select **Program Generation - Error Response**.



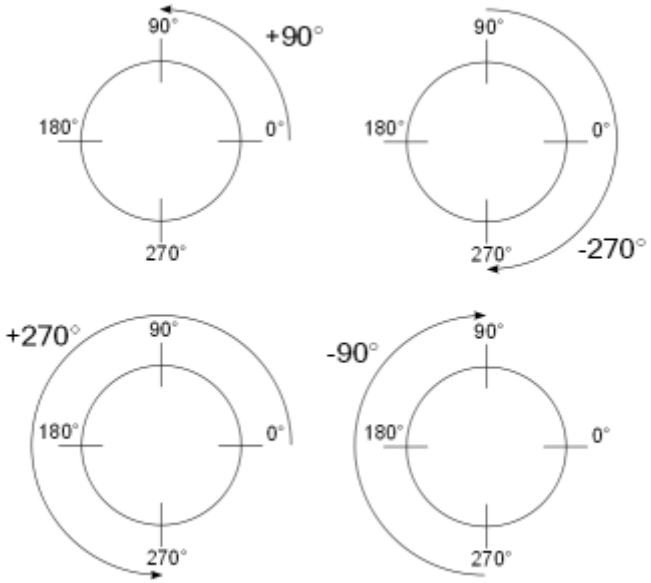
Parameter	Description
Parameter is Not Initialised	Specifies the action to be taken when no value is assigned to a parameter that is to be output.
Parameter Value Cannot be Converted to specified type	Specifies the action to be taken when a value cannot be converted to the type of the assigned parameter, such as when a text string is assigned to a numeric parameter.
Command is Inactive	Specifies the action to be taken when the Postprocessor attempts to output a Command that has not been activated.
On Attempt To Change Read-Only Parameter	Specifies the action to be taken when the Postprocessor attempts to change a parameter that is defined as read-only.

Angle Styles

The PostProcessor supports four different styles of specifying angles:

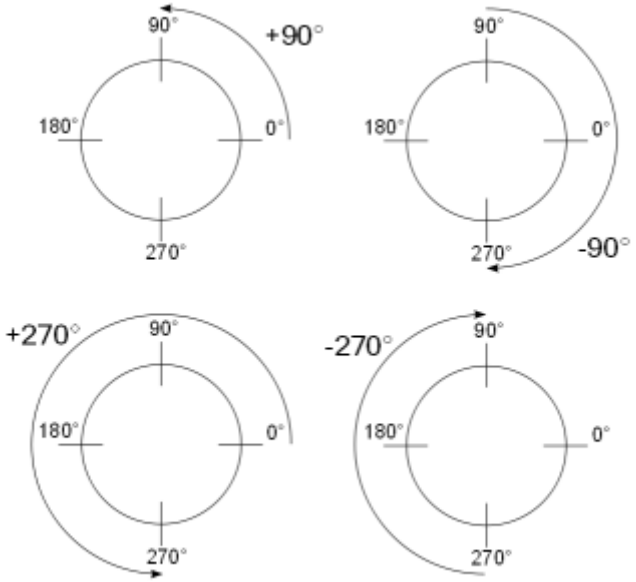
1. ANSI/EIA RS-274-D

In this style, the value of the angle specifies the angular position measured from zero in the positive direction. The sign of the angle indicates the direction of rotation.



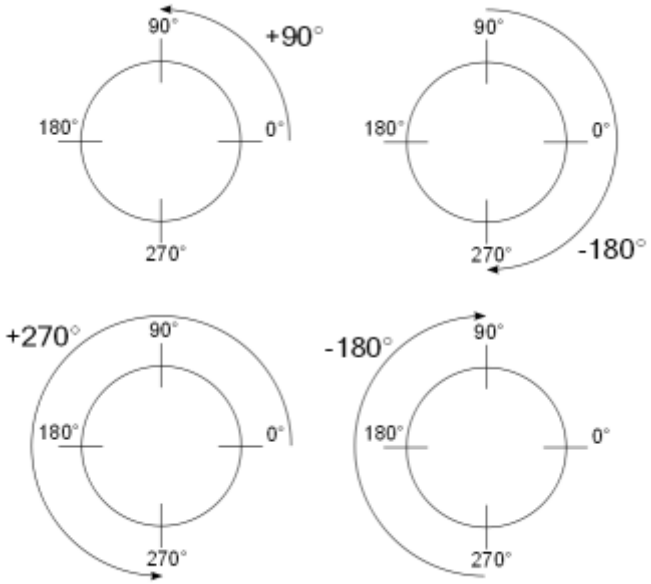
2. Mathematical

In this style, the value of the angle specifies the angular position measured from zero and the sign indicates the direction of measurement. The sign of the angle also indicates the direction of rotation.



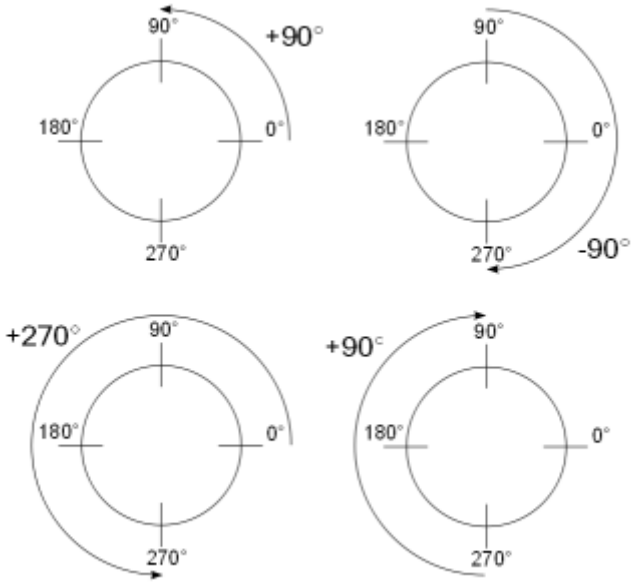
3. Relative

In this style, the value of the angle specifies the angular distance measured from the current position and the sign indicates the direction of measurement. The sign of the angle also indicates the direction of rotation.



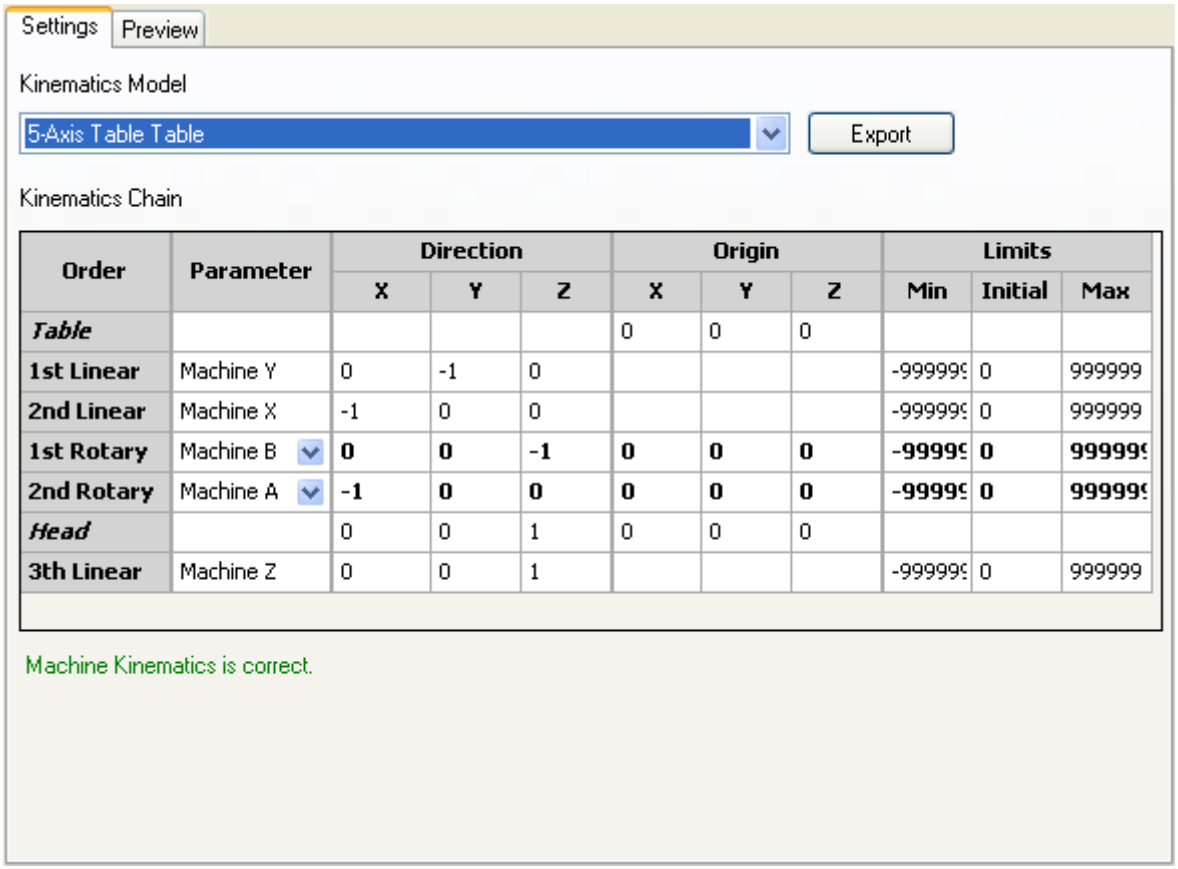
4. Winding and Unwinding

In this style, the value of the angle specifies the angular position measured from zero and the sign indicates the direction of measurement. The sign of the difference between the angle and the current position indicates the direction of rotation.



Machine kinematics properties

Use the **Machine Kinematics** page to define the number of axes on the machine tool, and the number, type and locations of the machine's tilting axes.



Order	Parameter	Direction			Origin			Limits		
		X	Y	Z	X	Y	Z	Min	Initial	Max
Table					0	0	0			
1st Linear	Machine Y	0	-1	0				-999999	0	999999
2nd Linear	Machine X	-1	0	0				-999999	0	999999
1st Rotary	Machine B	**0**	**0**	**-1**	**0**	**0**	**0**	**-9999**	**0**	**99999**
2nd Rotary	Machine A	**-1**	**0**	**0**	**0**	**0**	**0**	**-9999**	**0**	**99999**
Head		0	0	1	0	0	0			
3th Linear	Machine Z	0	0	1				-999999	0	999999

Kinematic model - Choose an option from the drop-down list to specify the type of machine for which you want to create an NC program file. Select:

- **3-Axis** for machines with three linear axes (X, Y and Z).
- **4-Axis Head or Table** for machines with three linear axes and a rotary axis. The rotary axis can be located on the machine head or table.
- **5-Axis** for machines with three linear and two rotary axes. Select:
  - **Table - Table** when both rotary axes are located on the table.
  - **Table - Head** (also known as Table - Spindle) when one rotary axis is located on the table, and the other is located on the head.
  - **Head - Head** (also known as Spindle - Spindle) when both rotary axes are located on the head.
- to load a model of the machine for which you are creating an option file. The model must be defined in an MTD (Machine Tool Data) file that includes the transformations between model and machine coordinates. Because this option loads the kinematics data from the model file, it also enables you to create programs for 6-axis machines with three rotary axes.

  The path of the selected file is displayed in the **MTD Path** field and stored in the option file.

The view changes to display the fields required for the selected machine.

Export - Click this button to save the currently selected model to an MTD file.

Preview - Click the **Preview** tab to display the selected **Kinematic model**. If you selected , the model specified in the **MTD Path** is displayed. Otherwise, a generic model is displayed.

Kinematics chain

The **Kinematics chain** contains the details of the machine axes and information about the location of the tool:

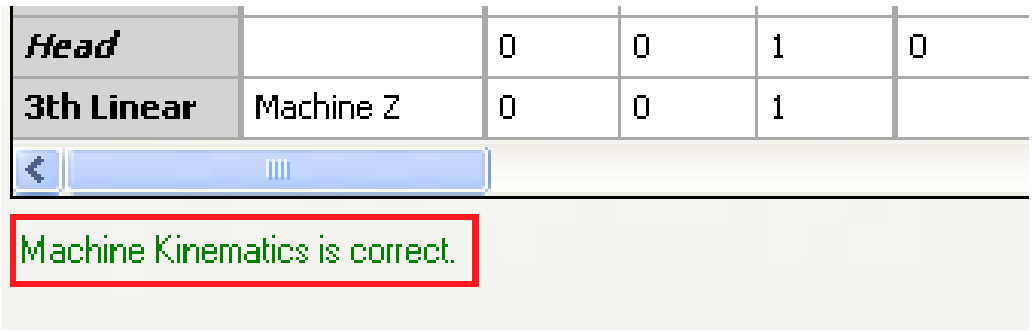
- **Order** lists the elements of the machine.
- **Parameter** lists the parameters that control each machine axis. To display or update the settings for an axis, click **Edit**. To change the **Parameter** associated with a rotary axis, select an entry in the drop-down list.
- **Direction** specifies the vector for each axis. All origins and directions are defined in the workplane in which the machine kinematics are defined. The **Head** values show the initial direction of the tool.
  - This workplane is used by **PowerMILL** to save cutfiles (**CLDATA Output Workplane**).
- **Origin** specifies the origin for each axis.

The **Head** values specify the start position of the tool (a tool tip point for a tool with zero length). This point is defined in the **Machine workplane** and must be the same as the **CLDATA Output Workplane** when CLDATA files are prepared in **PowerMILL**.

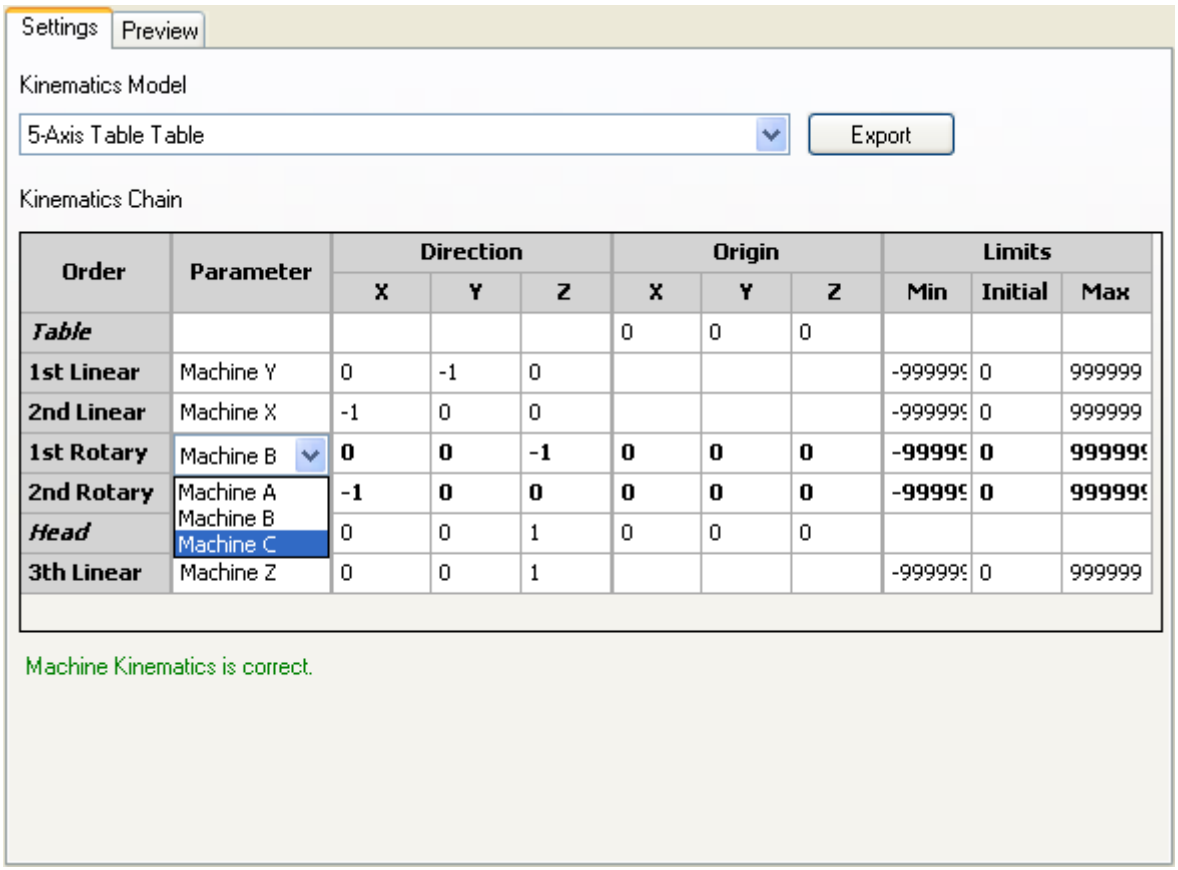
The **Table** values specify the **Origin** point where the billet is located. This setting is not currently used in **PowerMILL**.

- **Limits** specifies the **Min**imum and **Max**imum values for each axis. The **Initial** value specifies the start of the coordinate.

Delcam PostProcessor verifies your axis configurations, and displays the status below the table. For example:



Defining rotational axes



Order	Parameter	Direction			Origin			Limits		
		X	Y	Z	X	Y	Z	Min	Initial	Max
Table					0	0	0			
1st Linear	Machine Y	0	-1	0				-999999	0	999999
2nd Linear	Machine X	-1	0	0				-999999	0	999999
1st Rotary	Machine B	**0**	**0**	**-1**	**0**	**0**	**0**	**-99999**	**0**	**99999**
2nd Rotary	Machine A / Machine B / Machine C	**-1**	**0**	**0**	**0**	**0**	**0**	**-99999**	**0**	**99999**
Head		0	0	1	0	0	0			
3th Linear	Machine Z	0	0	1				-999999	0	999999

To change the letter designating a rotational axis. (for example, from **B** to **C**), select it from the drop-down list in the **Parameter** column.

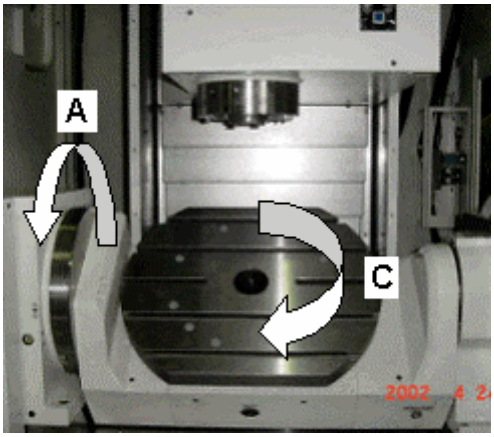
Each rotational axis is defined by the following attributes:

- Rotary Axis Location
- Rotary Axis Vector
- Rotary Axis Origin
- Rotary Axis Limits
- You can control the C axis position from within PowerMILL by entering the C axis value in NC program format. In Delcam PostProcessor this value automatically sets the Machine C value prior to machining.

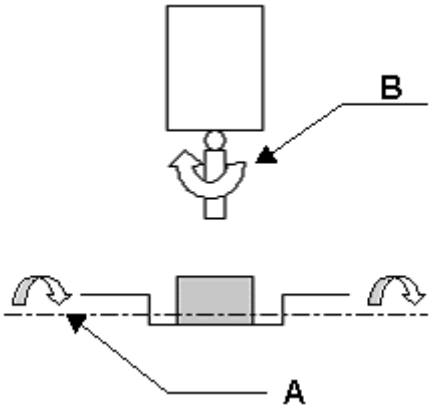
Rotary axis location

Although the configurations of rotary axes vary widely, there are only three basic types of 5-axis machine configuration:

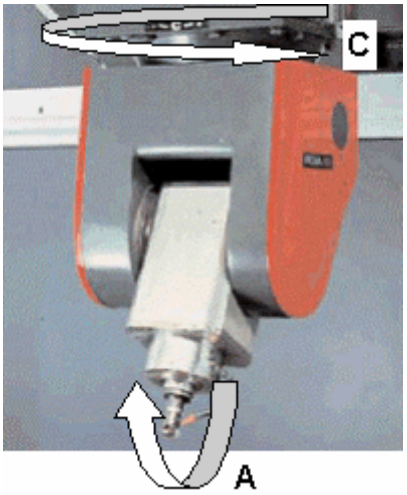
- **TABLE- TABLE** - both axes are on the table.



- **TABLE - HEAD** - the primary axis is on the head (or spindle) and the secondary axis is on the table. This is also known as Table - Spindle.



- **HEAD- HEAD** - both axes are on the head (or spindle). This is also known as Spindle - Spindle.



Each of these configurations has only two rotary axes:

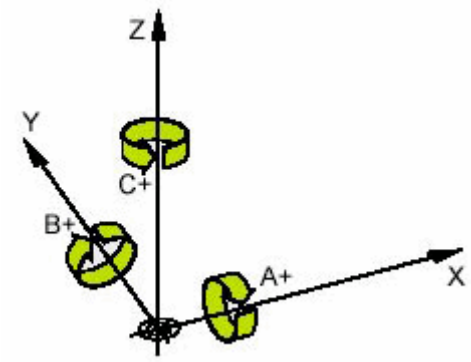
Table - the rotations cause the machine table to rotate, so that the part (and table) is moved around the machine head.

Head - the rotations cause the machine head to rotate, so that the tool tip direction is moved around the part.

Rotary axis vectors

Rotary axis vectors define the angle of rotation about the principal axes:

- Rotations about the **X axis** are normally called **A Axis** rotations.
- Rotations about the **Y axis** are normally called **B Axis** rotations.
- Rotations about the **Z axis** are normally called **C Axis** rotations.
- Rotations about other axes are defined by the machine control manufacturer and are not standardized.



The diagram shows the direction of positive angle rotations about an axis, assuming that the vectors are set as (1,0,0), (0,1,0) and (0,0,1). This is called the *corkscrew rule*.

Rotary axis origin

This locates the origin of the rotation and is specific to each machine type.

Rotary axis limits

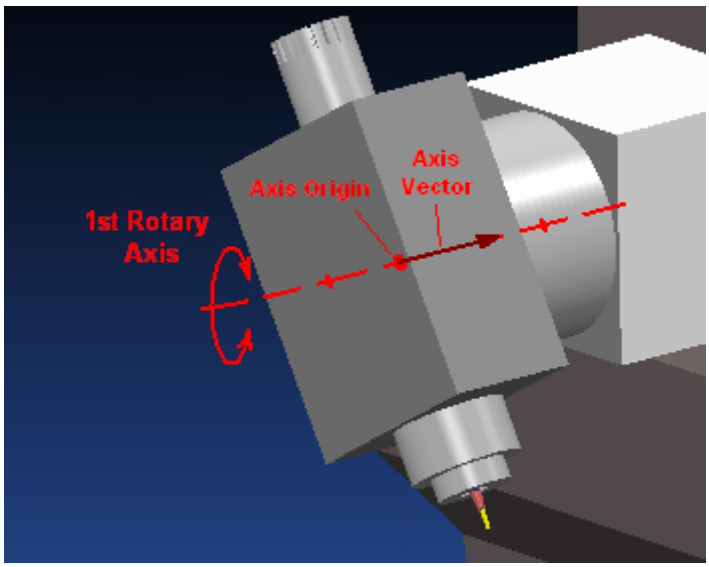
These define the machine tool's angular limits, which in turn define the angle at which a tool can be positioned, relative to the part, while cutting a multi-axis toolpath.

For example, if you can process ten turns from the zero position in the clockwise direction, and ten turns in the anticlockwise direction, the range is **+/- 3600**.

4-Axis head machines

To realize kinematics with one rotary axis in the option file, the **Kinematic Model** field can be set to **4-Axis Head** or **4-Axis Table**.

The following is an illustration of a configuration with the rotary axis on the head:

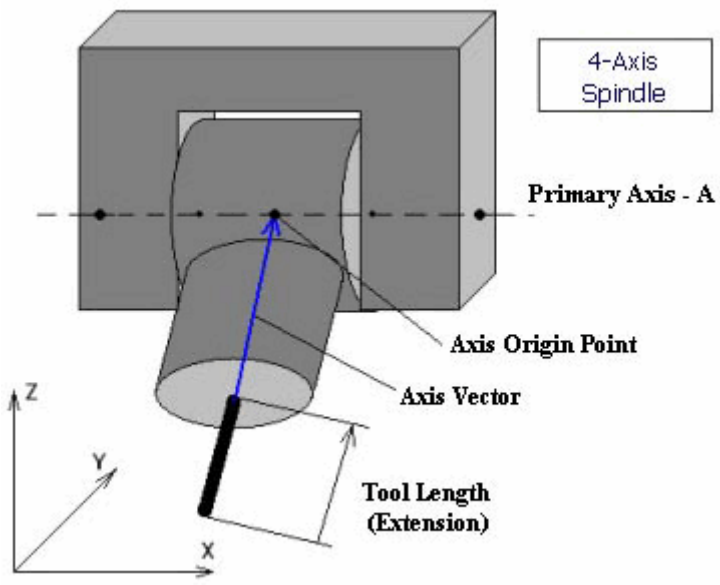


To set up this configuration, specify attributes for one rotary axis and set the initial tool vector. In the illustration, the axis is directed along Y linear axis, so it is the **B Axis**:

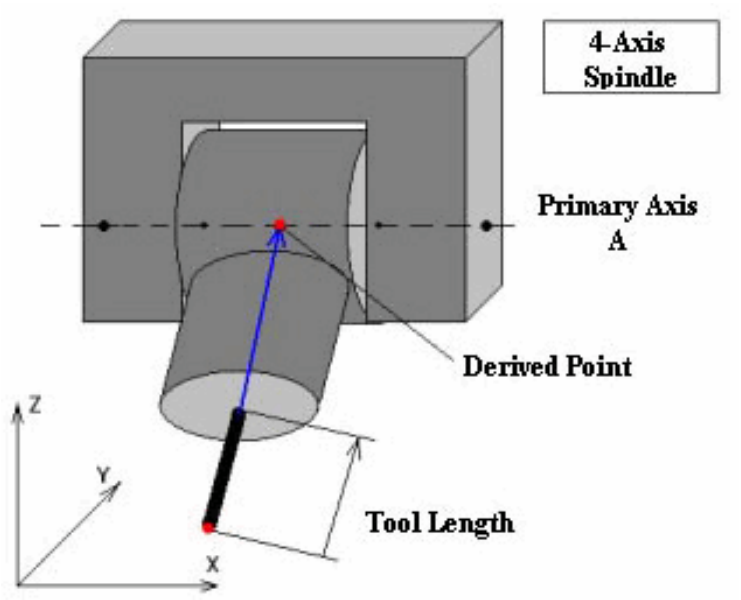
1st Rotary	Machine B	0	1	0	0	0	925	-90	0	90

Direction - the axis vector must NOT be collinear with the initial tool vector as rotations about the axis do not change the tool orientation.

Origin - the rotation axis is linked to the point on the tool holder where the tool length starts. The blue arrow in the diagram below shows how to determine the origin point. When defining the origin, ensure that all rotational axes on the machine tool are zero.



The rotary axis is along the X axis, so it is called the **A Axis**.



If a **Postprocessor** does not use dynamic compensation for outputting multi-axis moves, a **Pivot Point**, with coordinates different from the tool tip position, is used to output moves to an NC program. In the case of 4-axis kinematics, a pivot point is calculated using the formula:

Pnc = Pcl + TL*TV + Delta(P)

where:

- **Pnc =** pivot point
- **Pcl =** tool tip coordinates from CLDATA
- **TL =** tool length
- **TV =** current normal tool vector (so *TL*TV* is a tool vector whose module is equal to the tool length).
- **Delta(P)** - an origin vector (the blue arrow in the figure). It depends on the current rotation about the primary axis (**P**). In this case, the pivot point is the same as the tilted axis origin point. The main requirement is that the pivot point does not change its location in space when the machine rotates around the rotating axes. However, **Delcam PostProcessor** only supports cases where the pivot point lies on a rotational axis.

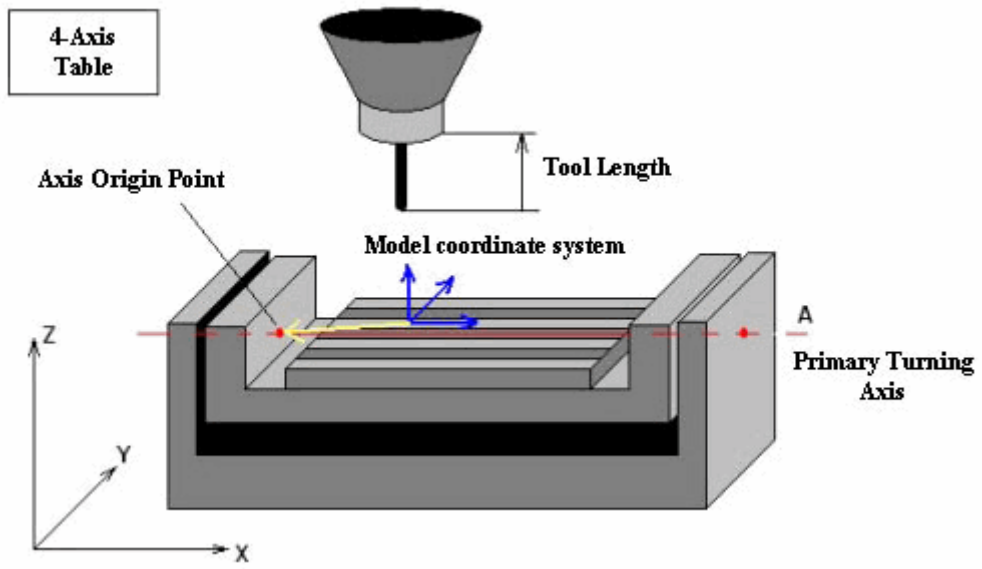
If you move an axis origin point along its axis, it has no effect on the axis location in space, so you can always define the origin point to coincide with the pivot point.

If a machine supports more than one way of configuring multi-axis processing, choose the variant where an axis origin point is not used in calculations, so tool tip coordinates rather than pivot point coordinates are output to the NC program. In this case, **Delcam PostProcessor** supports dynamic compensation for multi-axis moves and all origin points can be set to zero (0,0,0).

4-Axis table machines

To realize kinematics with one rotary axis in the option file, the **Kinematic Model** field can be set to **4-Axis Head** or **4-Axis Table**.

The following is an illustration of a configuration with the rotary axis on the table:

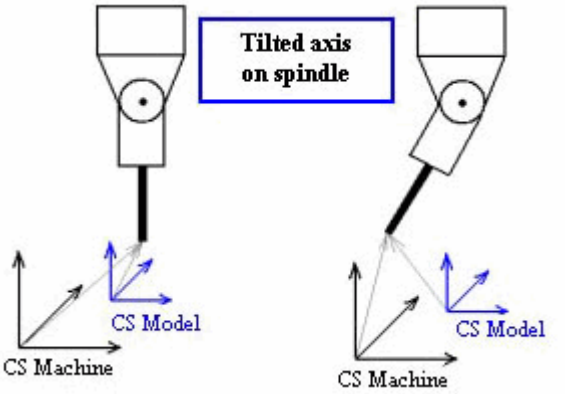


The rotary **A Axis** is along the machine X. The origin axis vector (which defines the origin point) is shown by the yellow arrow. The model workplane rotates about this axis together with the table (and the part).

Axis Vector - the axis vector must NOT be collinear with the initial tool vector as rotations about the axis won't change the tool orientation.

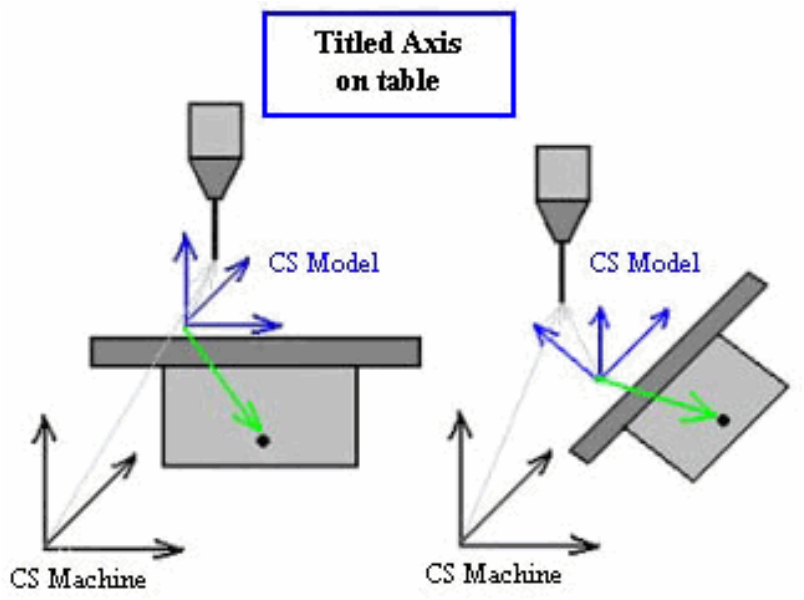
Axis Origin - the rotational axis origin point is set relative to the **Model Workplane** (that is, relative to the workplane where the CLDATA trajectory was generated by **PowerMILL**).

Relationships between Model Workplane and Machine Workplane during Multi-Axis Processing



In the case of a head-based rotating axis, a model (or part) does not change its position in space when rotating around this axis. The positional relationship between the model and the machine workplanes is unchanged. So, a rotation around the rotary axis does not cause a part to move or rotate in the machine workspace. As a result, it is sufficient to calculate a rotary axis origin point correctly to output it into an NC program. It is only necessary to know the tool tip

coordinates and the tool tip direction and to compensate for tool length and add the origin vector.



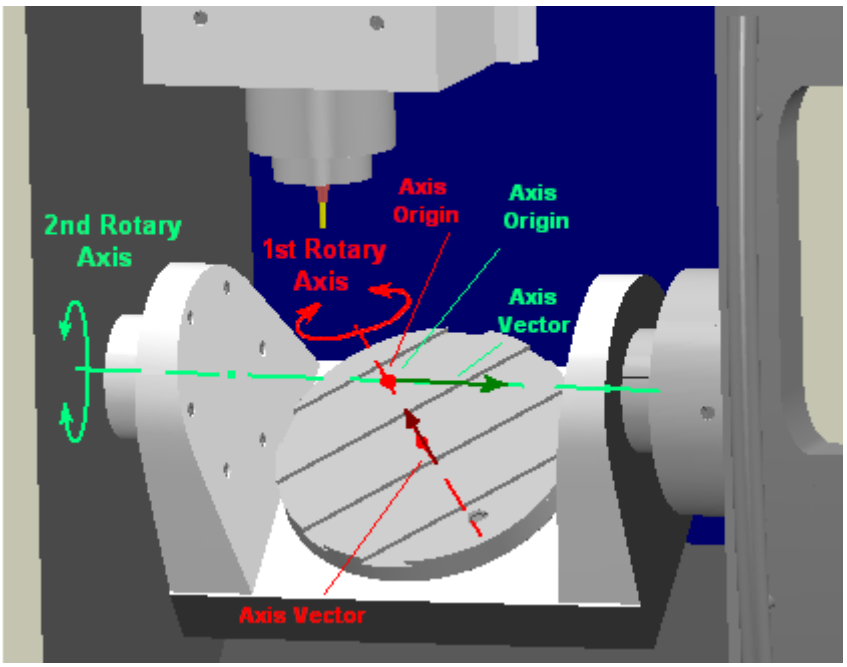
The case of a table-based rotating axis is more difficult. When rotating about the rotary axis, the model workplane position is changed relative to the machine workplane (the part rotates in space). You must define the correlation between these two coordinate systems (it correlates the model position in space with the tool tip). The simplest way to do this is to define the rotary axis origin point relative to the model workplane. In this case, the origin point does not change its coordinates in the machine workplane by default and you can then follow the model workplane rotation and calculate its position relative to the machine workplane (which in turn allows you to define the tool tip at the correct point in space at any point in time). The origin vector is shown by the green arrow in the above illustration.

The origin point must be defined for the initial state of the machine in the **Delcam PostProcessor** configuration when the coordinate of the rotary axis is zero.

Where the pivot point is the same as a tool tip point (or end point of a tool holder if **Tool Length Compensation Mode** is **On** in **Delcam PostProcessor**), the calculation of the correct position of the tool tip in the machine workplane is only necessary if dynamic compensation is **Off**.

The calculation of the axis origin point relative to the model workplane means there is only one valid place for the part on the table (if the part is moved, the rotary axis origin point is different. This results in a similar difference in the tilted axis origin point, and the machine kinematics in **Delcam PostProcessor** will not correspond to this configuration).

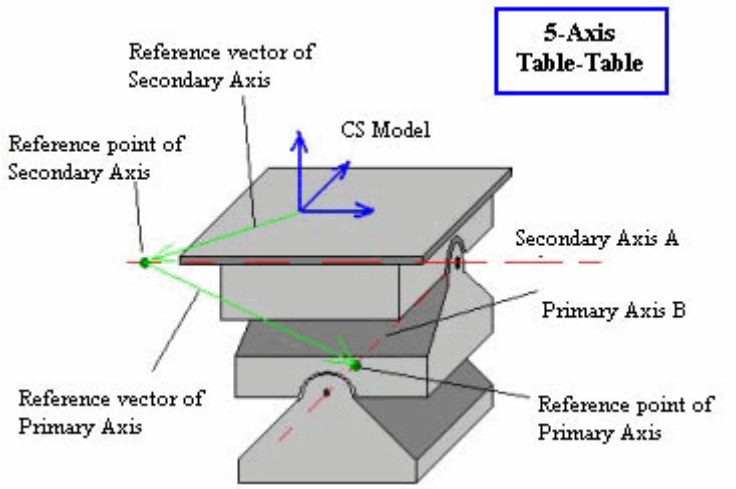
5-Axis table - table machines



Axis Vector - a typical configuration of the **Table - Table** class is shown above. The first rotary axis (A) has a direction of 1,0,0. The second rotary axis (C) has a direction of 0,0,1.

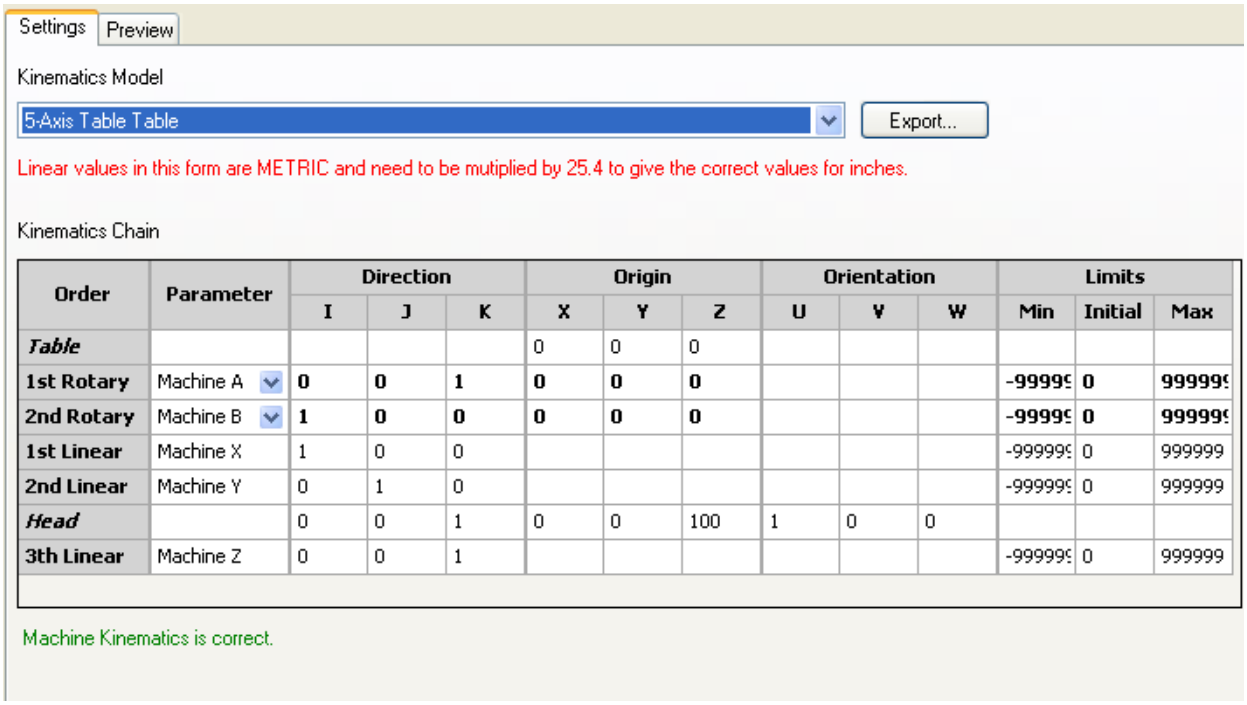
The first rotary axis position remains unchanged during rotations as its motor just 'rocks the cradle'. The second rotary axis motor is inside the 'cradle' and its direction is changed when the 'cradle' tilts.

Axis Origin - the secondary rotational axis origin point is set relative to the **Model Workplane** (relative to the workplane where the CLDATA trajectory was generated by **PowerMILL**). The primary rotational axis origin point is set relative to the origin of the secondary axis.



The origin point must lie on the axis it links. One complex variant is shown in the diagram above, although simpler cases are more common.

The example at the top of this topic has intersecting A and C axes and this greatly simplifies the configuration. If you place a model workplane at the point of intersection and define the axes' origin point as the intersection point, you get both axis origins as 0,0,0:



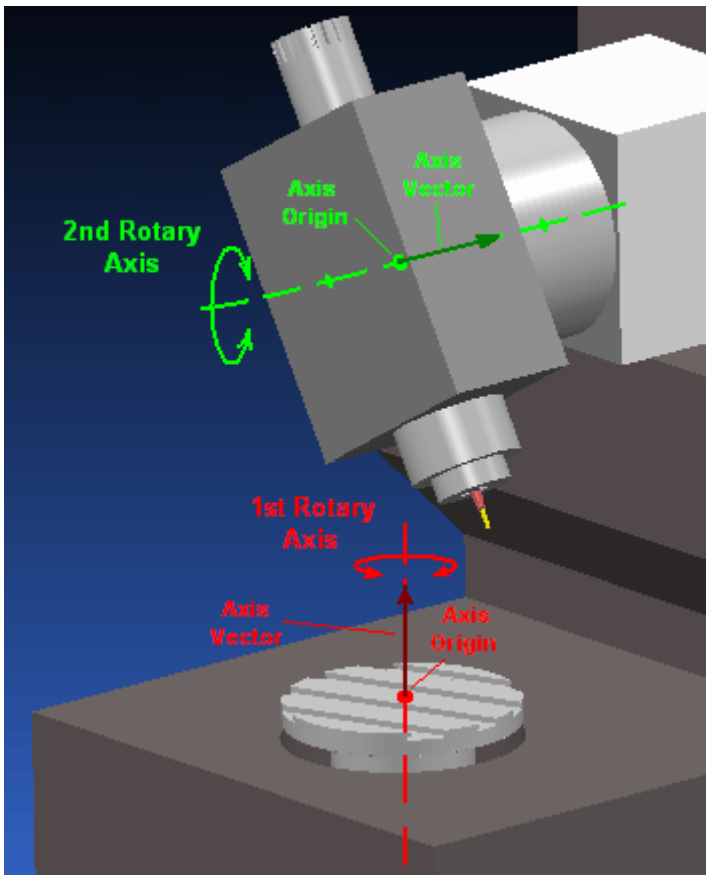
This is the simplest approach to configure the kinematics of the **Table - Table** machine. The only task is to place the part on the table so that the model workplane origin coincides with the tilted axes' intersection point in space (or place the part in a different position and compensate for this difference by configuring a machine controller).

You must conform to the following rules when configuring **Table - Table** kinematics:

- The first axis must not have a similar direction to the initial tool vector.
- The first and second axes must not have similar directions.

If these restrictions are not observed, the 5-axis kinematics degenerates to a 4- or 3-axis configuration, and **Delcam PostProcessor** outputs an error message.

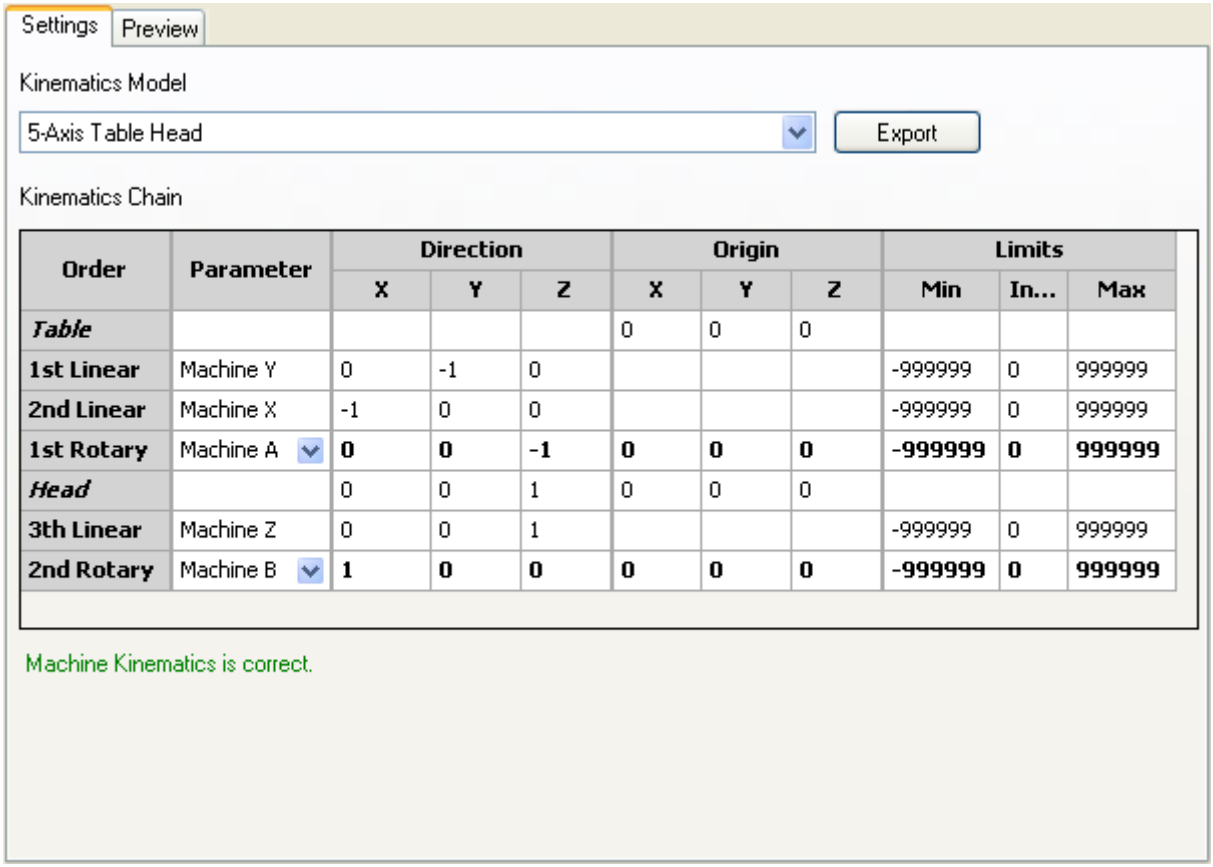
5-Axis table - head machines



Axis Vector - a typical configuration of the **Table - Head** class is shown above. The first rotary axis (C) is on the table and has a direction of 0,0,1 along the Z axis of the machine. The second rotary axis (A) is on the head and has a direction of 1,0,0 along the X axis. Neither axis changes its direction when the machine performs any other rotation.

The dull arrows mark the axis origin vectors. The second rotary axis (on the head) links to the tool holder end and the first rotary axis links to the model workplane origin. In this case, the pivot point is the origin point of the second rotary axis.

The dialog below shows typical values:

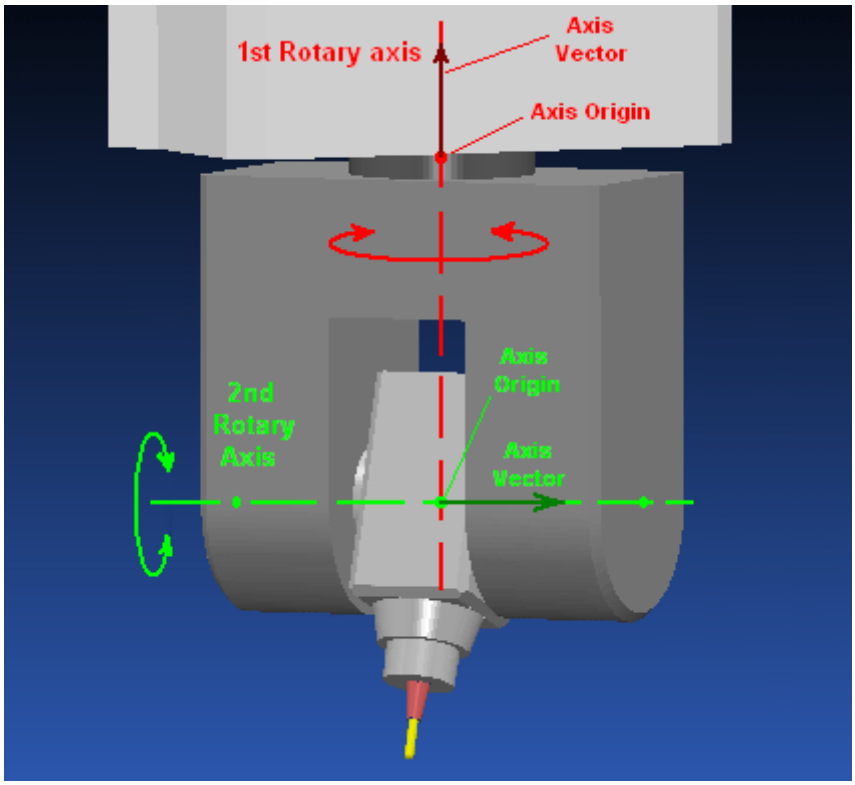


Order	Parameter	Direction			Origin			Limits		
		X	Y	Z	X	Y	Z	Min	In...	Max
Table					0	0	0			
1st Linear	Machine Y	0	-1	0				-999999	0	999999
2nd Linear	Machine X	-1	0	0				-999999	0	999999
1st Rotary	Machine A	**0**	**0**	**-1**	**0**	**0**	**0**	**-999999**	**0**	**999999**
Head		0	0	1	0	0	0			
3th Linear	Machine Z	0	0	1				-999999	0	999999
2nd Rotary	Machine B	**1**	**0**	**0**	**0**	**0**	**0**	**-999999**	**0**	**999999**

It is necessary to stick to the following rules when configuring **Head - Table** kinematics:

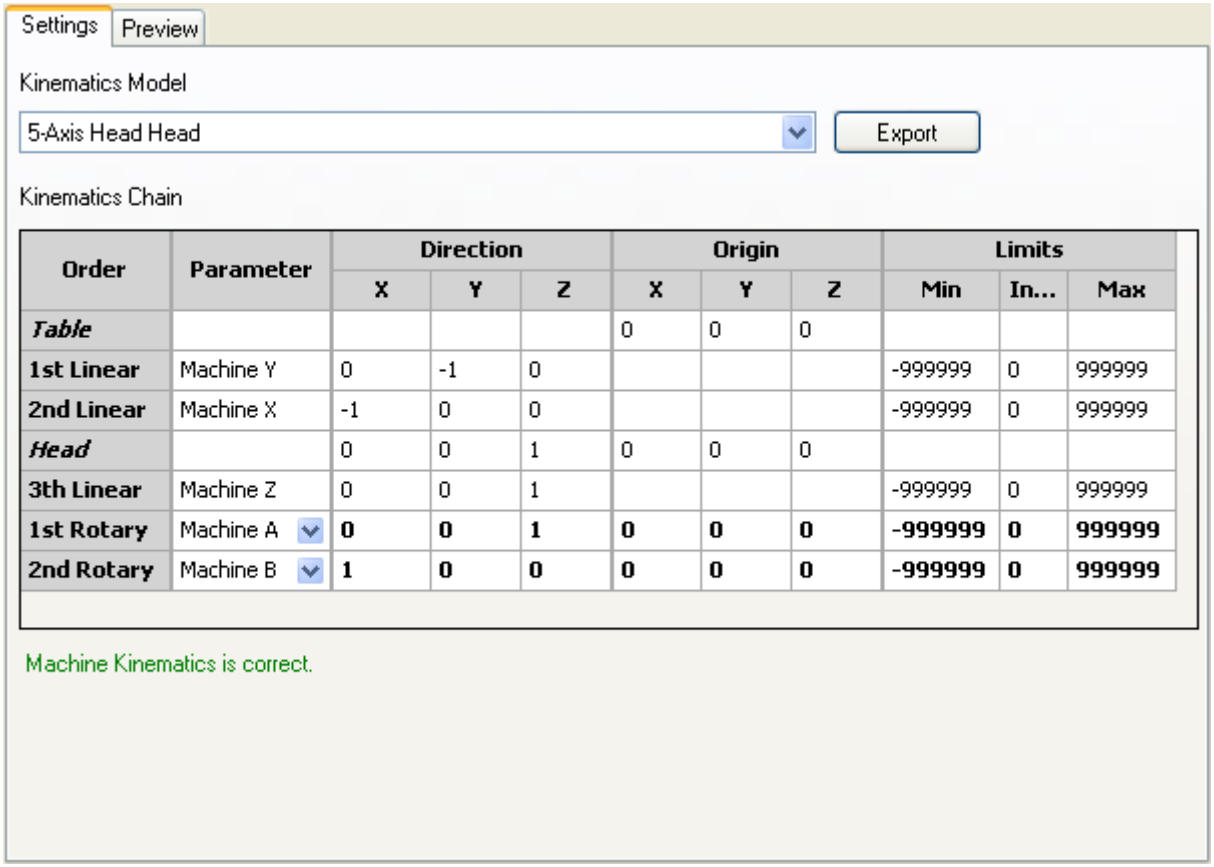
- The second axis must not have a similar direction to the initial tool vector.
- The first and second axes must not have similar directions.

5-Axis head - head machines



Axis Vector - a typical configuration of the **Head - Head** class is shown above. The first rotary axis (C) has a direction of 0,0,1. The second rotary axis (A) has a direction of 1,0,0 and rotates in the horizontal plane when the first rotary axis rotates.

The first rotary axis has its origin at a higher point than the point at which the axes intersect. This is done to demonstrate how to determine the origin points. The primary axis origin point can coincide with the intersection. In this case, the primary axis has an origin of 0,0,0. The pivot point is the origin point of the second rotary axis.



Everything needed for the pivot point calculation is on a machine head so there is no need to use a model workplane. This makes calculations simpler.

It is necessary to stick to the following rules when configuring **Head - Head** kinematics:

- The secondary axis must not have a similar direction to the initial tool vector.
- The first and second axes must not have similar directions.

MTD-based kinematics

Each machine tool for which **Delcam PostProcessor** generates programs can be described in terms of its Kinematic (mathematical) model.

This model contains information that **Delcam PostProcessor** uses to transform the model coordinates received from a CLDATA file into machine coordinates. This transformation is called the **Inverse Kinematic Problem**. Solving this problem is one of the main tasks in **Delcam PostProcessor** and is more complex when handling multi-axis trajectories.

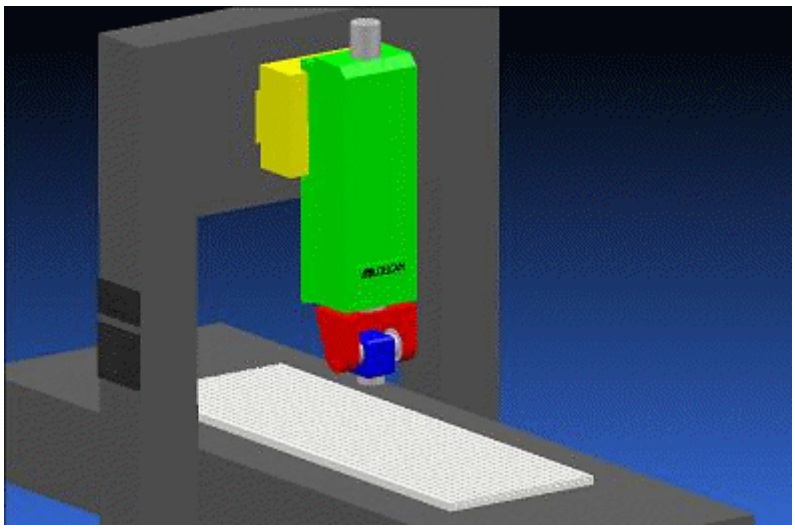
To represent machine kinematics, **Delcam PostProcessor** uses the **Machine Tool Data (MTD) format**. MTD-based kinematics offer a more effective representation than the kinematics offered by 3-axis, 4- axis and 5-axis configurations. The MTD-based approach also allows **Delcam PostProcessor** to create programs for 6-axis machines with three rotary axes.

This is shown in the following examples.

- **Example of an MTD Model**
- **Table Branch**
- **Head Branch**

Example of an MTD Model

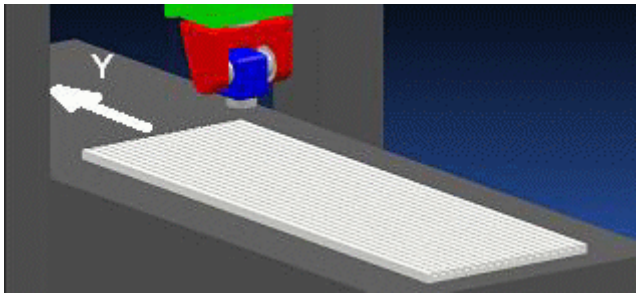
This example looks at how to create an MTD Model for a 5-Axis Head - Head machine. However, the MTD-format allows you to create models for any type of machines with any number of axes (linear or rotary) placed in any order, using similar principles.

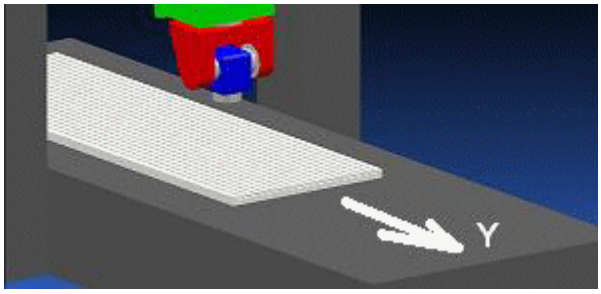


Different colours are used to represent different parts of the machine. The machine bed, which never moves, is shown in dark grey. The machine bed forms the root of the machine and branches of joints are created from this. In this case there are two branches, **Table** and **Head**.

Table Branch

Axis 1 Machine Y Linear





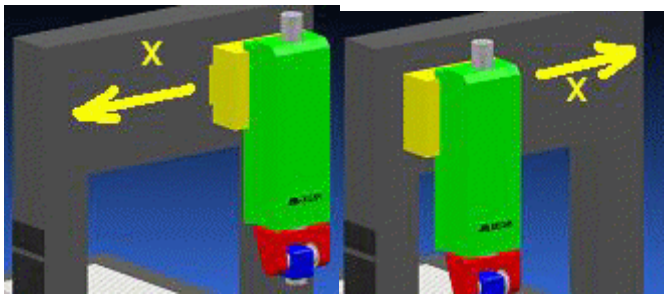
The table branch consists of just one element shown in white, the table itself. There is a joint between the table and the base. When this joint is actuated, the table moves relative to the base along the direction shown by the white arrow in the pictures. This joint is called **Y**.

> 💡 You must read the machine's manual to find out what name to use and not use some arbitrary name (it is not simply the **Y** axis of a workplane).

This axis is the closest to the part when it is fixed on the table. That's why it is **Axis 1**.

Head Branch

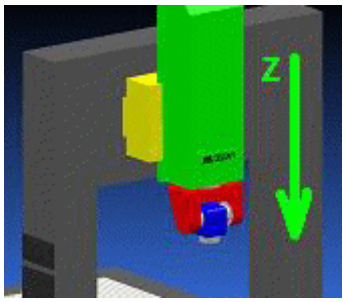
Axis 2 Machine X Linear



Start with the first joint (between the base and the yellow part) as it is closest to the base. It is **Axis 2**. You can create just one list of joints (axes) despite having two branches, because splitting the kinematics into the branches used in MTD format is only for presentation purposes. **Delcam PostProcessor Editor** displays both the axis location (head or table) and the axis when the MTD-model is loaded.

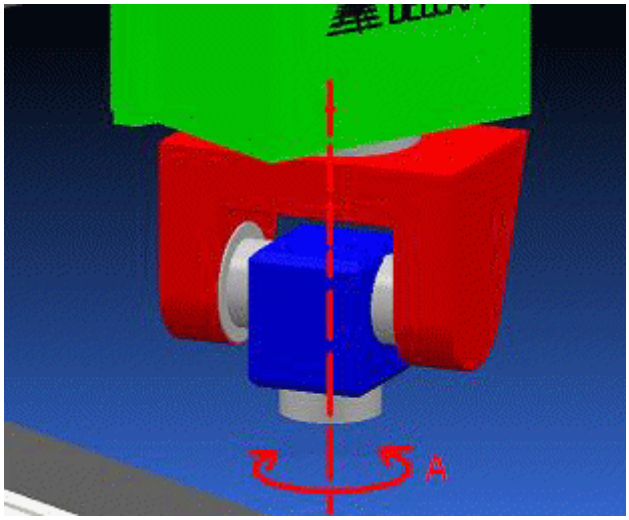
So, there is a linear joint here representing the **Machine X Axis**. When this joint is actuated, the whole head moves in the direction of the yellow arrows.

Axis 3 Machine Z Linear



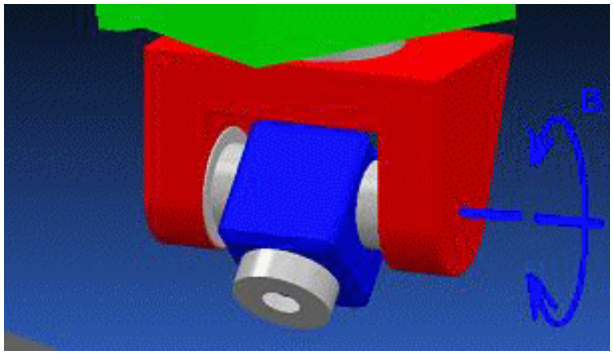
The next joint is between the yellow part and the green one. This is the last linear joint. It represents the **Machine Z Axis**.

Axis 4 Machine A Rotary



This joint connects the green part with the red part and produces horizontal rotations on the rest of the head (the red and the blue parts). **Delcam PostProcessor** uses the **Primary Rotary Angle** parameter to store the coordinate of this axis as this is first rotary axis. The axis is called **A**, and so the prefix **A** for the **Primary Rotary Angle** parameter must be added to the option file for this machine.

It represents the **Machine A Axis**.



The last joint rotates the blue part relative to the red one. The head rotation is ignored as it doesn't produce any effect when **Delcam PostProcessor** calculates machine coordinates. In this model the last joint is a second rotary axis. **Delcam PostProcessor** uses the **Secondary Rotary Angle** parameter to store the coordinate of this axis. The axis is called **B**, so the prefix **B** for the **Secondary Rotary Angle** parameter must be added to the option file for this machine.

It represents the **Machine B Axis**.

> 💡 This kinematic example has two rotary axes. But **Delcam PostProcessor** also supports 6-axis models which contain three rotary and three linear axes. For 6-axis kinematics a **Third Rotary Axis** parameter is used to store coordinate of the third rotary axis in the list (the rotary axis which is closest to the head).

Format properties

The section controls main options of output format for NC programs:

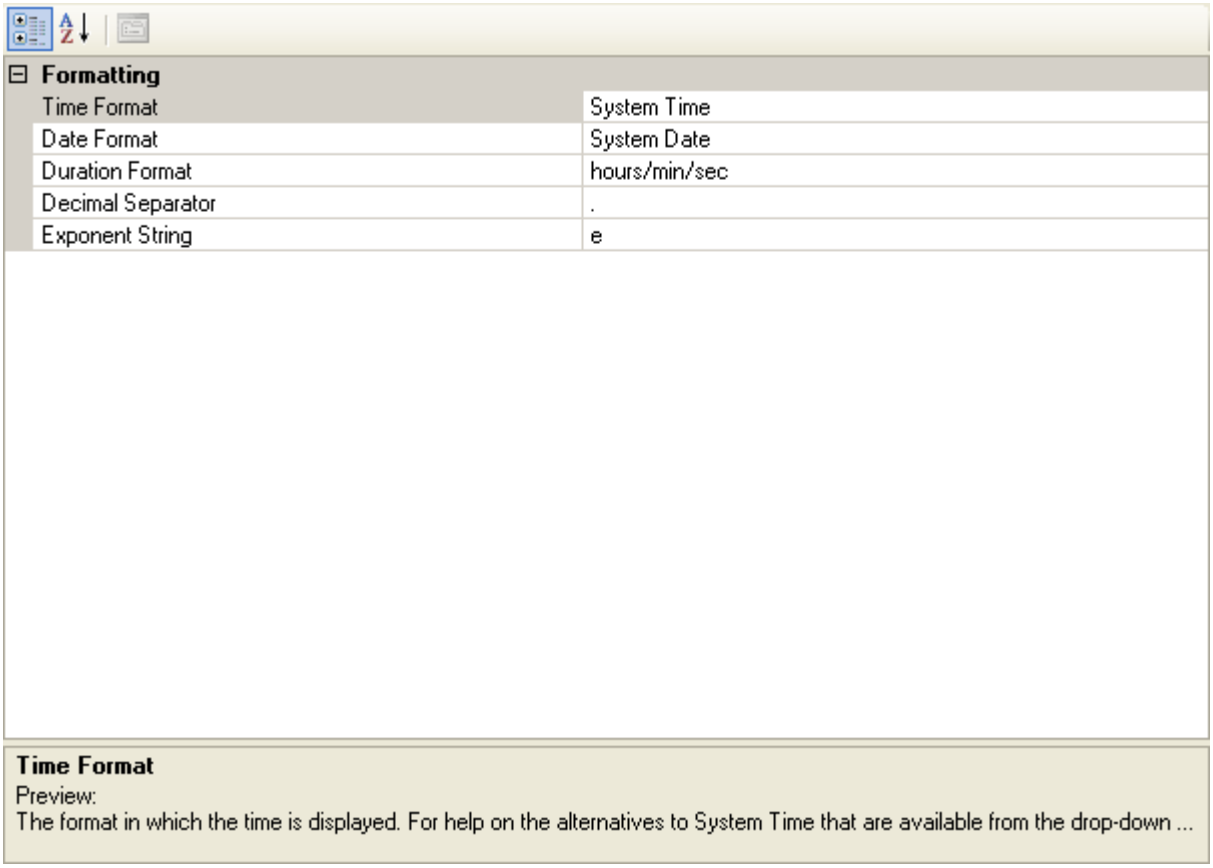
- **General**
- **Blocks**
- **NC Comments**
- **Substitutions**
- **String Variables**
- **Format - General**
- **Date and Time Format**
- **Format - Blocks**
- **Format - NC Comments**
- **Substitutions**
- **String Variables**

Format - General

The **General** page specifies the general settings for the options file.

To open the **General** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.
2. In the **Option File Properties** dialog, select **Format - General**.



Parameter	Description
Time Format	Select an entry in the list or type manually the format in which the time is displayed. See more about Date and Time Format.
Date Format	Select an entry in the list or type manually the format in which the date is displayed. See more about Date and Time Format.
Duration Format	Select an entry in the list to specify the format in which the times taken to perform tasks are displayed. An example of how durations are shown is displayed in the adjacent column.
Decimal Separator	Enter the character you want to use to separate the integer and fractional parts of numbers.
Exponent String	Enter the characters you want to use as the exponent indicator in exponential formats.

Date and Time Format

Date

M	month number without leading zeros
MM	month number with leading zeros
MMM	shortened month name
MMMM	full month name
d	day of month without leading zeros
dd	day of month with leading zeros
ddd	shortened day of week name
dddd	full day of week name
y	year of century
yy	year of century with leading zeros
yyyy	year

Time

h	hours in 12-hour format
hh	hours in 12-hour format with leading zeros
H	hours in 24-hour format
HH	hours in 24-hour format with leading zeros
m	minutes
mm	minutes with leading zeros
s	seconds
ss	seconds with leading zeros
t	AM or PM

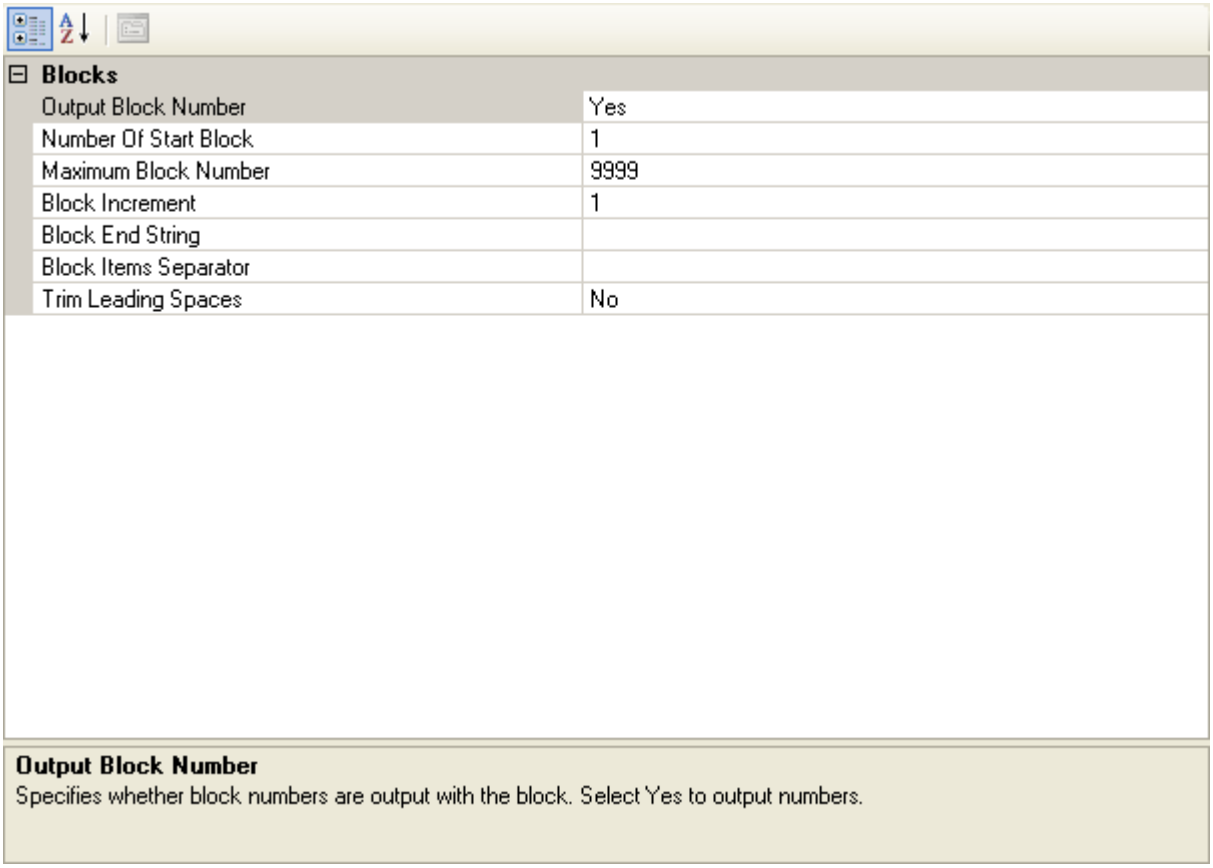
In order to output any other characters, use single quotes (').

Format - Blocks

The **Blocks** page specifies the format of the blocks output to NC programs.

To open the **Blocks** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.
2. In the **Option File Properties** dialog, select **Format - Blocks**.



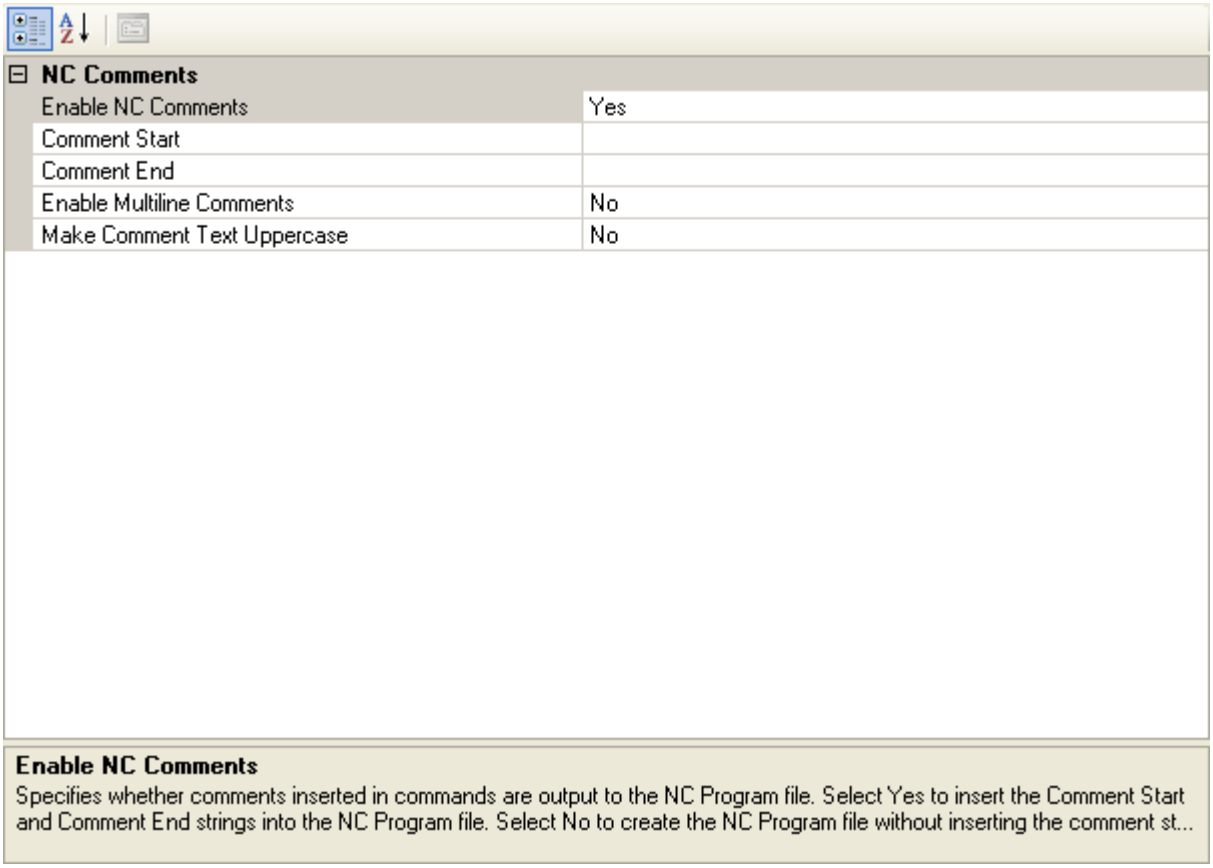
Parameter	Description
Output Block Number	Specifies whether block numbers are output with the block. Select **Yes** to output numbers.
Number Of Start Block	Specifies the number of the first block in the program.
Maximum Block Number	Specifies the maximum block number that can be used in the NC program. If Delcam PostProcessor reaches this number, numbering restarts.
Block Increment	Specifies the number by which the block numbers are incremented.
Block End String	Specifies the string to mark the end of a line in the program. If a string is set, each line of program must end with this string.
Block Items Separator	Specifies the string to be inserted between block items.
Trim Leading Spaces	Select **Yes** to remove leading spaces from the NC program file. Select **No** to preserve leading spaces.

Format - NC Comments

The NC Comments part defines format for output of NC comments to programs.

To open the **NC Comments** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.
2. In the **Option File Properties** dialog, select **Format - NC Comments**.



Parameter	Description
Enable NC Comments	Specifies whether comments inserted in commands are output to the NC Program file. Select Yes to insert the **Comment Start** and **Comment End** strings into the NC Program file. Select **No** to create the NC Program file without inserting the comment strings.
Comment Start	Enter the character string used to identify the start of a comment. Different machine controllers use different characters to identify the start of a comment. You must enter the appropriate character for the target controller.
Comment End	Enter the character string used to identify the end of a comment. Different machine controllers use different characters to identify the end of a comment. Others, such as Siemens controllers do not require an end comment character. You must enter the appropriate character for the target controller.

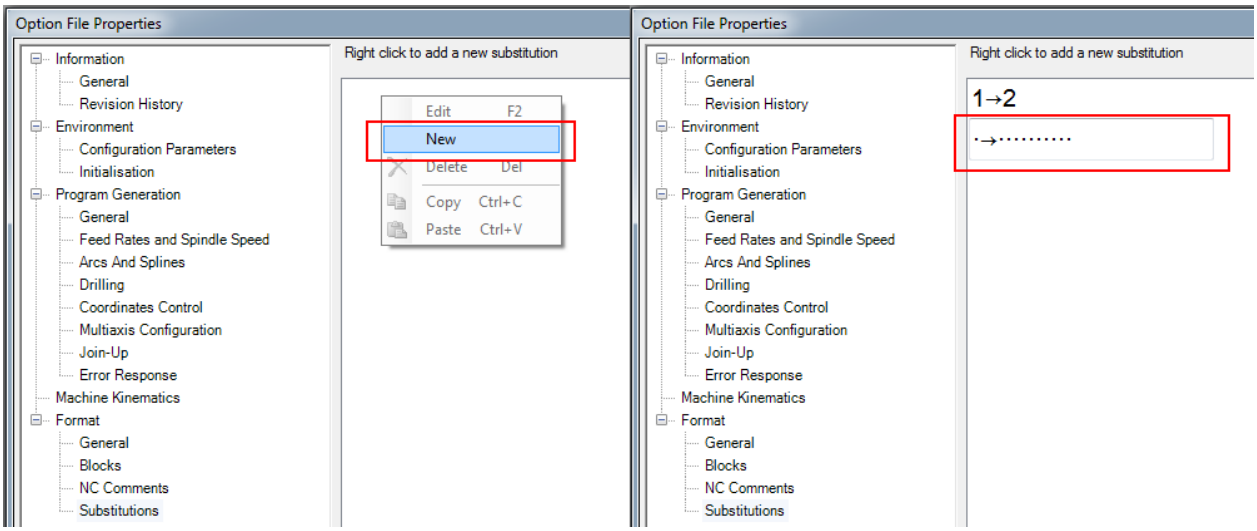
Parameter	Description
Enable Multiline Comments	Specifies how multi-line comments are written to the NC Program file. Select Yes to add the **Comment Start** and **Comment End** strings only at the start and end of the comment. Select **No** to add the **Comment Start** and **Comment End** strings to each line of the comment.
Make Comment Text Uppercase	Some machine controllers require comment text to be in uppercase. Select **Yes** to convert all comment text to uppercase. Select **No** to preserve the capitalization of the comment text.

Substitutions

Some machine controllers cannot recognize or cannot accept all characters that can appear in an NC program file. For example, Fanuc controllers ignore lower case characters in comments, and Haas controllers do not accept the underscore character. To avoid this problem, you can use the **Substitutions** page to automatically replace *all* instances of the specified characters in the NC Program file.

To open the **Substitutions** page:

1. Click the **Option File Properties** button, or select the **File > Option File Properties** menu option.
2. In the **Option File Properties** dialog, select **Format - Substitutions**.



To automatically replace one character with another:

1. Type the character you want to replace.
2. Type the replacement string.

To automatically remove a character from the output file:

1. Type the character you want to remove.
2. Leave the left part empty.

To automatically remove all non-English characters from the NC Program file, select the **Remove International Characters** check box.

String Variables

A string variable is a special string entity that replaced by its value in the end of processing. A string variable can be placed at any text field in the option file with the following syntax: "%s(name)%", where "name" is the identifier of the string variable. String variables are defined in a special table of the Option File Properties dialog.

A string variable can also be set in the script with the SetStringVariable() function:

SetStringVariable("My Variable", "1234.56789");

Note, that string variables are applied at very end of the processing. Therefore, only last set value of the string variable is applied.

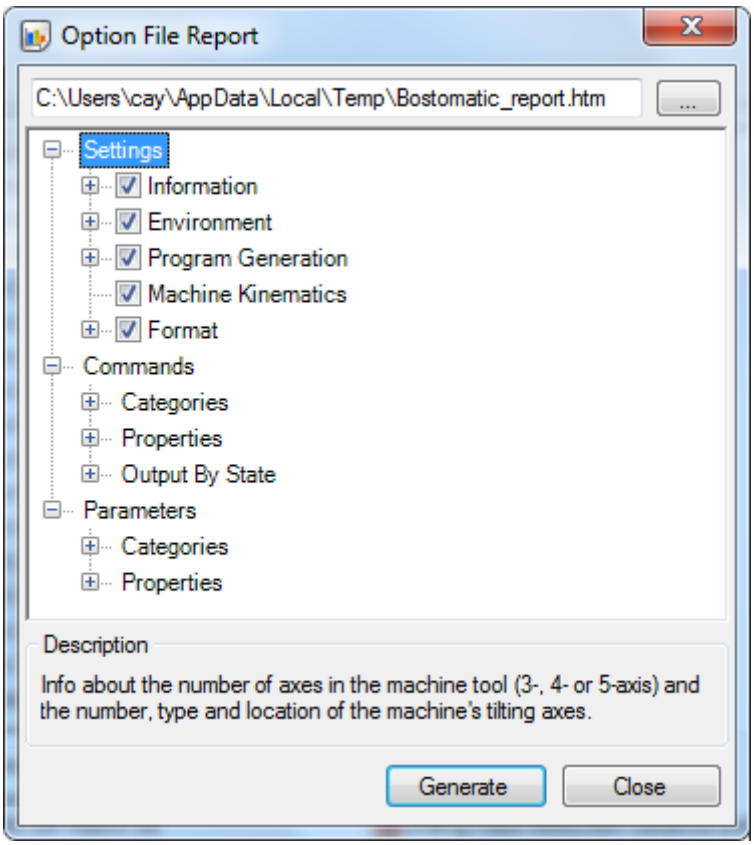
The string variables are useful where you need to customize multiple strings in one go. There is no limitation on where a string placeholder is output from: prefix or postfix of a parameter, text item in a block or even from CLDATA.

Generating an option file report

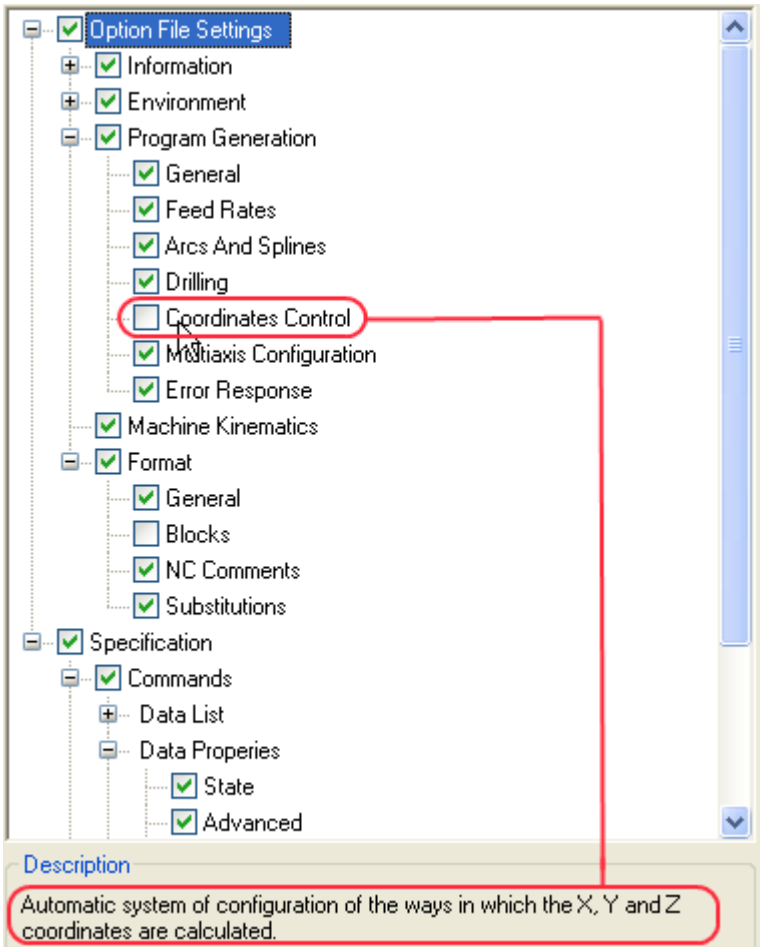
You can generate an HTML report that lists the file's history, its general and initialisation settings, and the commands and parameters used in the option file.

To generate a report:

1. Save your edits to the option file.
2. Select the **Tools > Option File Report** menu option or click on the toolbar.
3. In the **Option File Report** dialog, enter the location where you want to save the HTML report, or click ... to select a path for the report.



4. By default, the report includes all the elements of the option file. If you do not want include all the elements in the report, deselect the check boxes of the elements you want to exclude.



5. Deselect the **Data properties** you want to exclude from the command table:

State displays the state (active, inactive, disallowed) of each command.

Advanced displays any linked script functions for each command.

Preview displays a command preview for each command.

6. Deselect the **States** you want to exclude from the report:

Active outputs active commands in the commands table.

Inactive outputs inactive commands in the commands table.

Disallowed outputs disallowed commands in the commands table.

7. Click **Generate** to create the report and display it in your default browser.

Flow of information relating to the output of a parameter's value

A parameter has an internal **Updated** flag that indicates whether its value has changed since it was last output. A flag setting of **Yes** means that the value has changed and that the value is to be output where the parameter is used modally. A flag setting of **No** means that the value is unchanged and that the value is therefore not to be output where the parameter is used modally.

💡 A parameter is used modally when its **Output to Tape** flag is set to **If Updated** in the format or command.

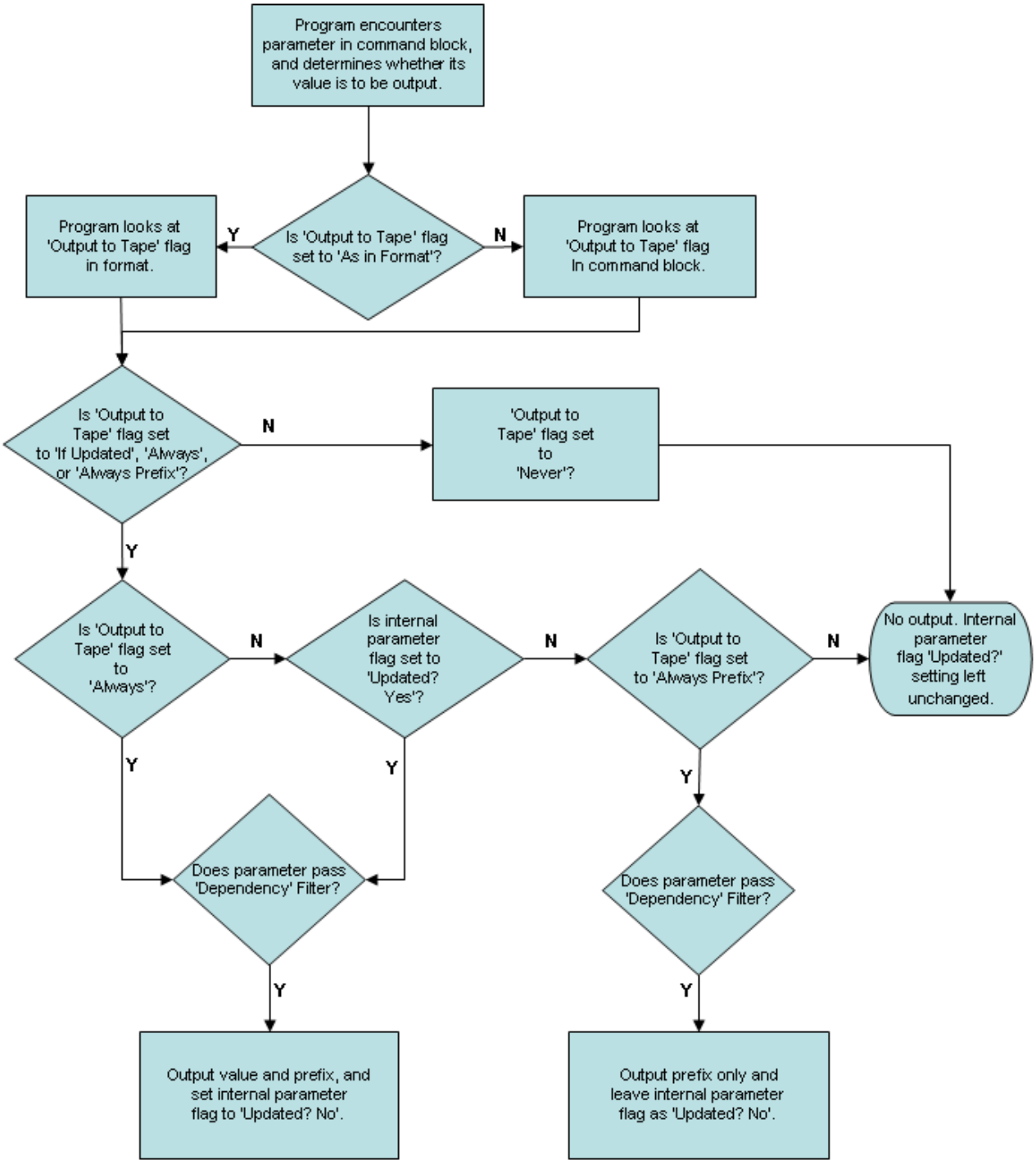
The setting of the **Updated** flag is affected by the following events:

Event	Effect
Manual setting of the **Updated** flag in the Initialisation Table.	Setting the **Updated** flag to **Yes** causes **Delcam PostProcessor** to treat the parameter's initial value as having changed (even if it has not).
A different value is assigned to the parameter using a format or command, or by CLDATA. Subsequent changes to the value do not affect the flag; the setting only changes (to **No**) when the current value is output.	Sets **Updated** to **Yes** if it was previously set to **No**.
The Output Next Time flag is set to **Forced**.	Sets **Updated** to **Yes** if it was previously set to **No**.

Event	Effect
The Output Next Time flag is set to **Suppressed**.	Sets **Updated** to **No** if it was previously set to **Yes**.
Output of parameter's value.	Sets **Updated** to **No** if it was previously set to **Yes**.

The following flowchart shows the sequence of events when the postprocessor encounters a parameter in a command block. It demonstrates how the setting of the internal **Updated** flag is important when a parameter is used modally.

For the Dependency Filter, refer to the Command Interface.


Program encounters parameter in command block, and determines whether its value is to be output.
Is 'Output to Tape' flag set to 'As in Format'?
Y
N
Program looks at 'Output to Tape' flag in format.
Program looks at 'Output to Tape' flag In command block.
Is 'Output to Tape' flag set to 'If Updated', 'Always', or 'Always Prefix'?
N
'Output to Tape' flag set to 'Never'?
Y
Is 'Output to Tape' flag set to 'Always'?
N
Is internal parameter flag set to 'Updated? Yes'?
N
Is 'Output to Tape' flag set to 'Always Prefix'?
N
No output. Internal parameter flag 'Updated?' setting left unchanged.
Y
Y
Y
Does parameter pass 'Dependency' Filter?
Does parameter pass 'Dependency' Filter?
Y
Y
Output value and prefix, and set internal parameter flag to 'Updated? No'.
Output prefix only and leave internal parameter flag as 'Updated? No'.

Commands

There are two ways in which commands can be generated:

- From the corresponding tags in CLDATA.
- From within the **Postprocessor**. For example, one of the commands **Circular Move YZ**, **Circular Move XZ** or **Circular Move XY** is typically generated by a plane where an arc lies.

As a minimum, the following commands should be activated and specified in each option file:

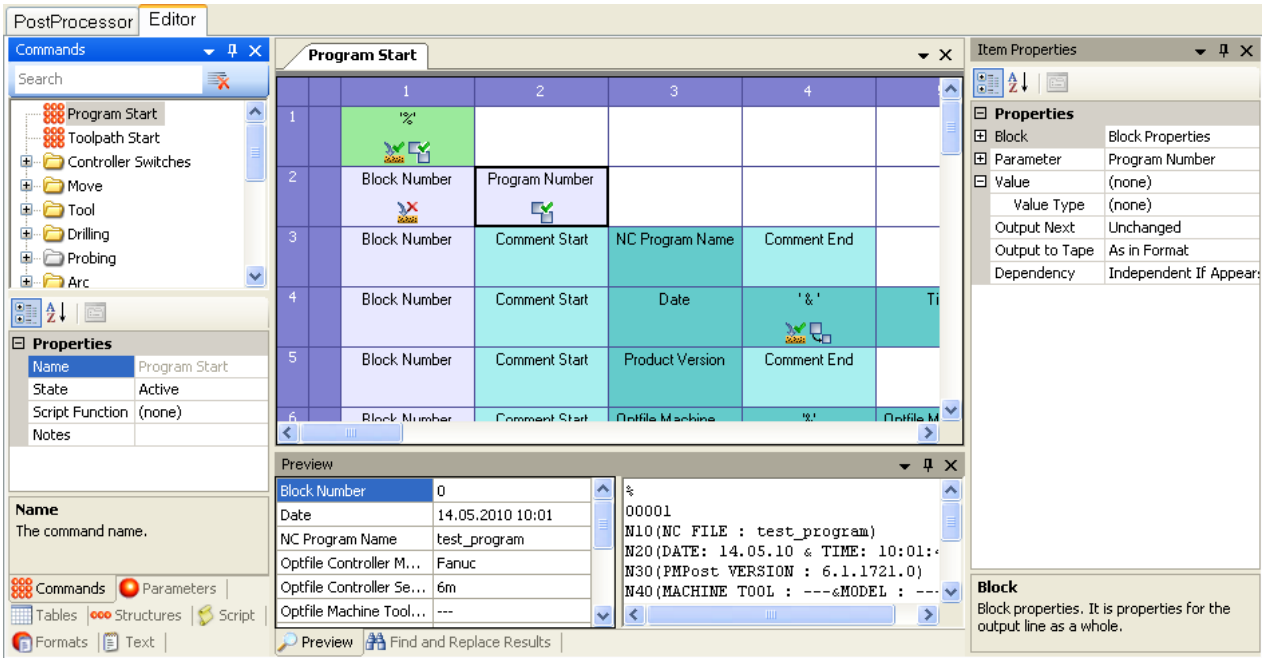
- **Program Start**
- **Load Tool First**
- **Load Tool**
- **Move Rapid**
- **Move Linear**
- **Circular Move YZ**, if the **Arcs** option is selected for Arcs and Splines.
- **Circular Move XZ**, if the **Arcs** option is selected for Arcs and Splines.
- **Circular Move XY**, if the **Arcs** option is selected for Arcs and Splines.
- **Spline Move**, if **Spline Configuration Output** is set to **Polynomial**.
- **Program End**
- To activate a command, right-click its entry in **Explorer** and select **Activate** from the context menu.
- By default, commands are shown according to option file properties. For example, if arcs are disabled in option file properties arc related commands become hidden.

There are two types of command:

- Generic commands are always present in the **Postprocessor** and are invoked during processing when required.
- User commands are required to combine some parts of the output. They are not invoked by default, but you can use them in scripts and nest them in other commands.

A command has two layers of postprocessing:

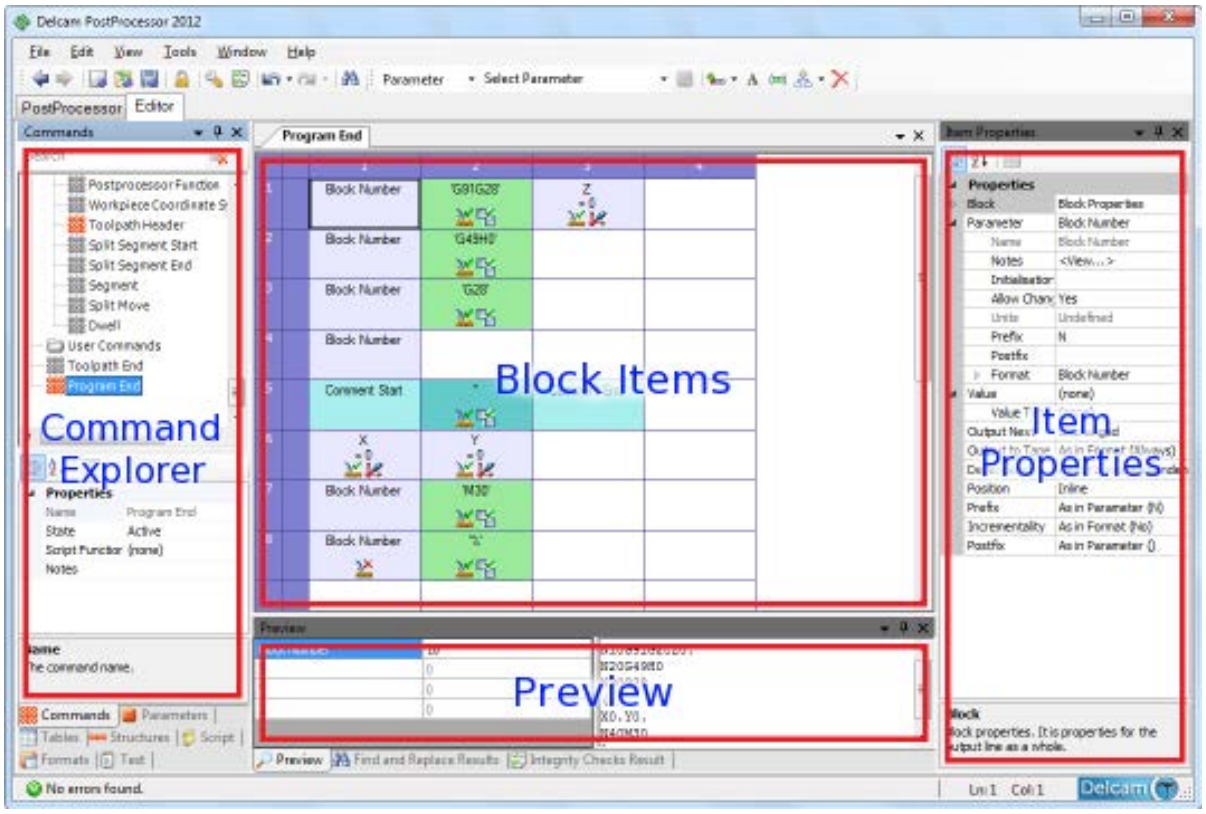
- Standard processing parses the block items sequentially.
- **Scripted** processing invokes the script selected in the item properties. This script can invoke the block items assigned to the command using the **StandardResponse()** function, or bypass the block items, using the **AdvancedResponse()** function, and call the advanced responses specified in the script.



Command interface

The command block is the NC program's basic component. A block consists of primary elements which are output, such as other nested commands, parameters and text fragments.

A command is made up of one or more blocks, with each block containing **Block Items**. A block represents an NC program string. Typically, a block begins with a **Block Number**, but it can start with any element which has a carriage return associated with it (for example, any command).



Command Explorer - displays list of commands and properties for selected command:

- **Name** - the name of selected command.
- **State** - current state of selected command (**Active**, **Inactive** or **Disallowed**).
- **Script Function** - displays the user-defined script function linked to selected command.
- **Notes** - allows to enter your own notes about the command.

Block Items - displays list of items for output to NC program.

Preview - shows preview of selected command.

Block Item Properties - displays properties of selected item in **Block Items**.

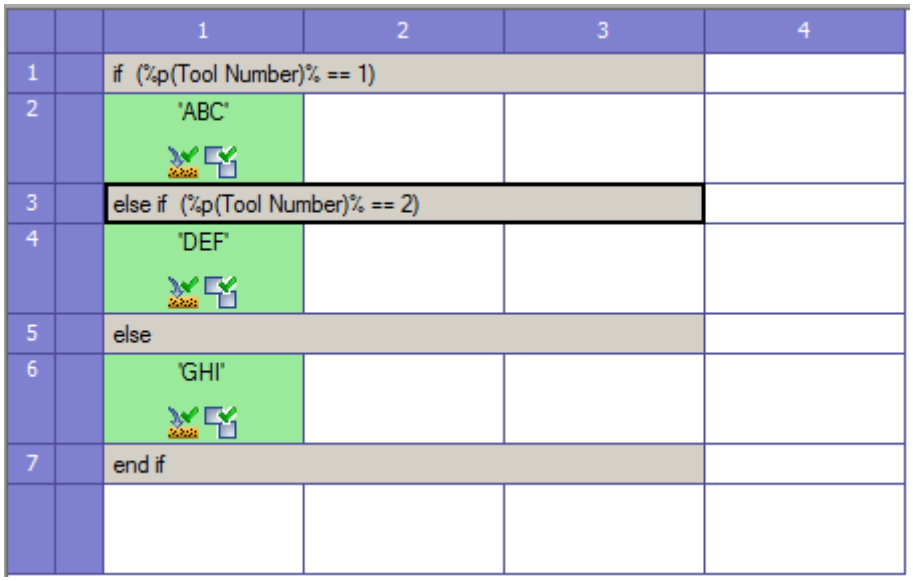
Command toolbar

The toolbar allows to edit the contents of selected command.

Parameter	Drop-down list of the block item types you want to add, which contains: **Parameter** **Command** **Structure** **Table**
Select Parameter	Contains the items of the specified type that can be added as block items.
	Adds selected item into the block.
	Drop-down button, which contains:  - Insert a new block with **Block Number** at begin.  - Inserts a new block empty block.
	Inserts a text block item. To edit, see Block Item Properties
	Comments selected block items. To set up comments style, see Option File Properties: Format - NC Comments.
	Drop-down button, which adds boolean condition item.
	Removes the selected block items.

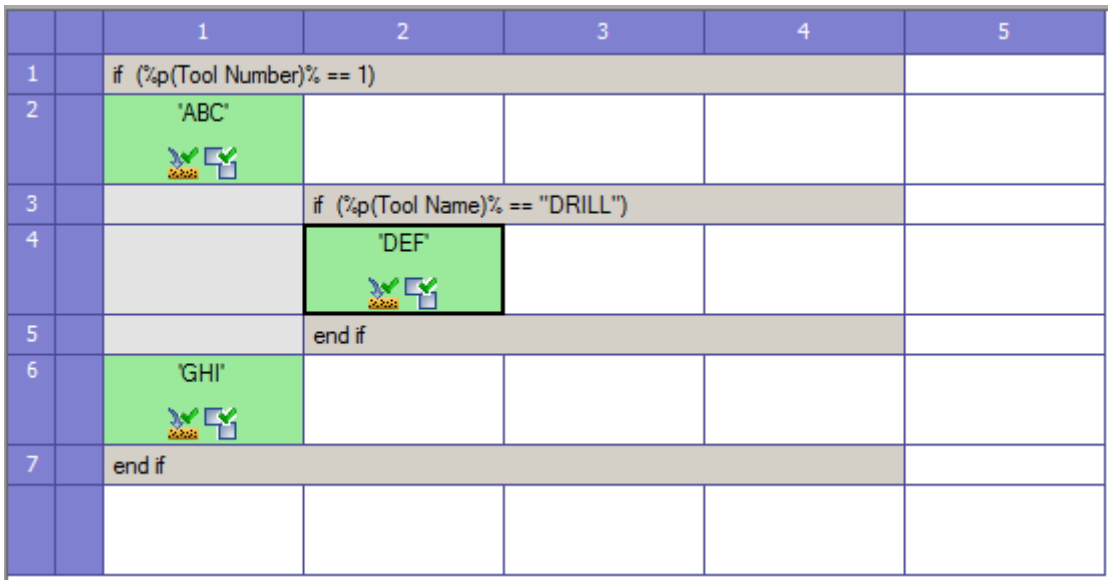
Conditional Statements

Command interface allows inserting conditional statements in command block. A conditional statement is associated with an expression which is evaluated in run-time. If the expression is true the blocks under the statement are executed and output.



There are four items available for building conditional statements: if, else if, else, end if.

Conditional statements can be nested one in another. The depth of the nesting is not limited. By default, nested conditional statements are shown with indentation. This mode can be toggled by button on the toolbar.



The expression syntax should correspond to the scripting language of the option file. For example, %p(Tool Number)% != 1 Is valid for Jscript, whereas the same expression for VBScript is %p(Tool Number)% <> 1

A string like %p(Tool Number)% is just abbreviated form of GetParam("%p(Tool Number)%")

The expressions of conditional statements are evaluated in the same context as the main script. Therefore, there could be used variables and functions defined.

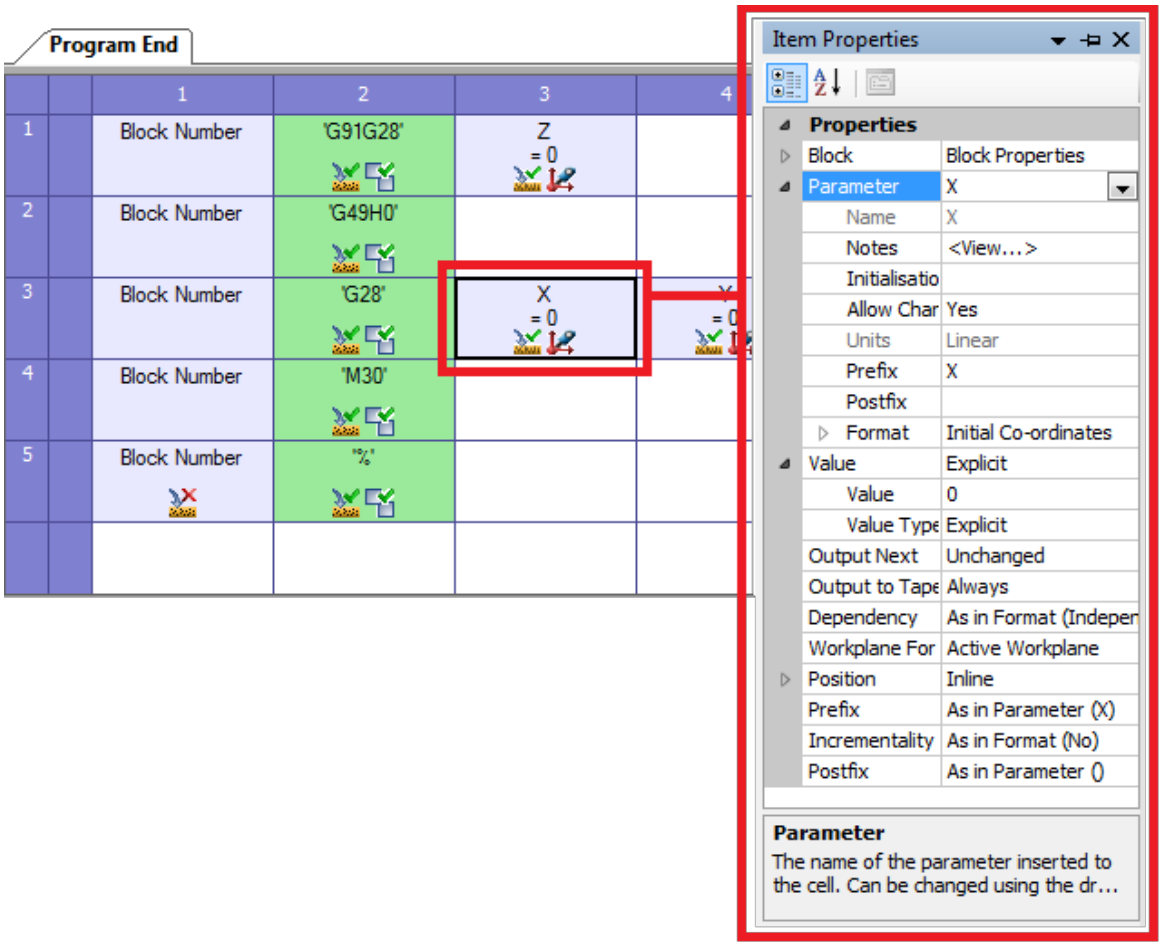
		1	2	3	4	5
1		if (myGlobalVariable==777)				
2		'ABC'				
3			if (MyScriptFunction("arg1"))			
4			'DEF'			
5			end if			
6		'GHI'				
7		end if				

In order to insert a conditional statement do the following:

- Press drop-down button and choose the statement you need
- Select the statement inserted and open its properties
- In the Item Properties change the expression

Block Item Properties

Details of the selected block item are shown in the **Item Properties** table. For example:



Block - displays properties of line where a block item is selected:

- **Description** - allows to add own description for the line.
- **Disabled** - to treat the selected block as a comment and prevent **Delcam PostProcessor** from creating any code for it. The block is greyed to indicate it is disabled.
- **Long Block** - to link blocks together.

Parameter/Command/Table/Structure - element properties for different types of inserted item:

- **Parameter**
- **Command**
- **Structure**
- **Table**

Value - the type of entry is determined by the entries for **Block Item** and **Value Type**.

Value Type / Type - this field determines the kind of output. For a command, it is set to **Output Block Items Only** (when **Delcam PostProcessor** generates standard NC output for the command block) or **Output Script Functions**. Otherwise, you can select an option from the drop-down list:

- **Expression** - assigns the result of an expression to the parameter. The expression can contain any built-in and user parameters, operators, trigonometric functions and so on.
- **Function** - assigns the result of a script function to the parameter.
- **Group** - select the required setting from the drop-down list in the **Value** field.
- **None** - no output.
- **Parameter** - select the required parameter from the drop-down list in the **Value** field.
- **Explicit** - enter the required value in the **Value** field. Input text string to the **Value** field in the case of a **Text** block item.
- The options listed depend on the type of the parameter.

Output Next - if the **Output To Tape** setting of the parameter is **If Updated** or **Always Prefix** (see above), this forces or suppresses the appearance of the value next time:

- **Unchanged** ignores **Output Next**
- **Forced** always writes out the value next time regardless of other settings
- **Suppressed** always stops the value from being written out next time regardless of other settings

Workplane For Output - specifies workplane transformation used for coordinate parameters. Select:

- **Active Workplane** (default) to output the coordinates according to the active coordinate system (this depends on the current setting of **Workplane Transformation** - see Parameter Control).
- **CLDATA Output Workplane** to output the coordinates of the block item in the workplane from the program that generated the CLDATA file (for example, the **PowerMILL** workplane).

- **Local Workplane** to output coordinates according to local workplane of current toolpath.
- **Machine Workplane** to output coordinates according to machine base workplane.
- **CAD Workplane** to output coordinates according to global workplane of CAD program.

Position - controls location where item will be outputted. Select:

- **Inline** - use default items order, one by one.
- **Absolute Position** - print value at absolute position from begin of line.
- **Insert Spaces** - insert number of spaces before current item.
- Using too small values in **Absolute Position** mode may overwrites previous outputted items.

Incrementality - controls numeric value which will be outputted. Select:

- **As in Format** - use format defined setting.
- **Yes** - print values relative to the previous (outputted) value.
- **No** - print actual value.

Prefix - controls prefix of parameter which will be outputted. Select:

- **As in Format** - use parameter defined setting.
- **Specify** - use overrided prefix value.

Postfix - controls postfix of parameter which will be outputted. Select:

- **As in Format** - use parameter defined setting.
- **Specify** - use overrided prefix value.

Using Output Next

Output Next block item flag is used as an addition to **Output to Tape** flag to control output of **modal** parameters when their initial output settings need to be redefined in some particular block. In difference from **Output to Tape** flag (which affects the block item directly), **Output Next** works little bit like "deferred" **Output to Tape** - it does nothing with parameter output when it is used but snaps into action when the parameter appears in blocks next time during processing.

A modal parameter is one for which **Output To Tape** is set to **If Updated** in its format or command.

Example of Output Next = Suppressed

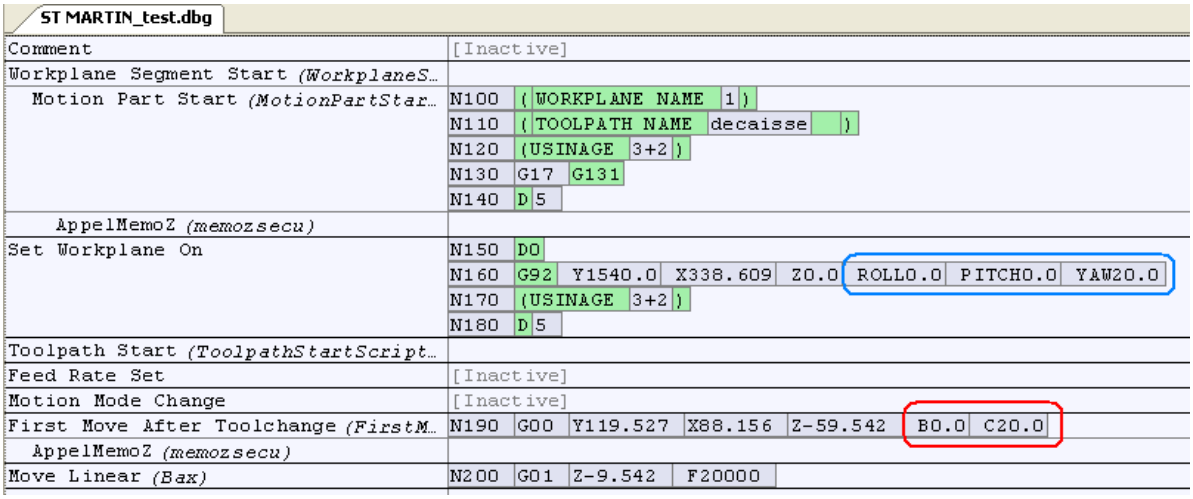
This state may be useful if output of some parameter within some block in addition to modality issues indirectly depends on previously processed blocks.

Lets consider an option file which supports 3+2 trajectories and uses Local Workplanes. The cutfile to be processed is intended to cut a set of pockets in differently oriented planes. Such cutfile contains a set of 3+2 trajectories connected by rotary connection moves (join-ups). After processing the cutfile the result NC-

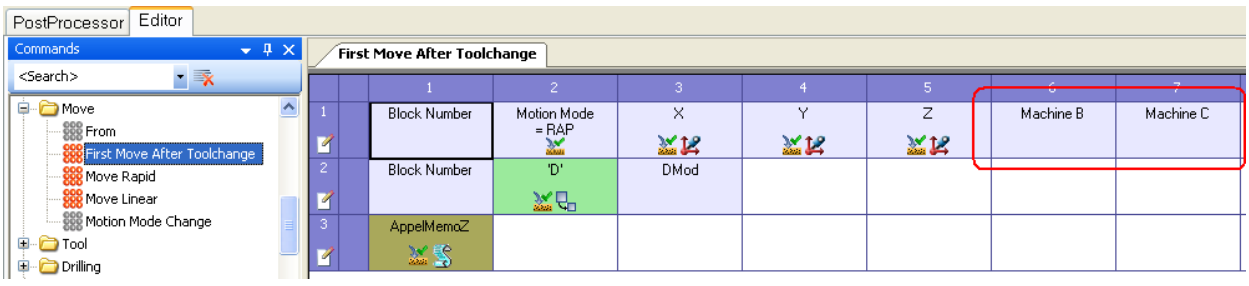
file had been generated and for each toolpath start part it contained output similar to the one shown below:

```
...
N100 (WORKPLANE NAME 1)
N110 (TOOLPATH NAME decaisse  )
N120 (USINAGE 3+2)
N130 G17 G131
N140 D5
N150 D0
N160 G92 Y1540.0 X338.609 Z0.0 ROLL0.0 PITCH0.0 YAW20.0
N170 (USINAGE 3+2)
N180 D5
N190 G00 Y119.527 X88.156 Z-59.542 B0.0 C20.0
N200 G01 Z-9.542  F20000
N210 Z0.216  F400
...
```

For better understanding what is in there we may look at the same part of the file in Postprocessor's Debug View window:

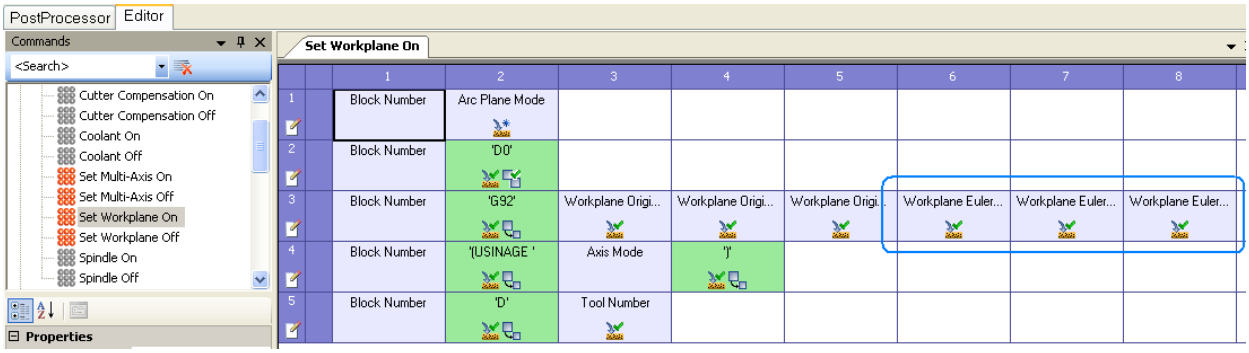


Set Workplane On command which starts the workplane section in the NC-program uses Euler Angles (Roll-Pitch-Yaw) to set Local Workplane orientation. They are encircled with the blue frame. At the same time the red frame encircles machine angles contained in **First Move After Toolchange**. And there is the problem - machine angles are not needed there as the Local Workplane section is already started. But machine angles appeared there just because this is the first time they have been met after a conntecting join-up rotation (or first time in program at all). It is necessary to remove the angles from there for 3+2 toolpaths. Lets look at the **First Move After Toolchange** block in the Editor:

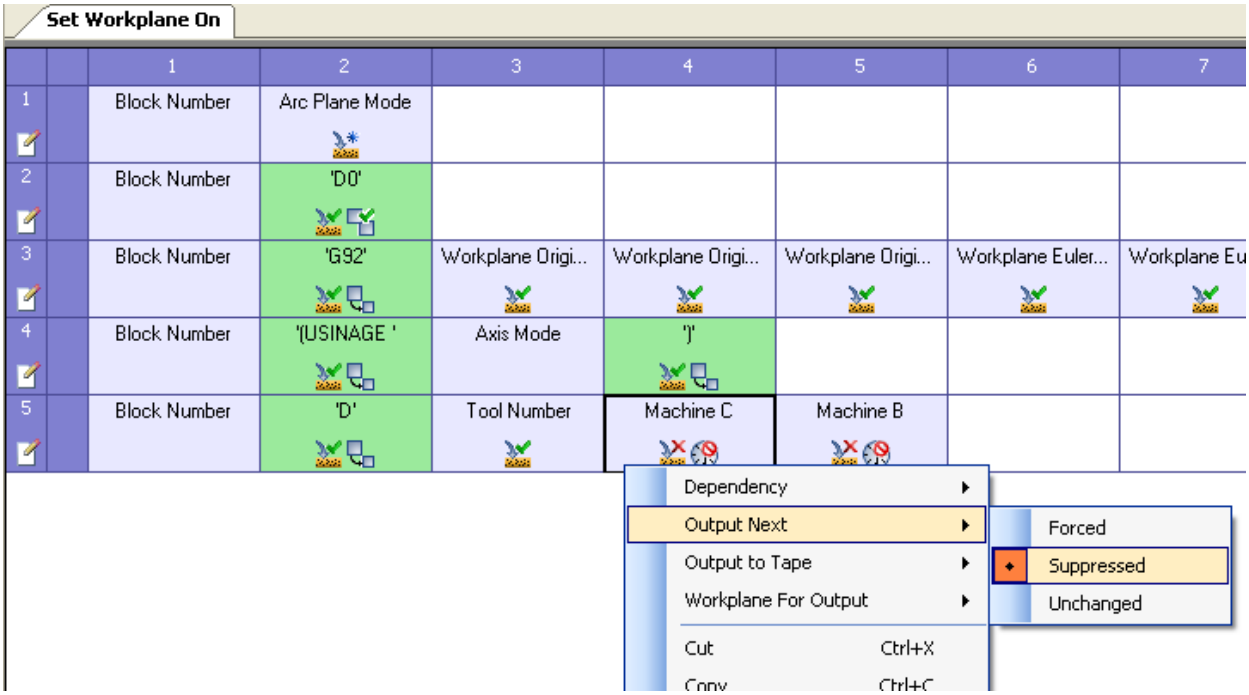


Machine angles cannot just be removed as they are still necessary for pure 5-axis toolpaths (when there is no any Local Workplane used). Instead of removing we

can suppress their output by **Output Next** flag but not in this block. Local Workplane section causes this problem, so this can be done in block **Set Workplane On**. In this case the addition we are going to make will only affect 3+2 toolpaths. There is **Set Workplane On** block:



The blue frame shows Euler angle parameters and they do not have any influence on the machine angles. It is necessary to add the machine angles in any place of the block and set **Output Next** flag to **Suppressed** state for them:



Note that this solution works together with **Output to Tape** flag set to **Never** state. This optional stuff is needed to exclude appearance of the machine angles from **Set Workplane On** block. After the block is processed the next call of the machine angles will happen in **First Move After Toolchange** and that is the moment when the **Output Next** flag snaps into action for them. Also note that this will only work if the machine angle parameters are **modal**. The result code after updating the option file will look like shown below. There are no machine angles in **First Move After Toolchange** any more:

```
...
N100 (WORKPLANE NAME 1)
N110 (TOOLPATH NAME decaisse  )
N120 (USINAGE 3+2)
N130 G17 G131
N140 D5
N150 D0
```

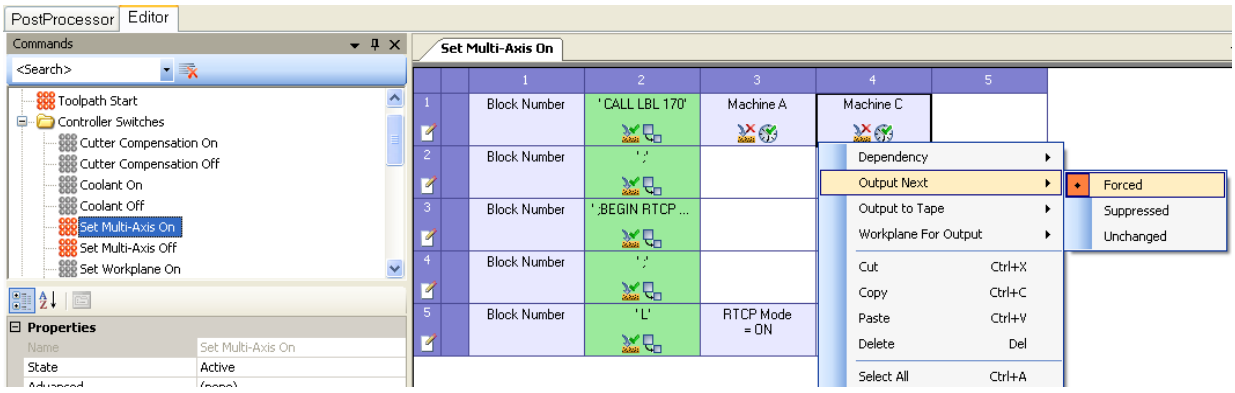
```
N160 G92 Y1540.0 X338.609 Z0.0 ROLL0.0 PITCH0.0 YAW20.0
N170 (USINAGE 3+2)
N180 D5
N190 G00 Y119.527 X88.156 Z-59.542
N200 G01 Z-9.542  F20000
N210 Z0.216  F400
...
```

Example of Output Next = Forced

This option may be used if it is needed to repeat output of a modal parameter later even if it will not change its value. Also it may be needed to change some parameter without writing it down but ensure its output when it is met next time during processing (deferred output). As an example we'll consider switching to multi-axis mode. Look at the following piece of code:

```
...
86 L X-38.507 Y-90.869 Z-17.984 B-54.145 C109.871
87 L X-40.0 Y-97.608 Z-21.874 B-57.853 C111.344
88 ;
89 ;END RTCP SECTION
90 ;
91 L M129
...
...
146 CALL LBL 170
147 ;
148 ;BEGIN RTCP SECTION
149 ;
150 L M128
151 L X-28.0 Y-74.627 Z-2.591 B-57.853 C111.344
152 L X-29.291 Y-78.524 Z-8.157 B-62.63 C113.277
...
```

There are multi-axis switching OFF/ON procedures. We may want to have machine angles written down after multi-axis is turned ON even if they will be repeated. But basically machine angles are **modal**. Blue frames at the picture above encircle repeating angles. To do that it is enough to open **Set Multi-Axis On** block in the editor and set flags as shown below:



Doing this we ensure machine angles to appear within the first motion block coming after **Set Multi-Axis On** whether they changed or not. As in previous example, **Output to Tape** flag is also set to **Never** as we do not need machine angles to appear in **Set Multi-Axis On** (deferred output).

Compound commands

Compound commands are commands that contain one or more nested commands (also called sub-commands). A nested command is parsed normally and the result of parsing is inserted in the output according to the nested command's place in the host command block.

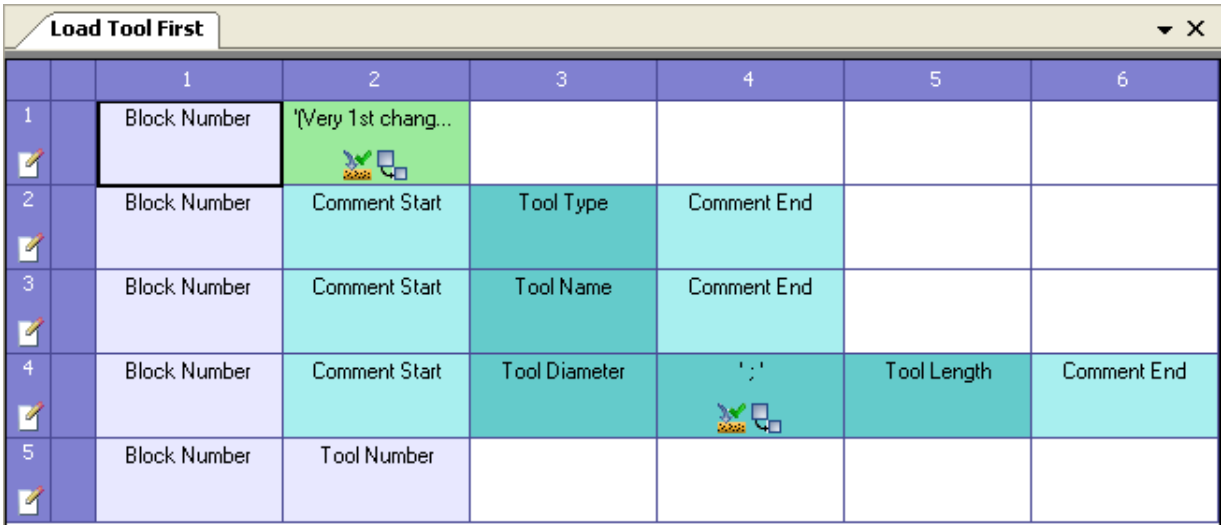
> Nested commands can be generic or user defined. They are subject to the **Always** and **Never** settings of the **Output to Tape** flag - refer to the **Command Interface**.

A compound command can be invoked by a **StandardResponse()** command. A nested command may itself have a script function assigned to it. In this case, the function is executed only when the **Type** of the nested command is set to **Output Script Function** (by default, it is set to **Output Block Items Only**).

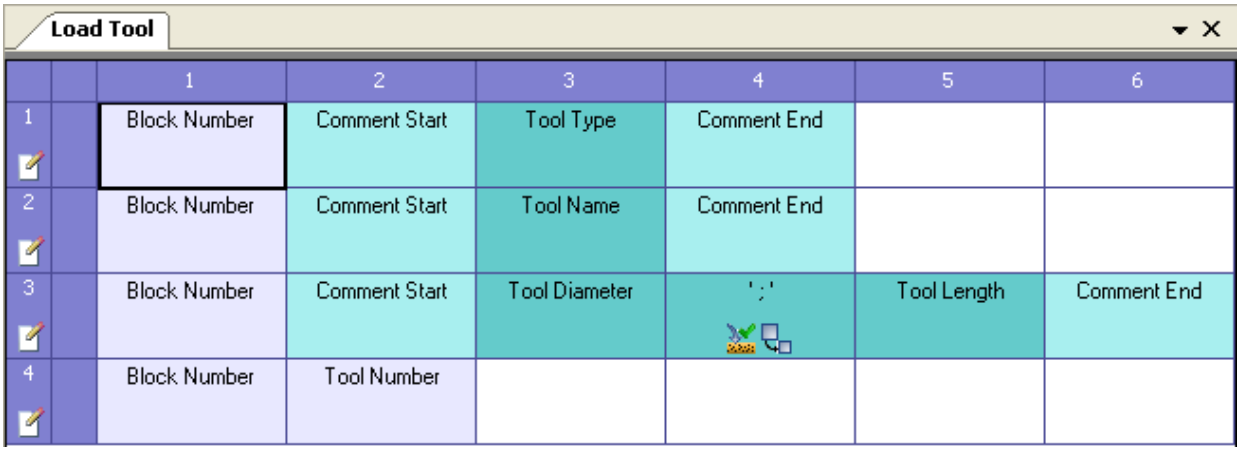
> Nested commands include nested commands.

Compound commands are useful when you need to re-use fragments of a command response. In this example, the option file has two commands, **Load Tool** and **Load Tool First**, configured as follows:

Load Tool First

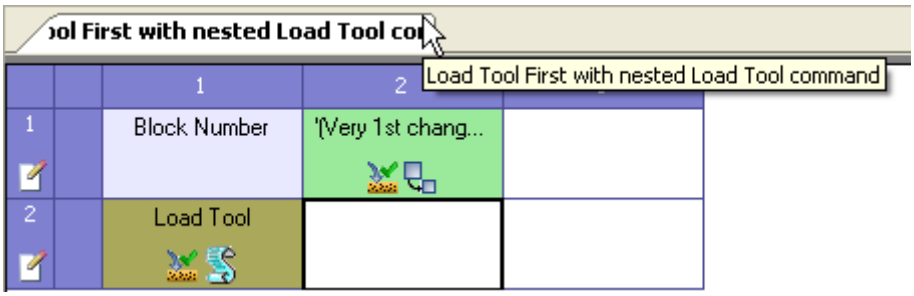


Load Tool



These commands are the same except for the first block of **Load Tool First**. You can avoid making changes in two places by nesting **Load Tool** within **Load Tool First**. Then, changes to **Load Tool** are automatically reflected in **Load Tool First**.

Load Tool First with nested Load Tool command



If a script function is assigned to the **Load Tool** command, you must set the block **Type** to **Output Script Function** in the **Load Tool First** command.

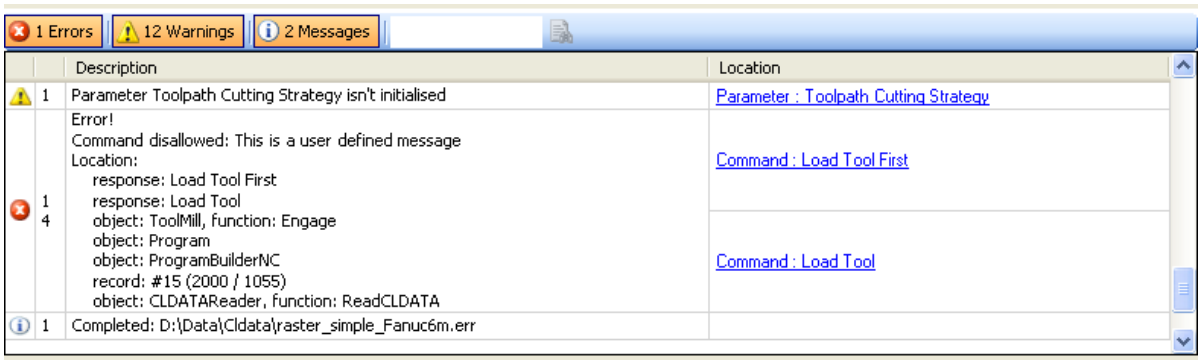
Command context menu

Right-click a **Command** to displays the context menu. Select:

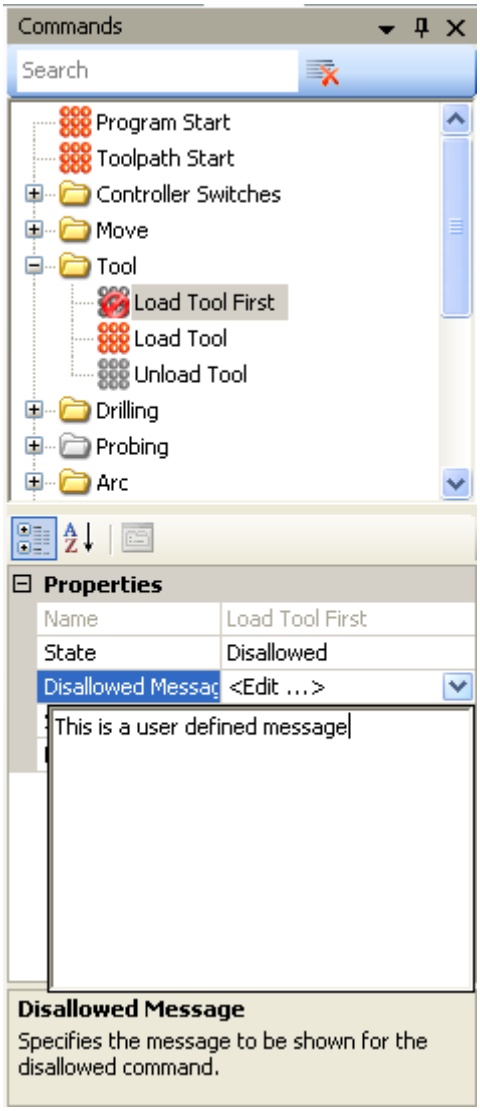
Activate to make an **Inactive** or **Disallowed** command **Active**. A command must be active if it is to be used in postprocessing.

Deactivate to make an **Active** or **Disallowed** command **Inactive**. Deactived commands are ignored in postprocessing.

Disallow to make an **Active** or **Inactive** command **Disallowed**. This allows users to choose whether to allow the postprocessing of a command that is unsupported by some machines.



Enter the message you want to display when Delcam PostProcessor encounters the command in the **Disallowed Message** field in **Command Properties**.



Rename to rename a user defined command.

Copy to copy a command.

Delete to delete a user defined command.

Block Item context menu

Right-click a block item in a **Command** view to display the context menu. Select:

Dependency to control the output of the parameter value and its prefix. Refer to **Block Item Properties** for details.

Output Next to force or suppress the appearance of the value next time. Refer to **Block Item Properties** for details.

Output to Tape to controls the output to tape of the value of a parameter, embedded command, or string, from the position in the current block. Refer to **Block Item Properties** for details.

Workplane For Output to output the coordinates of the block item in the workplane from the program that generated the CLDATA file (for example, the PowerMILL workplane). Refer to **Block Item Properties** for details.

Cut to cut the selected items to the clipboard.

Copy to copy the selected items to the clipboard.

Paste to copy the contents of the clipboard to the selected location.

Delete to delete the block item.

When you paste items from another option file, the **Paste Wizard** is displayed so you can choose whether to paste settings associated with those items. For example, if you paste a command from another option file, you must also decide whether to paste the states and values associated within any parameters that already exist in the target option file.

Select All to select all blocks in the command. You can then copy and paste the clipboard to another command.

Comment Selection to convert the selected block items into a comment. This adds a **Comment Start** and **Comment End** item to the start and end of the selection. When NC comments are enabled in the **General** view, these items insert **Comment Start** and **Comment End** character strings into the NC Program file, and cause the controller to treat the enclosed items as comments.

- For information on passing comments from the CLDATA file to the NC Program file, see the **Comment** topic.

Positive Link to link parameter A to parameter B, so that A is written out only when B is written out. For more information, see Linkage.

Negative Link to link parameter A to parameter B, so that A is written out only when B is not written out. For more information, see Linkage.

Unlink to cancel the link for the selected block. For more information, see Linkage.

Block to edit selected block properties.

- **Long Block** to link blocks together.
- **Disabled** to treat the selected block as a comment and prevent Delcam PostProcessor from creating any code for it. The block is greyed to indicate it is disabled.
- **Select** to select all boxes in the block. You can then copy and paste the block or drag it by keeping the left mouse button depressed
- **Comment** to convert the selected block into a comment. This adds a **Comment Start** and a **Comment End** item to the start and end of the block. When NC comments are enabled in the **NC Comments** view, these items insert **Comment Start** and **Comment End** character strings into the NC Program file, and cause the controller to treat the enclosed items as comments.

Go to Format to change the format of the item.

Item Properties to show properties of selected item.

- **Linking block items**
- **Block description**

Linking block items

Linking enables you to make the output of one block item dependent on the output of one or more other items in the same block.

To create a link between block items:

1. Right-click the block item you want to link and select:

   Positive if you want to output the block item only when the linked item is output.

   Negative if you want to output the block item only when the linked item is not output.

   The cursor changes to indicate the type of link you selected.

2. Click the block items you want to link to. A link icon is displayed in the first block item to show it is linked.
3. Right-click to complete the link.

   To display the block items in a link at any time, click the link icon.

To break a link:

1. Right-click the block item you want to unlink and select **Unlink**.
2. Click the block items you want to unlink. The colour of the block item changes to show it is no longer linked.
3. Right-click to stop unlinking.

   The link icon is removed when all the block items are removed from the link.

Block description

Use block description to enter information about the blocks in a Command.

To create or edit a block description:

1. Select any item in the row for which you want to enter information.

   The block description is displayed in **Item Properties - Block - Description** field.

2. Enter the information to the field. The **Block Description** displays icon to indicate a note has been entered.

To display the note as a tooltip, move the mouse cursor over the note icon:

Adding user commands

User commands enable you to output specific parameters. You can call user commands from scripts and nest them inside other commands.

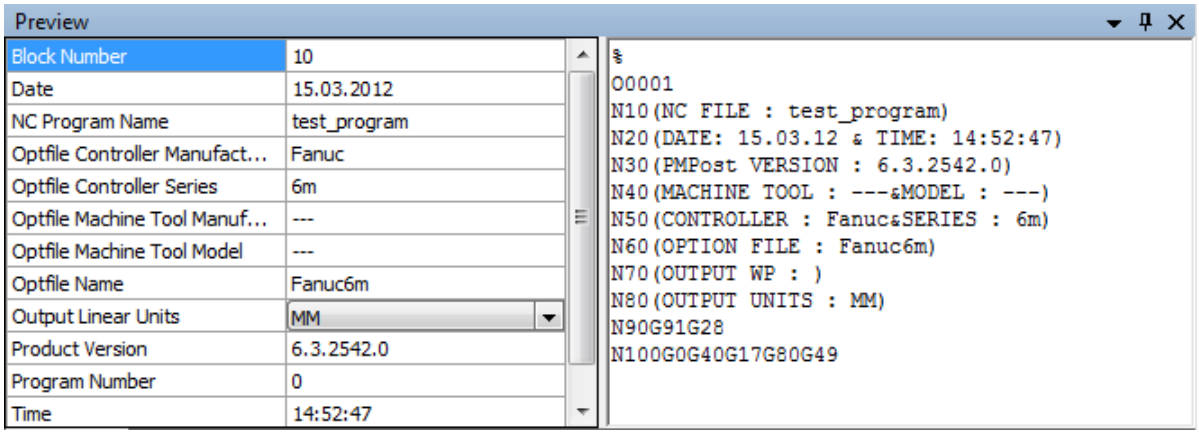
To create a user command:

1. In **Explorer**, right-click **Commands** or **User Commands** and select the **Add User Command** menu option.
2. In the **Add Command** dialog, enter a name for the new command and click **OK**.

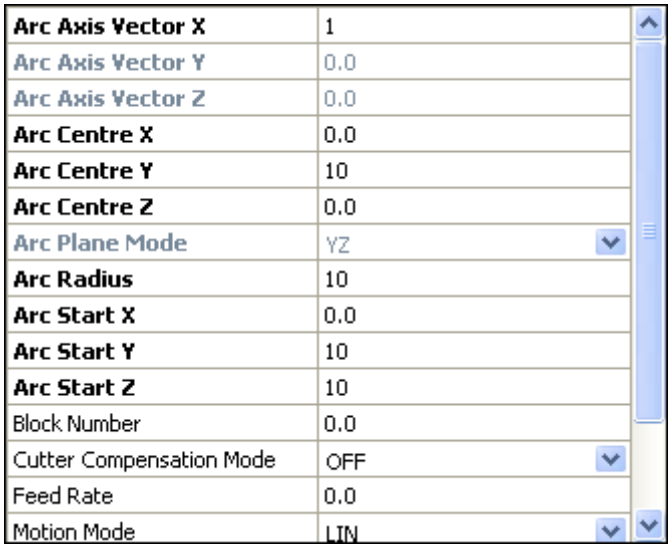
The command is added to the **User Commands** branch of **Explorer**.

Preview

When you have defined a command, you can assign values to the fields in the bottom left corner of the **Message** view, and see a preview of the command in the bottom right corner of the screen. For example:



The list of parameters contains two types of entry:



Parameters shown in **bold text** are permanently contained in the command table. These are often critical to the command - for example, **Circular Move** commands contain most of the parameters related to arc interpolation.

Parameters shown in this **bold text** are permanently contained in the command table and have fixed values, or that are dependent on other parameters for their values, are read only. For example, **Circular Move YZ** command where arc plane parameters are pre-defined already and cannot have other values inside the command.

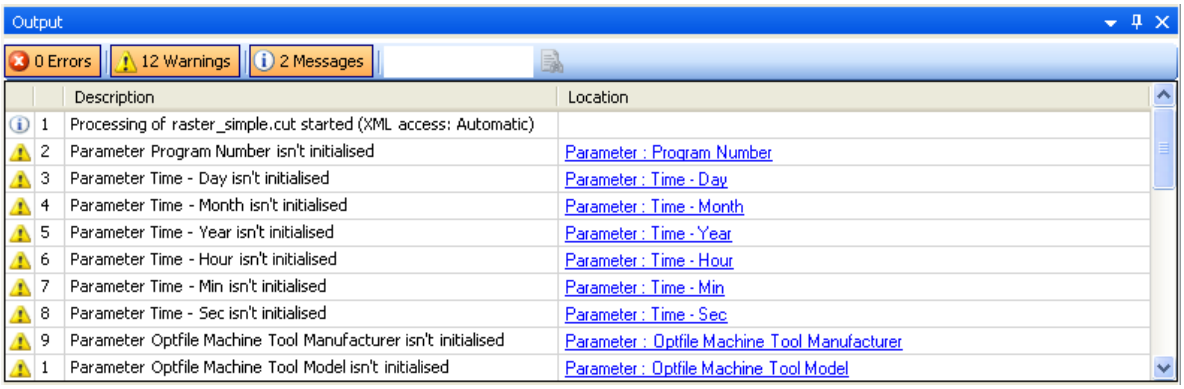
Parameters shown in normal text are not permanently in the command, but are currently in the command block.

All the parameters in the preview table have some initial values, so, for example, in the case of a **Move Circular** commands, **Delcam PostProcessor** can generate arc interpolation correctly in the preview.

You can change most of the values in the parameter list.

Links from error messages

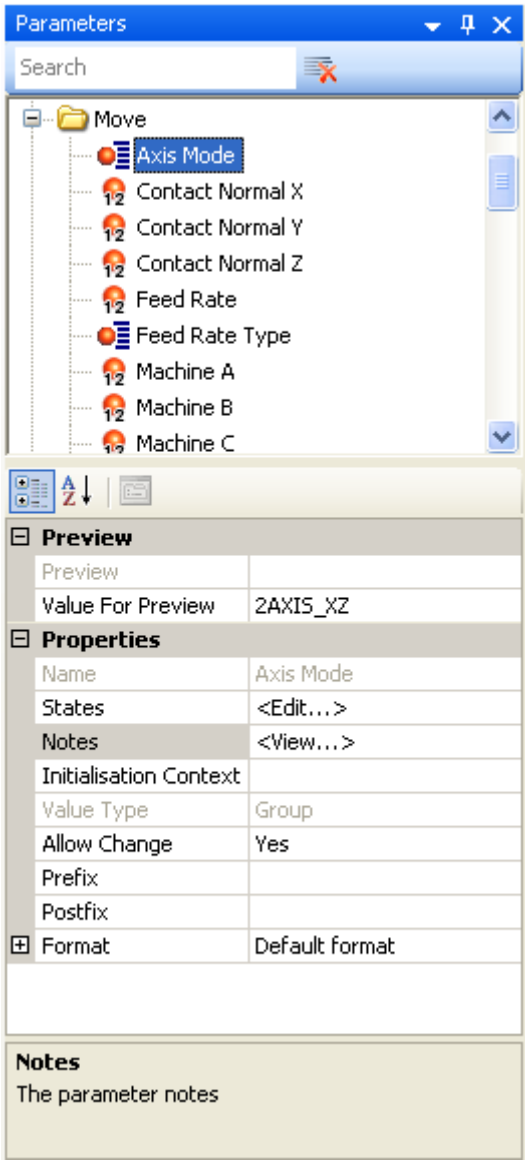
If a cutfile cannot be postprocessed successfully, an error message is displayed in the **Output** pane. This message may contain a hyperlink to the corresponding command, which enables you to jump to the command and amend the definition:



Parameters

Parameters enable you to input values from a CLDATA file and to output values specific to a machine controller.

To list the parameters available in Delcam PostProcessor, click the **Parameters** tab in **Explorer** or select the **View > Parameters** menu option. The tab lists all the available parameters categorized by usage. To display the individual parameters, expand the sub-folders.



The icon displayed to the left of each parameter indicates the type of value it can contain:

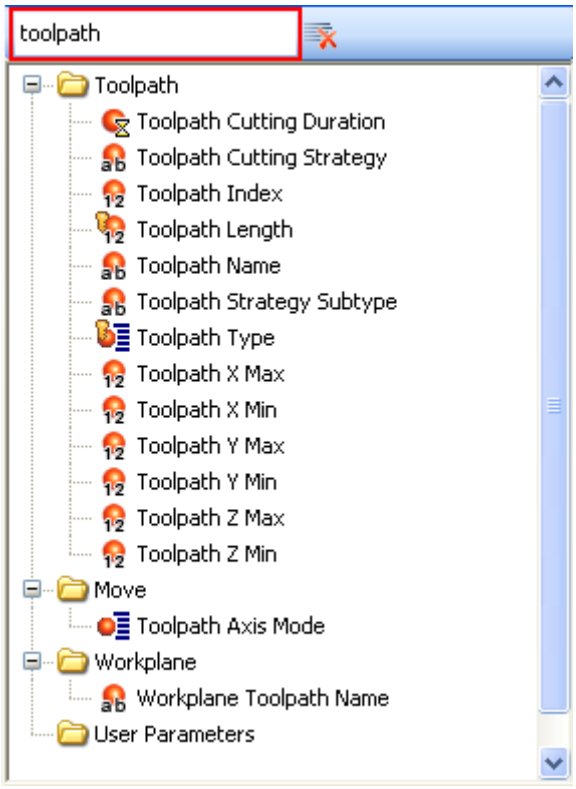
indicates an integer or real number

indicates a text string

indicates a measure of the time taken to perform a task

indicates a value chosen from a list.

To filter the parameters list, enter a string, in the **search field** of the tab's toolbar. The list displays the parameters whose name contains the string. To list all the parameters again, click the **Clear** button.



To edit a parameter's settings, or to change the format assigned to it, select the parameter in the list and edit the properties view at the bottom of the tab.

To add a parameter to a command block:

- Select the cell to which you want to add the command, select the parameter in the parameters list, and click .
- Select its entry in the **Parameter** tab of the **Explorer** and drag it to the block item in which you want to use it.

To create a parameter, right-click **Parameters** or **User Parameters** and select **Add User Standard Parameter** to create an integer, real, string, or duration parameter, or select **Add User Group Parameter** to create a group parameter.

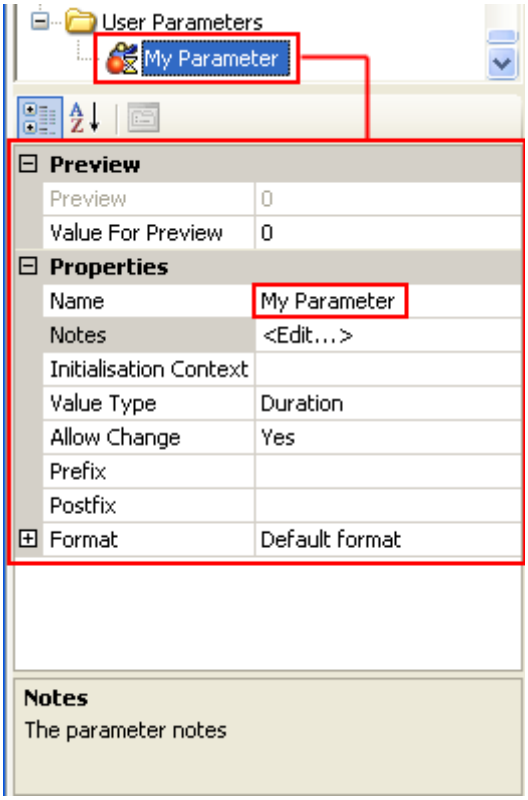
- You must define a machine value for each option in a group parameter before using it; nothing is written to the output file when the **Value** of a **State** is blank. To check the options for a group parameter, select the item in the **Block Items** view. The parameter options are displayed in the **Item Properties**.
- By default, parameters are shown according to option file properties. For example, if arcs are disabled in option file properties arc related parameters become hidden.

Creating parameters

In addition to using the built-in parameters supplied with Delcam PostProcessor, you can create your own parameters for use in Commands. These parameters are listed in the **User Parameter** branch of **Explorer**.

To create a parameter:

1. Right-click in the **Explorer**, and select:
   - **Add User Standard Parameter** to create an integer, real, string, or duration parameter.
   - **Add User Group Parameter** to create a group parameter.
2. In the new parameter dialog, enter a **Name** for the parameter.
3. Select the values that the parameter can hold:
   - For a **User Standard Parameter**, select: **Duration**, **Integer**, **Real** or **String**.
   - For a **User Group Parameter**, define the states that users can choose from and their values.
4. Click **OK** to save your changes and close the dialog. The parameter is displayed in the **User Parameters** folder where you can view to edit the parameter properties.



5. Use the parameter properties view to edit the new parameter's settings:

Preview displays preview for selected parameter with a value inputted to ValueForPreview field.

ValueForPreview specifies a value for Preview output.

Name specifies the parameter name.

Notes displays extra information about the parameter.

ValueType specifies types list. Select the type of value the parameter can hold.

Access defines how the parameter can be used. Select:

- **Read** to specify the value of the parameter cannot be changed by the user. This option is used for parameters whose values are defined in the CLDATA file.
- **Read - Write** to specify the user can change the value of the parameter. Most standard parameters have **Read - Write** access.

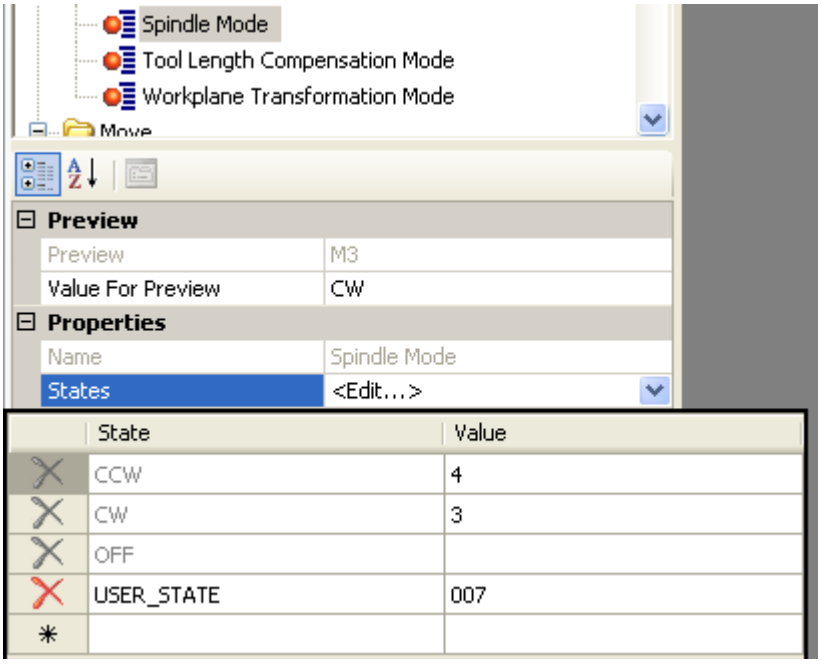
Prefix specifies any text you want to display in front of the value when it is output. For example, when the Prefix is X, a value of -1.23 is output as X-1.23.

Postfix specifies any text you want to display after the value when it is output.

Format specifies the format you want to use to control how the parameter is output.

In Group parameters, specify the options users can choose from. To specify an option, enter a **State** name and its **Value**.

For example, if the possible **Spindle Mode** in CLDATA are **CCW** (counter-clockwise), **CW** (clockwise) and **OFF**, you must specify the machine code for each of these states. If a letter is required before the codes, enter it in the **Prefix** field.



If there are any **User Defined Settings** in **PowerMILL** (under the **Description** tab on the **Tool** form) that you want to be postprocessed, then, for each **Setting**, you must create a **User Parameter** of the same name in **Delcam PostProcessor**, and add a prefix of **udp_**.

For example, if you have a **PowerMILL Setting** called **Pressure**, you must create a parameter in **Delcam PostProcessor** called **udp_Pressure**, and then set a starting value for the parameter in the Initialisation table.

When **Delcam PostProcessor** encounters this **User Defined Setting** in a cut file, it looks for a matching user parameter. If it finds one, **Delcam**

PostProcessor converts the **PowerMILL** value to a value compatible with the type of the **Delcam PostProcessor** parameter. If the conversion is successful, the value is assigned to the **Delcam PostProcessor** parameter. Otherwise, an error (type **Mismatch**) is generated.

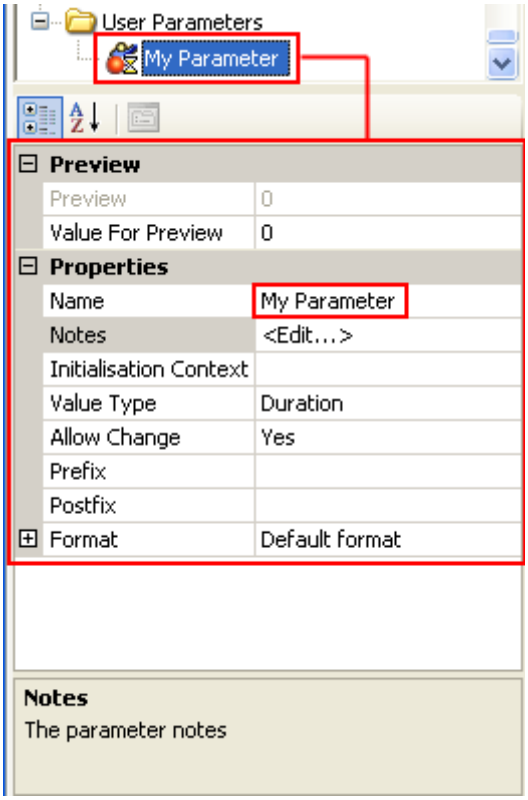
If no equivalent **User Parameter** is found, **Delcam PostProcessor** ignores the **User Defined Setting** from **PowerMILL**.

Editing parameters

Use the **Properties** view to change the settings of built-in parameters and user-defined parameters.

To edit a parameter:

1. Select the parameter's entry in the **Parameters** view of **Explorer**.



2. If you want to record extra information about the parameter, click **Notes**.
3. In the **Format** list, select the format you want to use to control how the parameter is output.
4. In the **Value Type** list, select the type of value the parameter can hold.
5. In the **Access** list, choose how the parameter can be used. Select:
   - **Read** to specify the value of the parameter cannot be changed by the user. This option is used for parameters whose values are defined in the CLDATA file.
   - **Read - Write** to specify the user can change the value of the parameter. Most built-in parameters have **Read - Write** access.

6. In the **Prefix** field, enter any text you want to display in front of the value when it is output. For example, when the Prefix is **X**, a value of **-1.23** is outpu t as **X-1.23**.

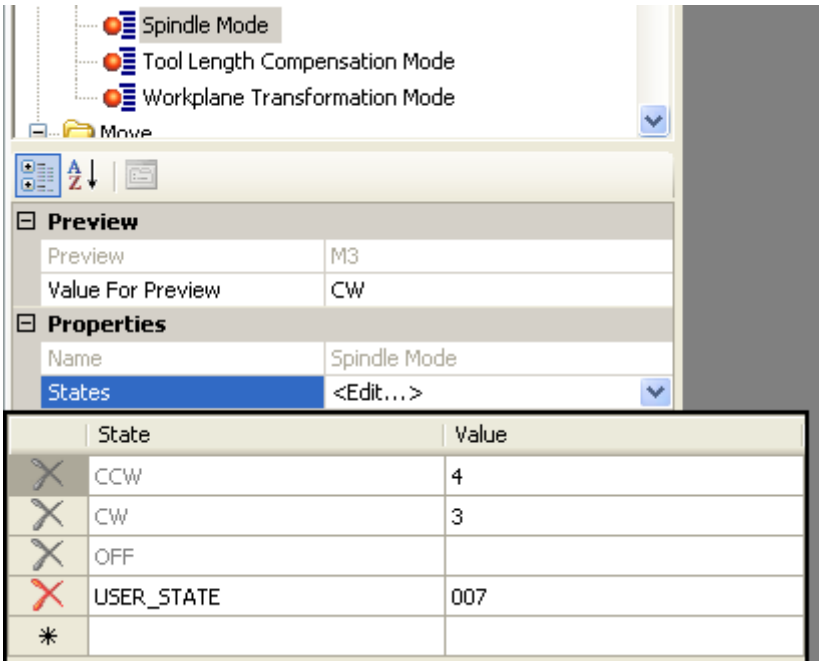
7. In the **Postfix** field, enter any text you want to display after the value when it is output.

8. **Incremental Mode** - Select **True** to make output values relative to the previous value. For example, if the value to output is 10 and the previous value was 4, **Incremental Mode** outputs the value as 6 (10-4). Select **False** to output the actual value. **Incremental Mode** is allowed for most coordinates, machine angles and real parameters created by user (see built-in parameters for more information).

9. If you want to check how the parameter values will look when they are output, enter a value in the **ValueForPreview** field. The value is displayed in **Preview** field with any **Prefix** and **Postfix,** plus the effect of the currently assigned **Format**.

10. In **Group** parameters, specify the options users can choose from. To specify an option, enter a **State** name and its **Value**.

For example, if the possible **Spindle Mode** in CLDATA are **CCW** (counter-clockwise), **CW** (clockwise) and **OFF**, you must specify the machine code for each of these states. If a letter is required before the codes, enter it in the **Prefix** field.



- You must define a machine value for each option in a group parameter before using it as nothing is written to the output file when an option is left blank.

Polar coordinates

There is another way to specify coordinates of a point named as "polar coordinates".

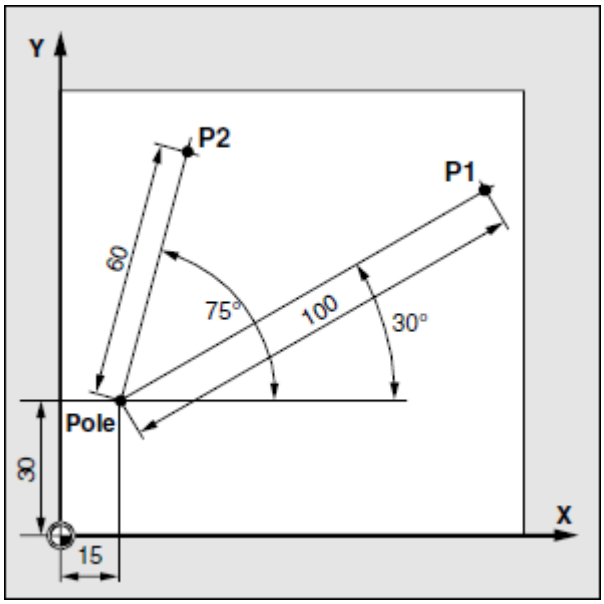
Polar coordinates are useful in cases where a workpiece or part of a workpiece is dimensioned by radius and angle. The origin of the dimensional measurements is referred to as the "pole".

Example:

The points P1 and P2 can then be described with reference to the pole as follows:

P1 corresponds to radius =100 plus angle =30?

P2 corresponds to radius =60 plus angle =75?



The following parameters implement polar coordinates in the option file:

Polar Angle, **Polar Radius**, **Polar Centre X**, **Polar Centre Y**, **Polar Centre Z**.

Polar coordinates are calculated relative to the program coordinate system (workplane). They are not affected by workplane transformation or by disabled RTCP. Polar coordinates are calculated in a plane defined by Plane Mode parameter. The polar centre values are read/write and set as (0, 0, 0) by default. In other words, it points to the origin of the program coordinate system.

The examples below demonstrate how polar coordinates are used by different CNC controllers:

Siemens 840D

```
G111 X25 Y25 Z10      ; set up the pole
G1 AP=-30 RP=110 F250 ; linear move (RP ? radius, AP ? angle)
G0 AP=30 RP=110       ; rapid move
```

Heidenhain TNC530

```
18 CC X+25 Y+25       ; set up the pole
19 LP PR+20 PA+0 F250 ; linear move (PR ? radius, PA ? angle)
20 CP PA+180 DR+      ; circular move
```

Fanuc 16i

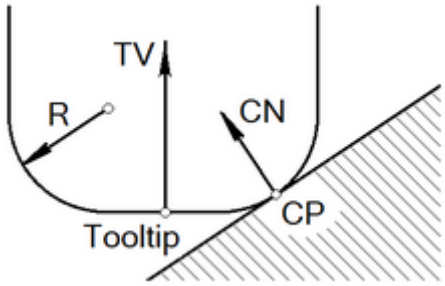
```
N1 G17 G16          ; enable polar coordinates mode in XY plane
N2 G1 X120 Y100 F1000 ; linear move (X ? radius, Y ? angle)
N3 G0 X220 Y100      ; rapid move
N4 G15               ; cancel polar coordinates mode
```

Contact Point

Usually moves in NC-program contain positions of tooltip. However, some controllers are able to compensate tool geometry. This feature is known as **3D Cutter Compensation**. When it's used, the CAM system additionally provide **Contact Normal**.

PostProcessor calculates coordinates of **Contact Point** for each move from **Tooltip Radius**, **Tool Diameter** and the contact normal.

Difference between **Tooltip** point and **Contact Point** is shown on the image below:



Used abbreviations:

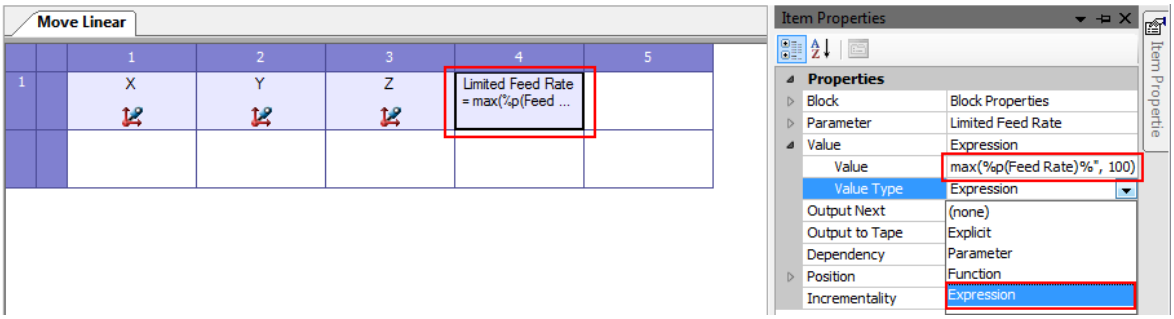
- TV - Tool Vector
- CN - Contact Normal
- CP - Contact Point
- R - Tooltip Radius
- **Contact Point** calculated only when the CAM system provides **Contact Normal**. To achieve that you need to enable **3D Cutter Compensation** in **PowerMill**.

Assigning expressions to parameters

You can assign the result of an expression to any parameter using the **Expression** option in the **Value Type** and **Value** fields.

To assign an expression to a parameter:

1. In the required Command, add the parameter to a Block Item.
2. Select the item. The **Item Properties** window displays the properties of the item.
3. In the **Value Type** list, select **Expression**.



4. In the **Value** field, type the expression. The expression can contain:
   - Any built-in or user parameter wrapped as %p(name)%.
   - Any of the following operators: +, -, *, /, %, (, or).
   - Any of the following mathematical functions: abs, acos, asin, atan, ceil cos, exp, floor, log, max, min, pow, round, sin, sqrt, tan.

For example, you can assign any of the following expressions to a parameter:

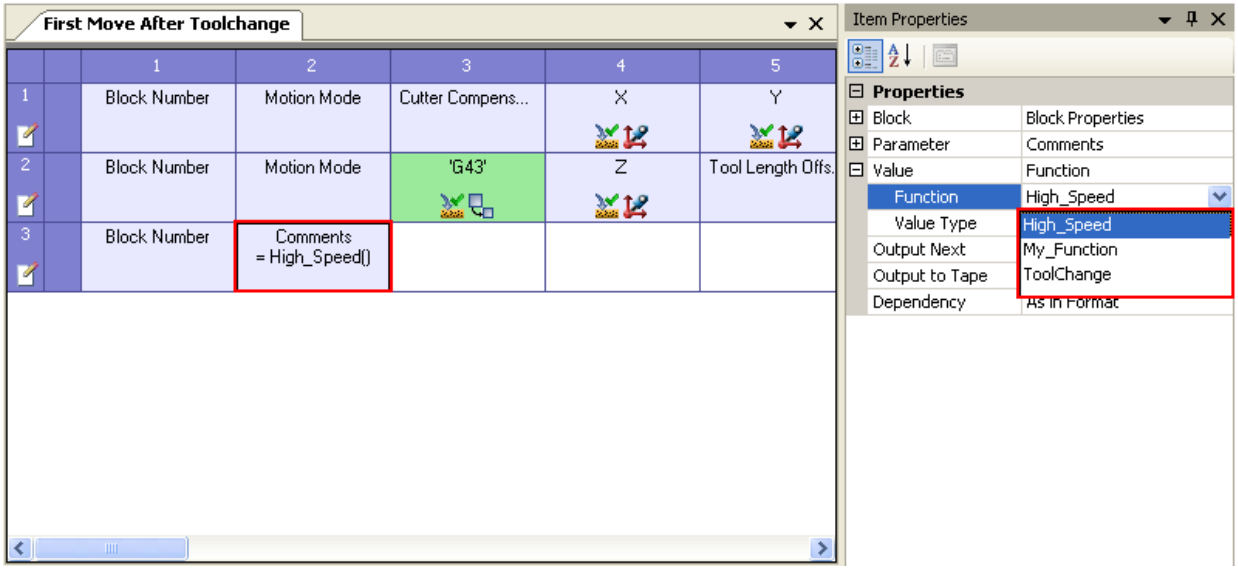
```
Feed Rate = max ( %p(Feed Rate)%, 100 )
X = %p(X)% + %p(User Shift X)%
User X = sin ( %p(User Angle)% ) * %p(User Radius)%
User Y = GetParamPrevValue("%p(Y)%")
```

Assigning script functions to parameters

You can assign the output of a script function to any parameter using the **Value** list in the **Item Properties** window.

To assign the output of a function to a parameter:

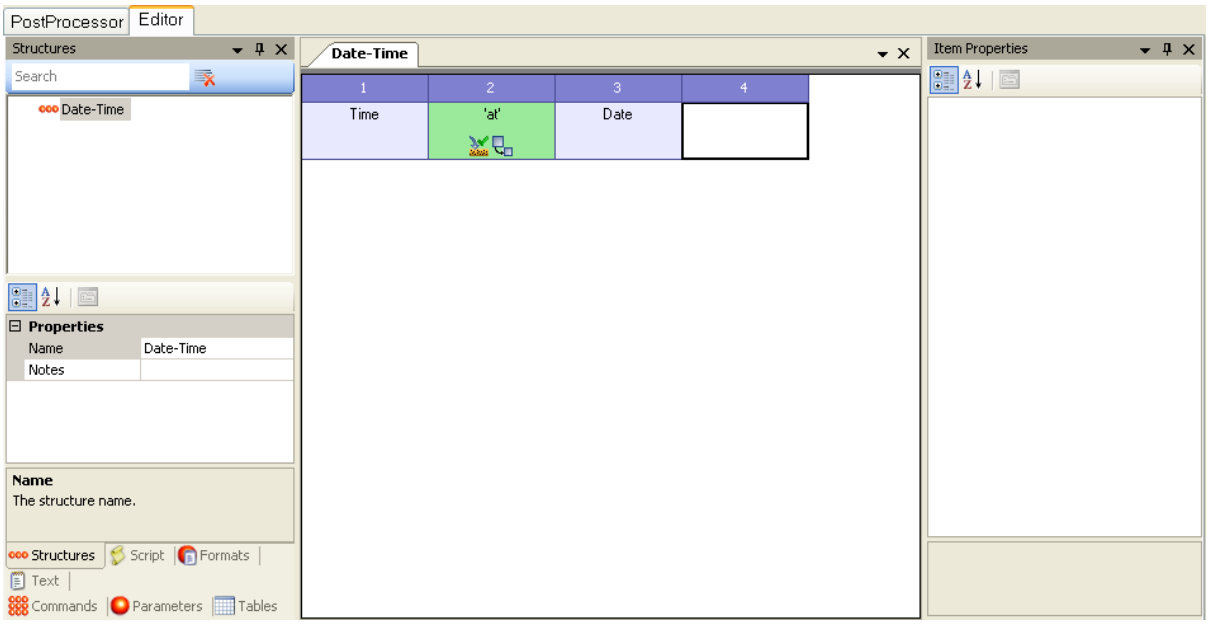
1. Create the script function.
2. In the required Command, add the parameter to a Block Item.
3. In the **Value** list of the **Item Properties** window, select **Function**.
4. In the **Function** list, select the script from the drop-down list. For example:



Structures

Structures enable you to combine several parameters into a single block item. Use them when you want to quickly assign the same set of parameters to several different command blocks. Structures also enable you to quickly make changes to several commands simultaneously because changes to a structure are reflected in all the commands in which it is used.

A structure looks like a simplified block, except that it contains a single string and no **Block Number**:



To create a structure:

1. Select the **Structures** tab, right-click in the **Explorer** and select the **Add Structure** menu option.
2. In the **Add Structure** dialog, enter a **Name** for the structure, and click **OK**. The structure is displayed in **Explorer**.
3. Add parameters to the structure by selecting them from the drop-down list and clicking the **Add Parameter** button or by dragging them from the **Parameter** tab in **Explorer**.

Using structures

To add a structure to a command:

- Select **Structure** in the **Select Item Type** list, select the structure in the **Select item** list, and click the **Add Parameter** button.
- Drag the structure from **Explorer** and drop it into the block where you want to use it.

The structure is displayed with an orange background.

To display the parameters within a structure, right-click the block item and select the **Unfold** menu option. The parameters are shown with an orange background.

You cannot edit the parameters within the structure, or insert items in between them.

To collapse the structure again, right-click a parameter within the structure and select **Fold**.

To change the contents of a structure, right-click the block item and select the **Go To Structure** menu option.

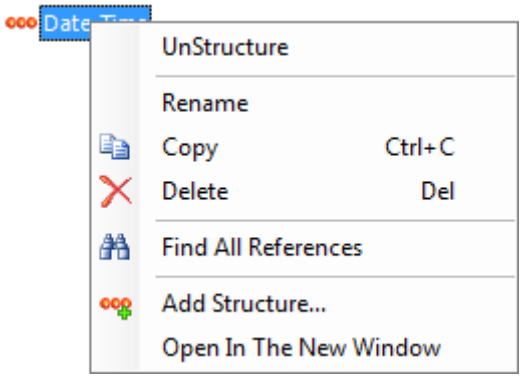
> Editing a structure affects all commands in which the structure is used.

To convert the contents of the structure into block items, right-click the structure and select the **Unstructure** menu option.

> This option breaks the connection with the structure, and further modifications to the structure will have no effect on the command.

Individual Structure context menu

Right-clicking an individual structure displays the following context menu:



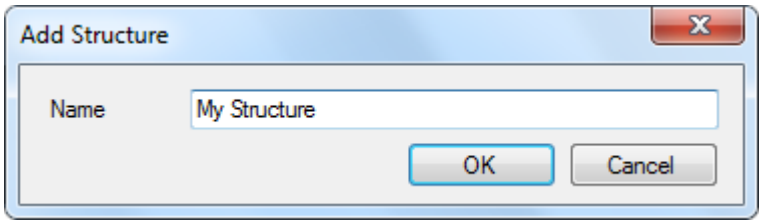
Unstructure removes the individual parameters from the structure in **every** command block that contains the structure. The individual parameters can then be reconfigured.

Rename renames the structure.

Copy copy the selected structure to the clipboard.

Delete deletes the structure.

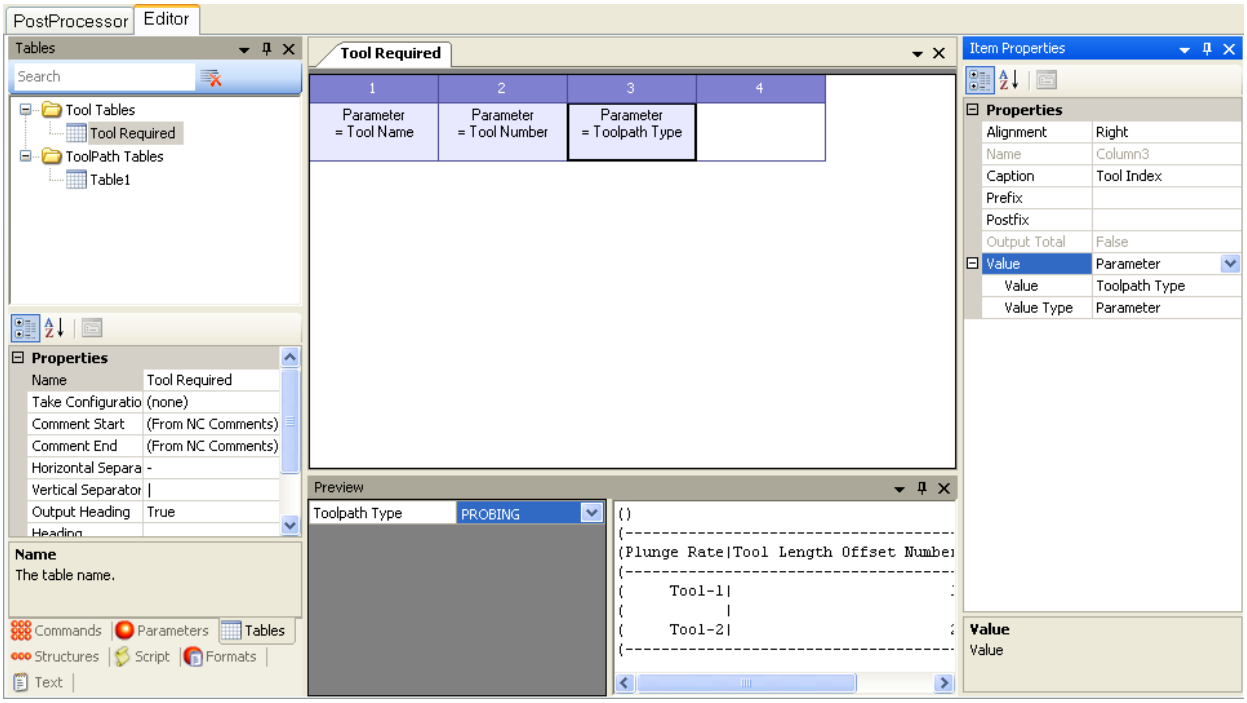
Add Structure allows you to add new structure.



In the **Add Structure** dialog, enter a **Name** for the structure, and click **OK**. The structure is displayed in **Explorer**

Tables

Tables enable you to list all tools and cutters together so that you can notify the machine operator of the NC program's requirements (by, for example, adding the relevant **Tool Table** to **Program Start** and the relevant **Toolpath Table** to **Toolpath Start**).



If you insert a **Toolpath Table** into the **Toolpath Start** command, you can choose the **Type** of output that creates. Select:

- A **Type** of **Toolpath** outputs the rows individually. For example:

```
N110G0G90Z0
N120( ================= )
N130(Start Toolpath)
N140( ================= )
N150(-------------------------------------------------------------------------------)
N160(Tool Number|Tool Name|     Tool Type|Toolpath Name|Toolpath Type|Toolpath Length)
N170(-------------------------------------------------------------------------------)
N180(           1|    tool1|        ENDMILL|    toolpath1|    FINISHING|          147.19)
N190(-------------------------------------------------------------------------------)
N200( ================= )
N210( TOOLPATH : toolpath1 )
N220( STRATEGY USED : Pattern )
N230( TOOLPATH WP : World )
N240( ================= )
N250( TOOL TYPE : ENDMILL )
N260( TOOL NAME : tool1 )
N270( TOOL DIA.: 10 & TIP RAD.: 0 & LENGTH : 50 )
N280T1M6
N290G54G90
N300S1500M3
N310M8
N320G0X25.579Y23.917
N330G43Z0H1
N340X0Y0
N350G1X51.157Y25.045F1000
N360X20.534Y47.834
N370X0Y0
N380G0X25.579Y23.917
N390( ================= )
N400(Start Toolpath)
N410( ================= )
N420(-------------------------------------------------------------------------------)
N430(Tool Number|Tool Name|     Tool Type|Toolpath Name|Toolpath Type|Toolpath Length)
N440(-------------------------------------------------------------------------------)
N450(           3|    tool3|    TIPRADIUSED|    toolpath2|    FINISHING|          147.19)
N460(-------------------------------------------------------------------------------)
```

Output one row for each toolpath

A **Type** of **Program** outputs all the rows in the table. For example:

```
%
N10 (ToolPath Table)
N20 (-------------------------------------------------------------------------------)
N30 (Tool Number|Tool Name|     Tool Type|Toolpath Name|Toolpath Type|Toolpath Length)
N40 (-------------------------------------------------------------------------------)
N50 (           1|    tool1|        ENDMILL|    toolpath1|    FINISHING|          147.19)
N60 (           3|    tool3|    TIPRADIUSED|    toolpath2|    FINISHING|          147.19)
N70 (           2|    tool2|      BALLNOSED|    toolpath3|    FINISHING|          147.19)
N80 (           2|    tool2|      BALLNOSED|    toolpath4|    FINISHING|          147.19)
N90 (           5|    tool5|    TAPERTIPPED|    toolpath5|    FINISHING|          147.19)
N100(           2|    tool2|      BALLNOSED|    toolpath6|    FINISHING|          147.19)
N110(           4|    tool4|TAPERSPHERICAL|    toolpath7|    FINISHING|          147.19)
N120(           1|    tool1|        ENDMILL|    toolpath8|    FINISHING|          147.19)
N130(           5|    tool5|    TAPERTIPPED|    toolpath9|    FINISHING|          147.19)
N140(           5|    tool5|    TAPERTIPPED|   toolpath10|    FINISHING|          147.19)
N150(-------------------------------------------------------------------------------)
N160(     Total:|         |              |             |             |         1471.9)
N170(-------------------------------------------------------------------------------)
:0001
N180( NC FILE : dicc00325_ascii_Fanuc )
N190( DATE : 17.07.06 & TIME - 10:43:04 )
N200( PMPost VERSION : 4.30 CB0216 )
N210( MACHINE TOOL : --- & MODEL : --- )
N220( CONTROLLER : Fanuc & SERIES : --- )
```

Output whole table

If you insert a **Tool Table** into a command, or insert a **Toolpath Table** into any command other than **Toolpath Start**, the table is output automatically.

Creating a table

To create a new table:

1. In the **Tables** tab, right-click in the **Option File Explorer**, select **Add Table** from the context menu. The **Add Table** dialog is displayed.
2. Enter a **Name** for the tool.
3. Choose a **Type** from the drop-down list. Select:
   - **Tool Table** to base the number of rows on the number of tools in the cutfile. For example:

     A cutfile contains three toolpaths (**Toolpath1**, **Toolpath2** and **Toolpath3**), and two tools (**tool1** for **Toolpath1** and **Toolpath2**, and **tool2** for **Toolpath3**). The table contains the columns **Tool Name** and **Toolpath Name**.

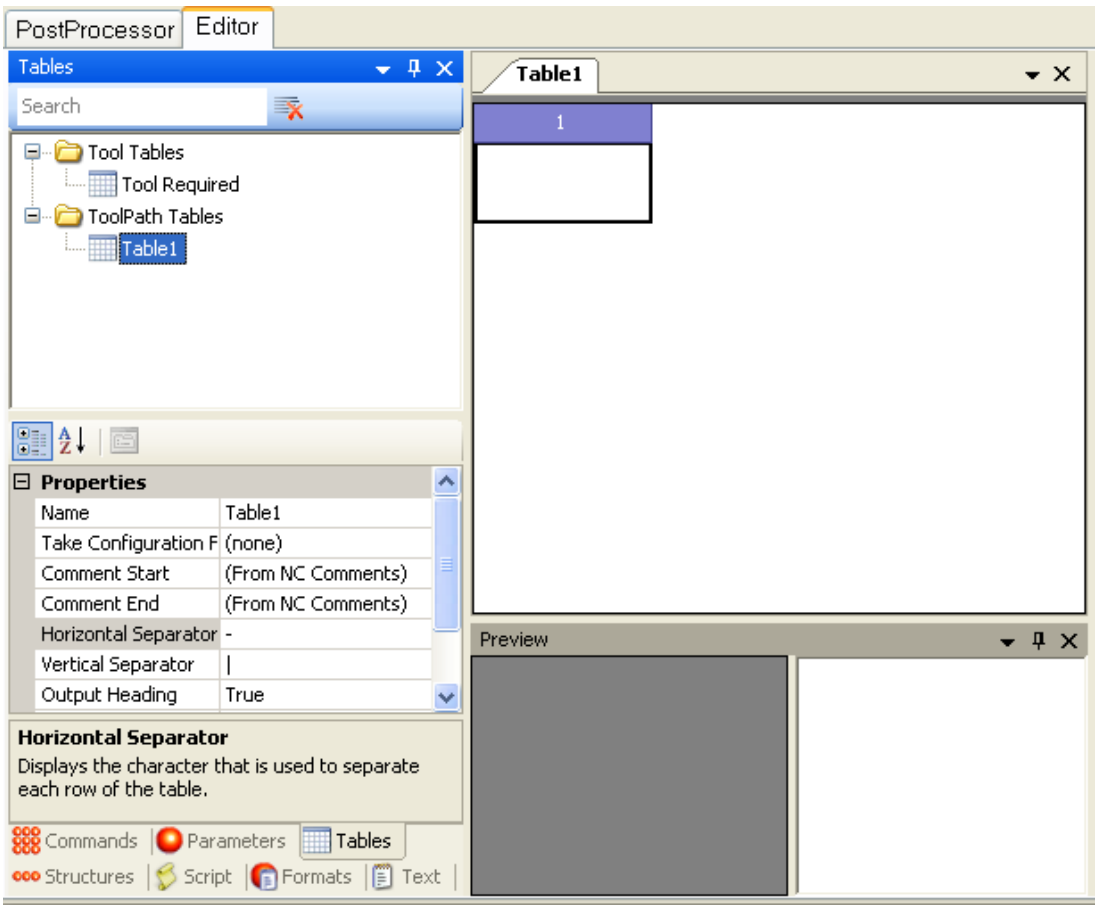
     The **Tool Table** is output as follows in the NC program:

```
-----------------------------
| Tool Name | Toolpath Name |
-----------------------------
|     tool1 |     Toolpath1 |
|           |     Toolpath2 |
|     tool2 |     Toolpath3 |
-----------------------------
```

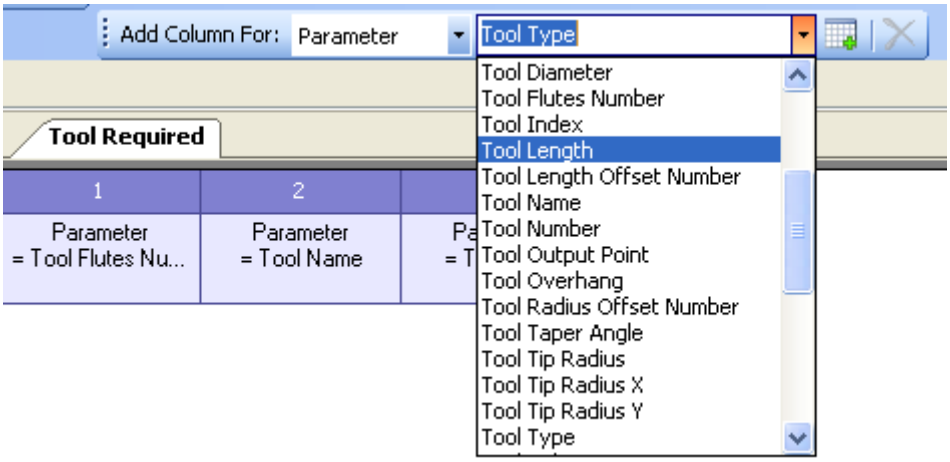
   - **Toolpath Table** to base the number of rows on the number of toolpaths in the cutfile. To continue the above example, the **Toolpath Table** is output as follows in the NC program:

```
-----------------------------
| Tool Name | Toolpath Name |
-----------------------------
|     tool1 |     Toolpath1 |
|     tool1 |     Toolpath2 |
|     tool2 |     Toolpath3 |
-----------------------------
```

   - If the **Toolpath Table** is inserted into the Toolpath Start command, only a single row is output. Refer to Tables.
4. Click **OK**. The table is added to the **Tables** branch, and its contents are displayed in the **View** pane:



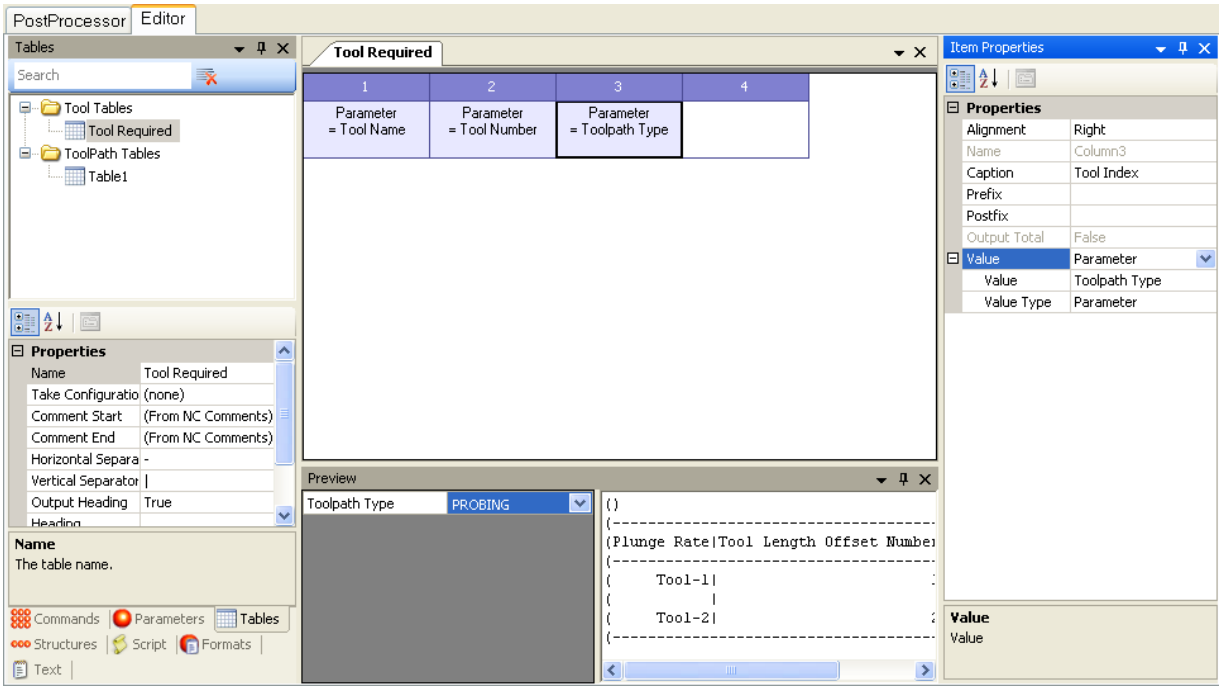
5. Complete the **Properties** fields:
   - **Name** - Specifies the name of the selected table. To edit the name, click in the field, and overwrite the text.
   - **Take Configuration From** - If you want to base the configuration on an existing table, select the table from the drop-down list. The remaining configuration settings then become unavailable for editing.
   - **Comment Start** - Adds the specified characters to the start of each row in the table.
   - **Comment End** - Adds the specified characters to the end of each row in the table.
   - **Horizontal Separator** - Specifies the character to be used to separate each row of the table in the NC program.
   - **Vertical Separator** - Specifies the character to be used to separate each column of the table in the NC program.
   - For the display of the characters, refer to Previewing a Table.
   - **Output Heading** - Select **True** to output the specified **Heading** in the NC program.
   - **Heading** - Specifies the heading of the table for the NC Program.
   - **Notes** - Use this field to record additional information about the table.
6. To add or remove a column, use the **Table** toolbar:



For information on adding a table to a command, refer to Command Interface.

Previewing a table

The **Preview** pane is available for tables and commands.

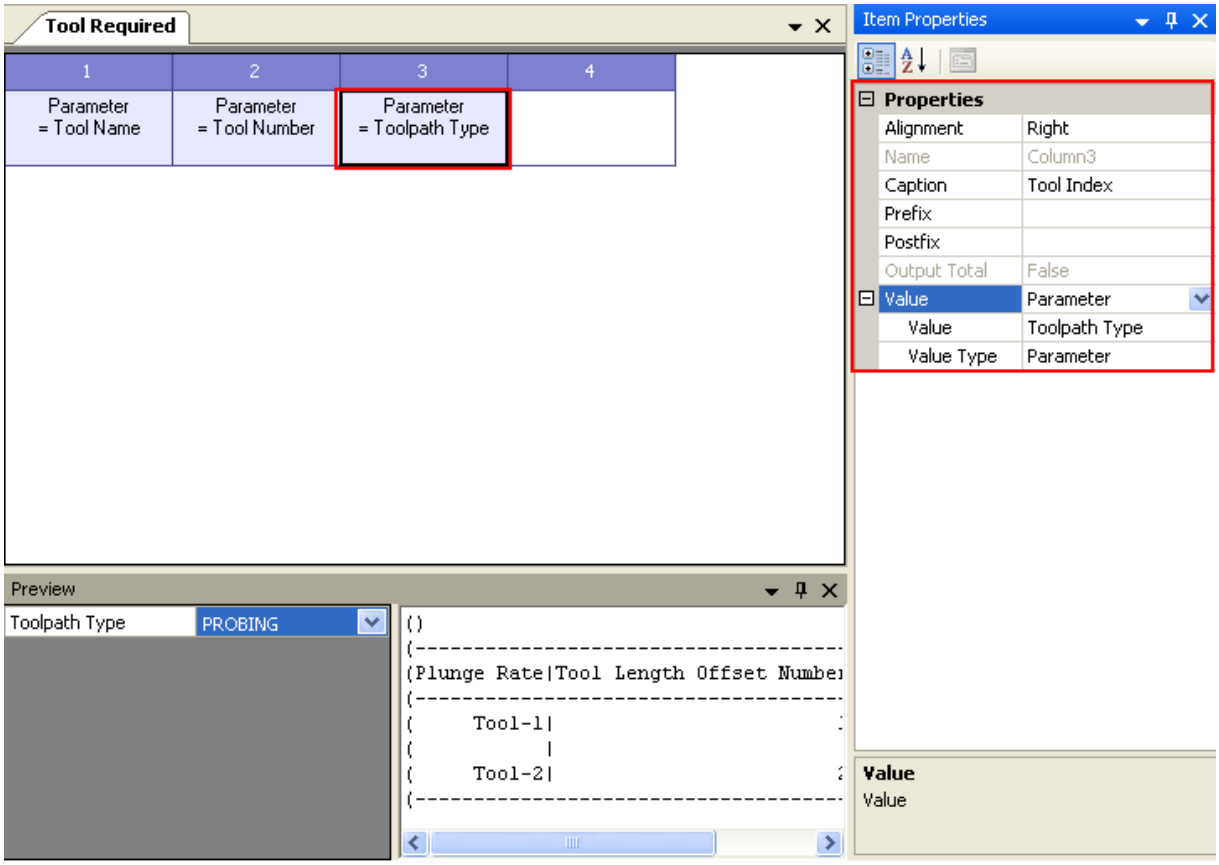


To test the table, enter values in the parameter list at the bottom left of the window. The values are displayed in the example pane on the right.

Editing a column

Use the **Item Properties** view to specify the layout of the selected column.

To display the properties, select a column in the table. The **Item Properties** view displays all the settings for the selected column:



Parameter specifies the name of the parameter assigned to the column. Click ⊞ to display the parameter properties.

Alignment specifies the alignment (**Left**, **Right** or **Center**) of the column values.

Name specifies the internal name of the column.

Caption specifies the name of the column used in the output.

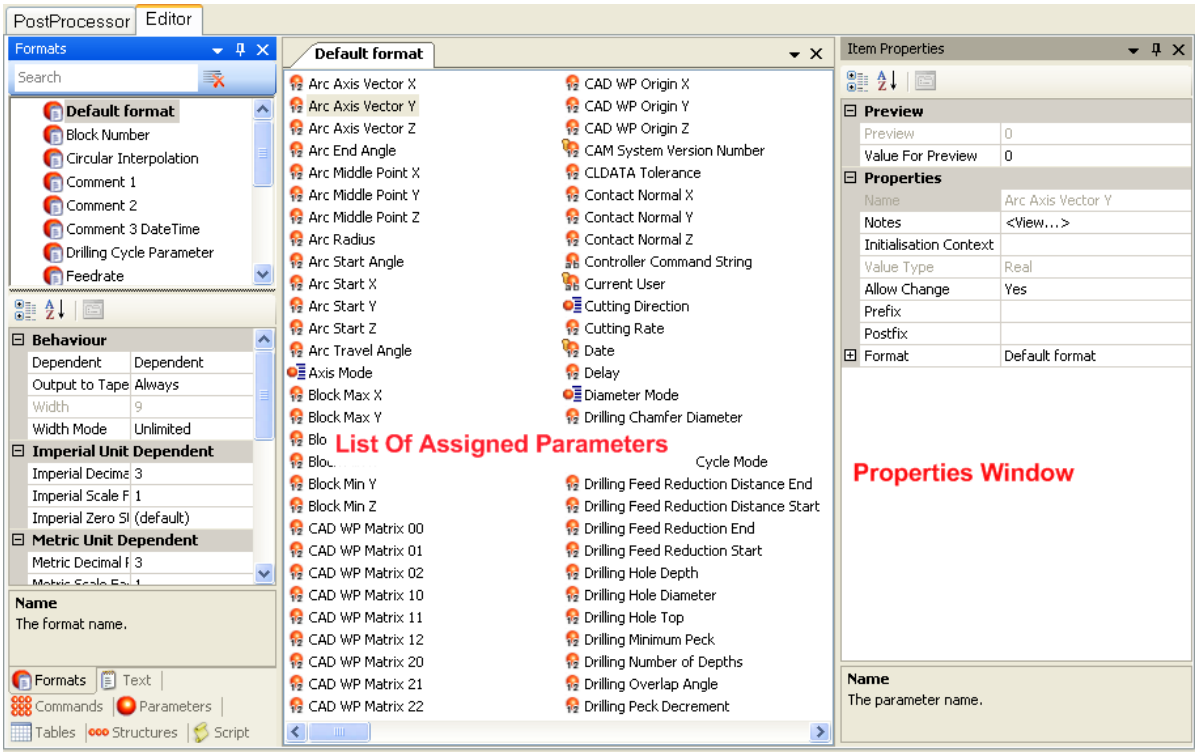
Prefix specifies any text to be added to the start of the column values when output. For example, if the **Prefix** is *X* and the column value is *-1.23*, the output is *X-1.23*.

Postfix specifies any text you to be added to the end of the column values when output. For example, if the **Postfix** is *_A* and the column value is *-1.23*, the output is *-1.23_A*.

Output Total when *True* is selected for a column parameter that has a numeric value, Delcam PostProcessor outputs the total value for the column.

Formats

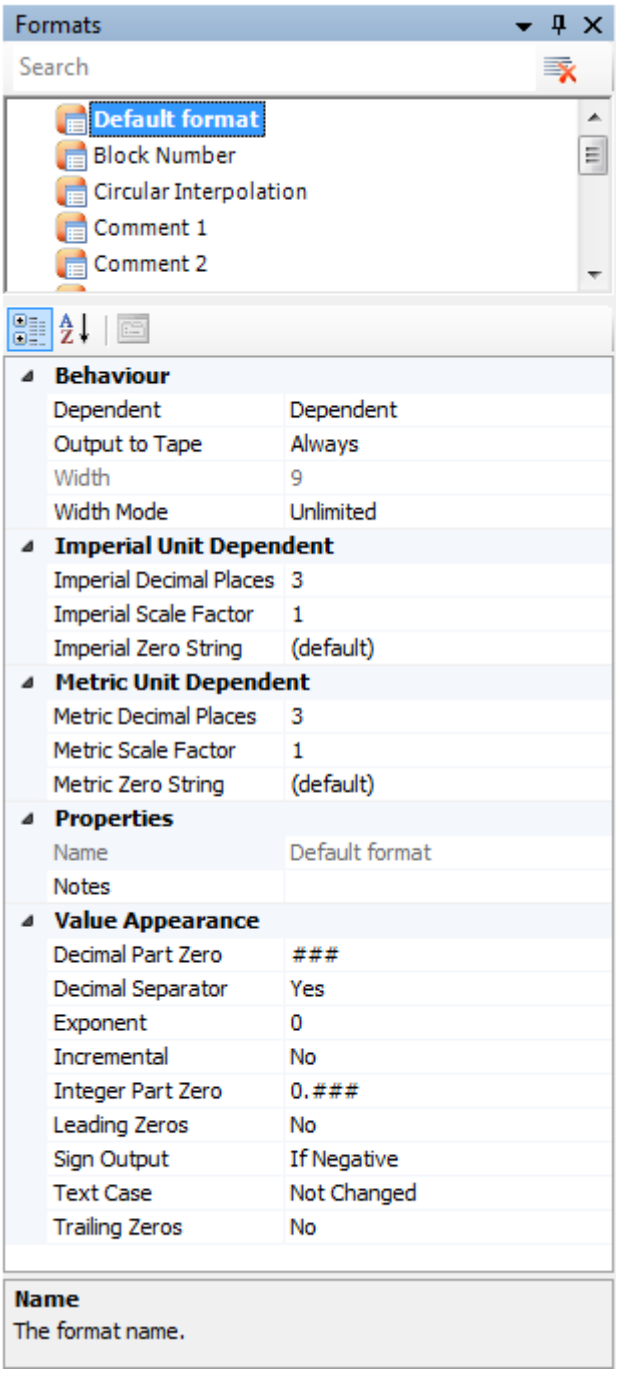
A format is a group of settings that control how parameters are output. Formats enable you to quickly apply settings to parameters so you can ensure that similar values are output in a similar way. Formats also enable you to change the output of all associated parameters in one place.



To create a format, right-click in the **Formats Explorer** and select the **Add Format** menu option.

Using the Formats Explorer

The **Formats Explorer** lists all defined formats and the properties of the selected format.



The **Properties** panel contains the following sections:

Behaviour settings control how the parameters associated with this format are output by default.

Dependency - if **Output to Tape** is set to **Always**, **Always Prefix**, or **If Updated**, the **Dependent** setting controls the output of the parameter value and its prefix. Select:

- **As in Format** to use the setting from the format assigned to this parameter. If you change the default **As in Format** setting in the command block, the

format setting is overridden. This setting is only available for selection in command blocks.

- **Independent if Appears** to treat the block item as independent when it appears. Output is controlled by the **Output to Tape** setting.
- **Independent if Updated** to treat the block item as independent when its value changes. Output is controlled by the **Output to Tape** setting only when the value is different.
- **Dependent** to output the block item in accordance with the **Output to Tape** setting only when there is at least one other, **Independent**, item in the block. This is a useful setting for text items and feed rates.
- A string parameter has no **Output to Tape** setting. Its output is determined by the **Dependency** setting, which can be **Independent** (output it regardless of the other items in the block); **Dependent** (output it only if one of the other block items is **Independent**); or **None** (always output it).
- **Output to Tape** - controls the output to tape of the value of a parameter, embedded command, or string, from the position in the current block. Select:
- **As in Format (Parameter only)** to use the setting from the format assigned to this parameter. If you change the **As in Format** setting in the command block, the format setting is overridden. This setting is only available for selection in command blocks.
- **Never (Parameter, String and Embedded Command)** to never output the value.
- **Always (Parameter, String and Embedded Command)** to always output the value (for a parameter, this is subject to the **Dependency** and **Output Next Time** settings - see below).
- **Always Prefix (Parameter only)** to always output the prefix (subject to the **Dependency** and **Output Next Time** settings), regardless of whether the value is output.
- **If Updated (Parameter only)** to output the parameter only if the **Updated** setting is **Yes** and the **Dependency** setting allows it.
- A modal value is not output if the **Output Next Time** flag was set to **Suppress** when it was last invoked.
    - For further information, refer to Flow of information relating to the output of a parameter's value.
- **Width** - Specifies the size of the output value in characters when the **Width Mode** is **Maximum** or **Constant**. The value must be a positive number between **1** and **80**.
- Signs and decimal points count as characters in the **Width** calculations.
- **Width Mode** - Controls the width of the parameter value in characters. Select:
- **Unlimited** to display the full value using the minimum number of characters.
- **Maximum** to restrict the width of the output to the number of characters specified in the **Width** field.
- **Constant Left Align** to left align all values at the specified **Width**. Values that contain fewer characters than the specified **Width** are padded (with spaces by default, or with zeros when the **Leading Zeros** check box is selected).

- **Constant Right Align** to right align all values at the specified **Width**. Values that contain fewer characters than the specified **Width** are padded (with spaces by default, or with zeros when the **Leading Zeros** check box is selected).
- Behaviour settings can be overridden in command blocks.
- **Imperial Unit Dependent and Metric Unit Dependent** specify how values are output when using imperial measurements and when using metric measurements.
- **Decimal Places** - When **Decimal Separator** is selected, use this row to specify how many decimal places are displayed.
- **Scale Factor** - Use this option to specify how many times **Metric** and **Imperial** values are to be multiplied by. The default is 1, but if, for example, you change it to 2, this enables you to create diameters from radii.
- **Zero String** - Select an option to specify how a value of zero is represented.
- **Name** specifies the name of the format. To change the name, overwrite the text.
- **Notes** enables you to record extra information about the format.
- **Value Appearance** settings control how values are represented when output.
- **Decimal Part Zero** - Specifies how numeric values with trailing zeros are represented. Select:
    - **XXX.000** to display trailing zeros (even when **Trailing Zeros** is not selected). If **Decimal Point** is not selected, the display becomes **XXX000**.
    - **XXX.0** to display one trailing zero (if **Trailing Zeros** is not selected).
    - **XXX.** to display no trailing zeros (unless **Trailing Zeros** is selected).
    - **XXX** to display no trailing zeros or decimal point (unless **Trailing Zeros** or **Decimal Point** are selected).
- **Decimal Separator** - Select **True** to display decimal points in numeric values.
- **Exponent** - Displays numeric values in exponential format. Select an option in the list to specify the number of exponential digits you want to use. For example, to display -23.45 as **-02.345e+001**, select **3 Digits**.
- **Integer Part Zero** - Specifies how numbers with an integer value of zero are represented. Select:
    - **000.XXX** to pad the value with leading zeros (even when **Leading Zeros** is not selected).
    - This setting is not available when **Width Mode** is **Not Used**.
        - **0.XXX** to display one leading zero.
        - **.XXX** to display no leading zeros (unless **Leading Zeros** is selected).
        - **XX** to display no leading zeros or the decimal point.
    - This setting is not available when **Decimal Point** or **Leading Zeros** is selected.

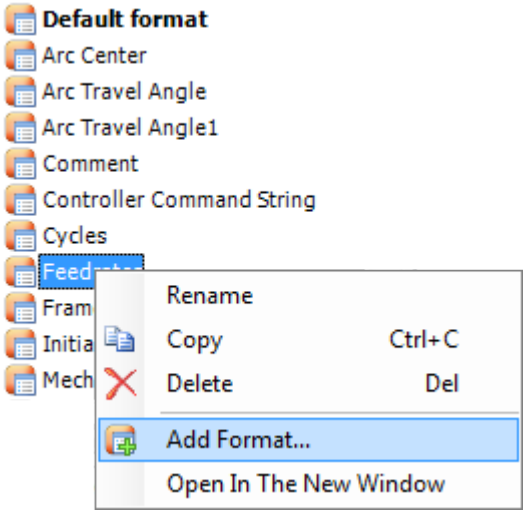
- **Leading Zeros** - Select **True** to pad numeric values with leading zeros up to the number of characters specified in **Width**. For example, if **Width** is **7**, **-23.45** is output as **-023.45** when this is selected.
    - This option is not available when the **Width Mode** is **Not Used**.
- **Sign Output** - Specifies the sign used to display positive and negative values. Select:
    - **If Negative** to display a minus character (-) in front of negative values.
    - **Always** to display a plus (+) character in front of positive values, and a minus character (-) in front of negative values.
    - **Never** to display no sign in front of values.
- **Trailing Zeros** - Select **True** to pad numeric values with trailing zeros up to the number of characters specified in **Width**. For example, if **Width** is **9**, **-23.45** is output as **-23.45000** when **True** is selected.
    - This option is not available when the **Width Mode** is **Not Used**.
- **Text Case** - Specifies the character case of the value. Select:
    - **Not Changed** to keep it as is.
    - **Upper** to force Upper Case.
    - **Lower** to force Lower Case.
- The behaviour settings can be overridden in command blocks.

To re-assign the selected parameter to a format, use the context menu.

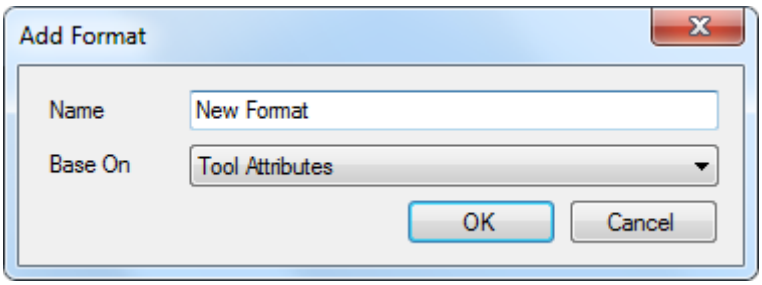
Creating formats

To create a new format:

1. Right-click the **Formats Explorer** and select the **Add Format** menu option. For example:



The **Add Format** is displayed



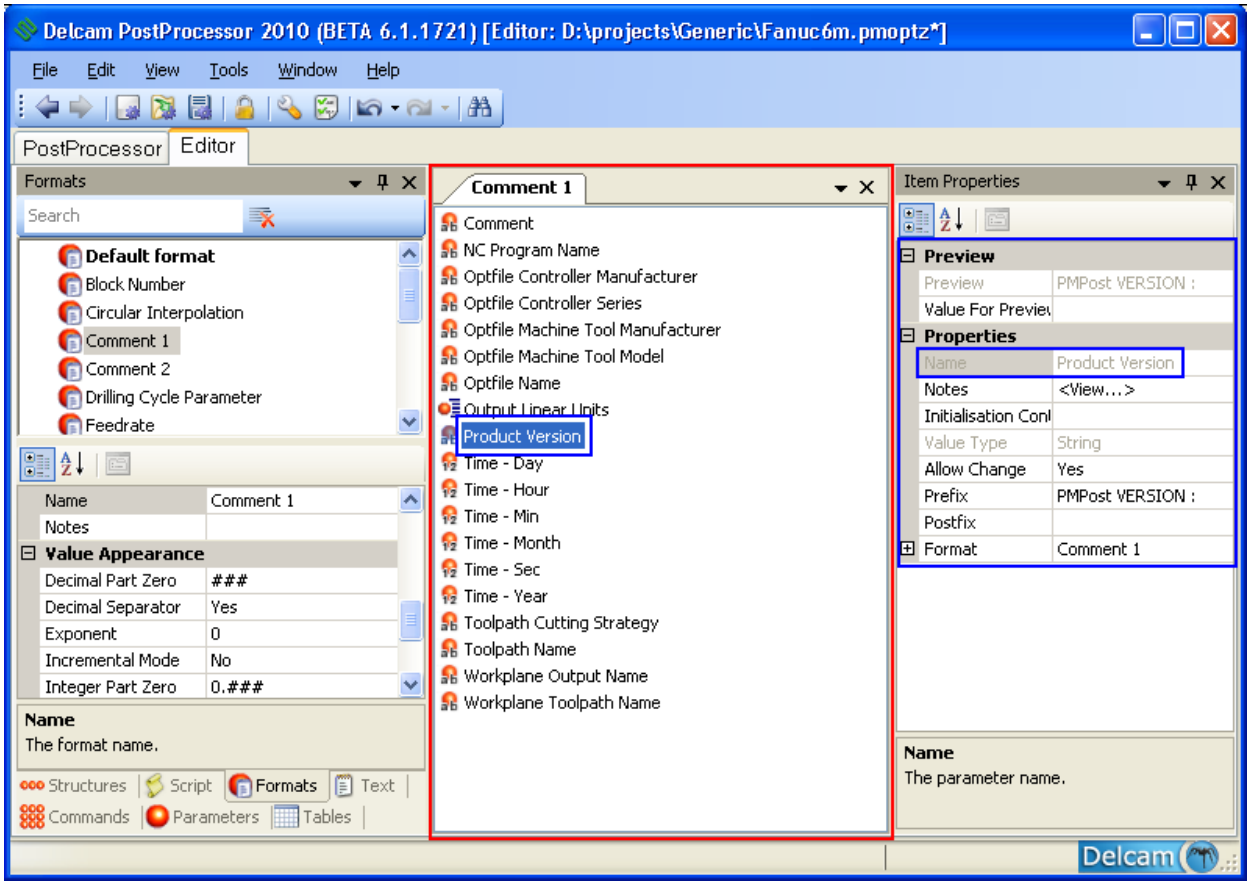
2. Enter a **Name** for the format.

3. In the **Based on** list, select the format on which you want to base the new format's properties, and click **OK**. The new format is displayed in the **Formats** folder. For example:

- To delete a format, select its entry in the **Formats Explorer** and click the **Delete** button. Any parameters that use the deleted format are re-assigned to **Default format**.

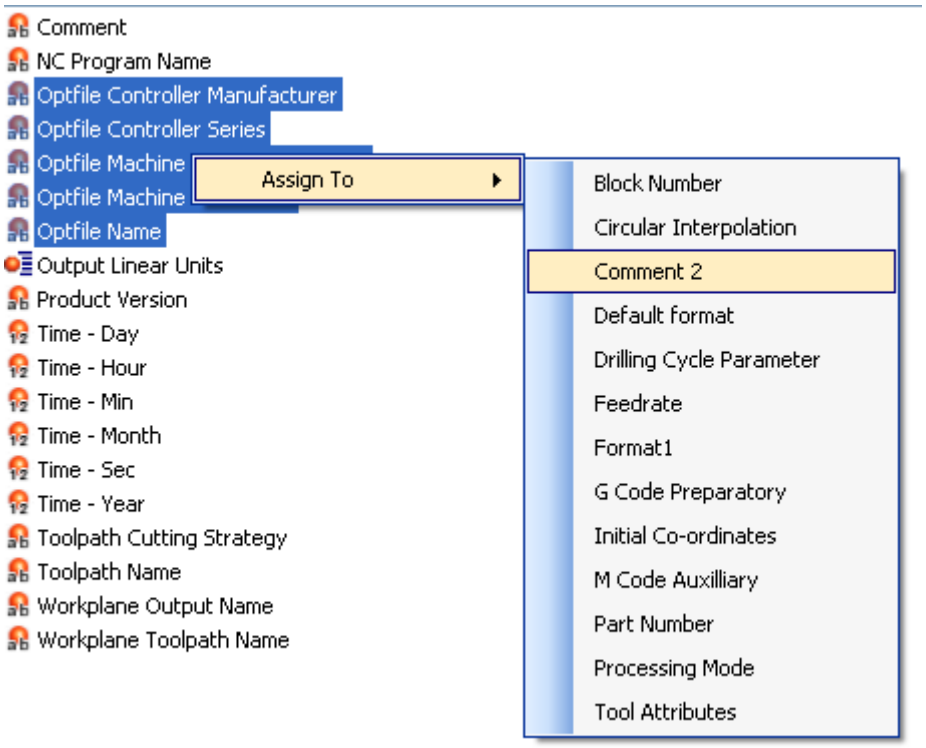
Assigning parameters and properties

The **View** pane summarizes the parameters associated with the selected format. For example:



To view and edit the properties of a parameter, select its entry in the list. The properties are displayed in the **Item Properties** window.

To assign a parameter to another format, right-click the parameter and select the format from the menu. For example:



The selected parameter is changed to chosen format.

A parameter cannot be assigned to more than one format. If you assign a parameter that is already assigned, it is removed from its original format.

Script

Postprocessessing behaviour may be extended by using JScript or VBScript. User-defined functions have access to built-in functions and all option file settings. The toolbar allows you to create script functions quickly. Refer to Command Interface for instructions on how to associate a script function with a command.

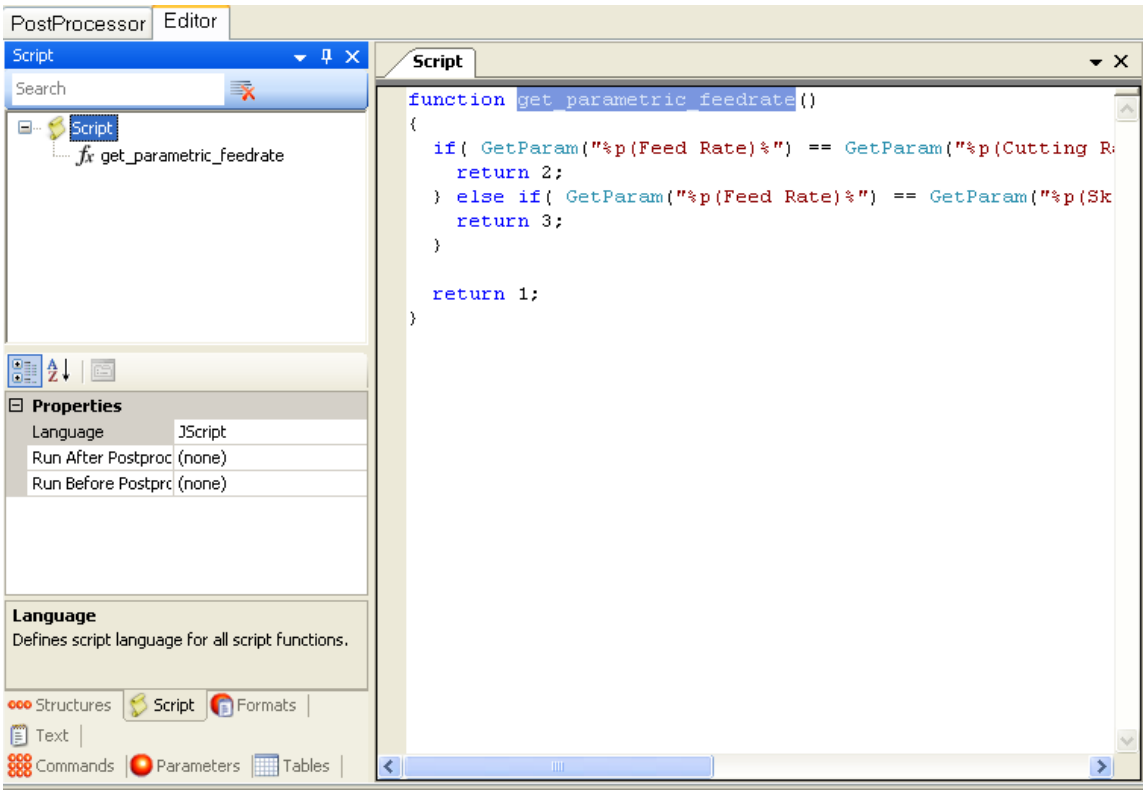
When **Delcam PostProcessor** processes a command, it responds as follows:

- If the command block is not associated with a script function, **Delcam PostProcessor** processes the list of block items in the command and generates the NC output. This default behaviour is equivalent to a script's **StandardResponse()**.
- If the command block is associated with a script function, the script is run. If the script function contains a **StandardResponse()**, **Delcam PostProcessor** generates output from the command block as above. If the script function contains an **AdvancedResponse()**, **Delcam PostProcessor** generates output in accordance with the script associated with the command referenced by the **AdvancedResponse()**.

Creating a script

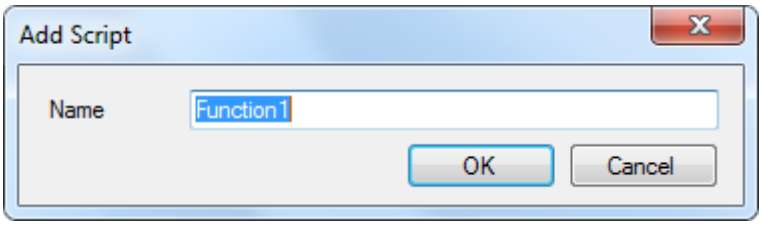
To create a script:

1. Select the **Script** tab in the **Explorer**.



2. In the **Properties** bar, select the **Language** you want to use.

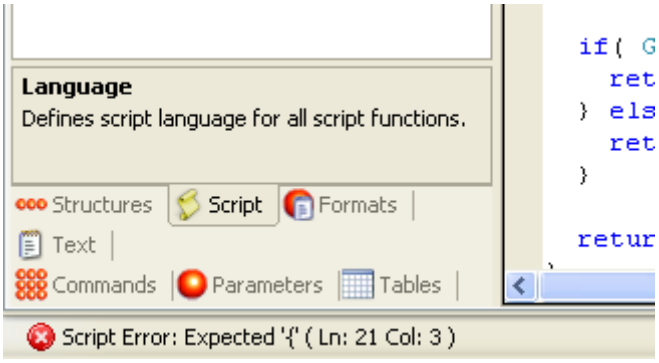
3. Click on the **Script Functions** toolbar. The **Add Script** dialog is displayed.



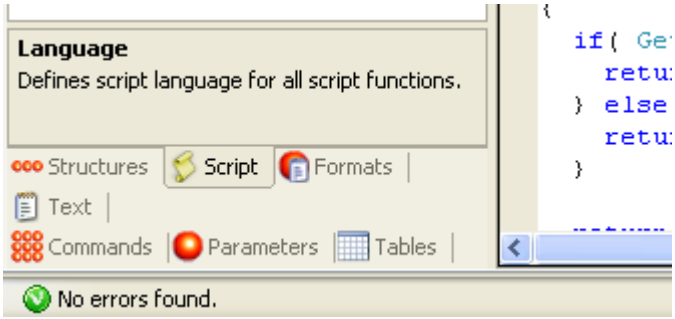
4. Enter a **Name** for the function. The name must not start with a digit or include spaces.

5. Click **OK**. The new script is added to the **Script** folder in the **Explorer**, and the function code is shown in the **Script** view.

6. Enter your script. The code is checked as you write it, and its status, and any errors, are displayed in the **Status Bar**. For example:



If no errors are found, the following message is displayed:



- To add a function to the script, use the **Script Functions** toolbar.

7. Select the command to which you want to add the script, and attach it.

8. Repeat step 7 to add the function to other commands.

Renaming a script function

To rename a function and update all references to it in the option file:

1. Right-click the function in the **Script Functions** tree and select the **Rename** menu option.

2. Type the new name in the background box that encloses the function.

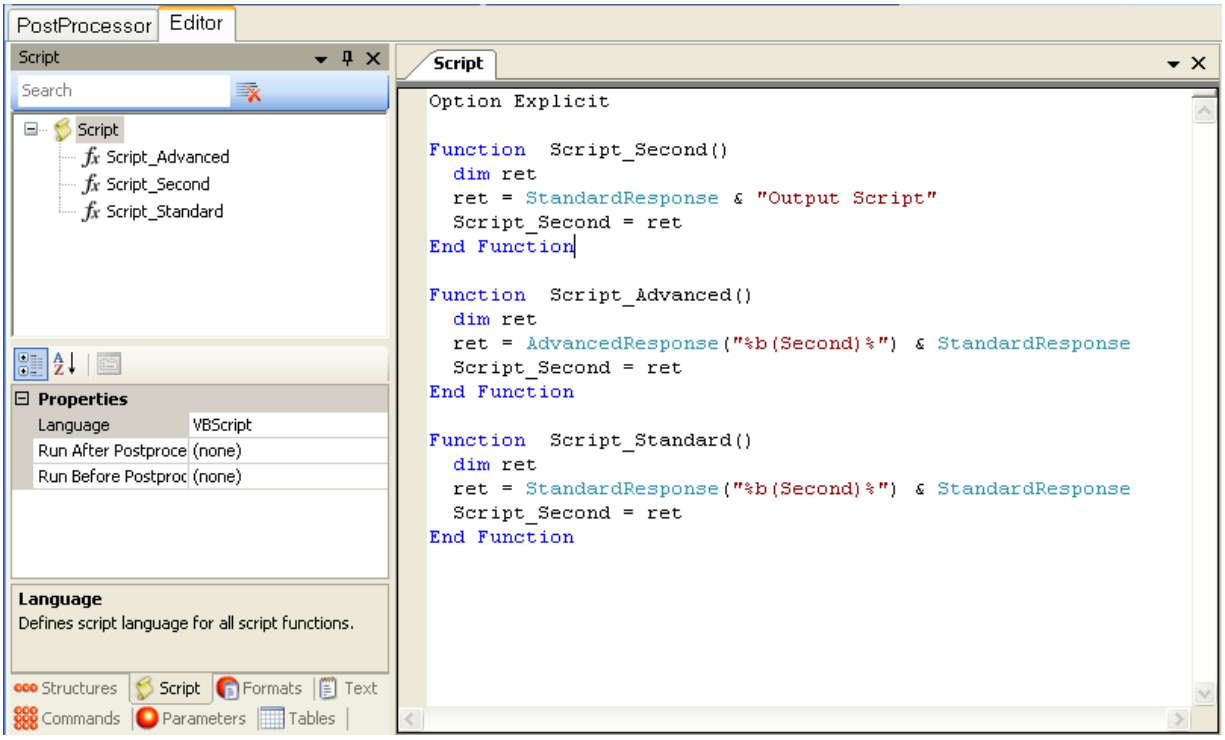
Delcam PostProcessor updates the option file and informs you which commands have been updated.

3. Click **OK** to clear the message.

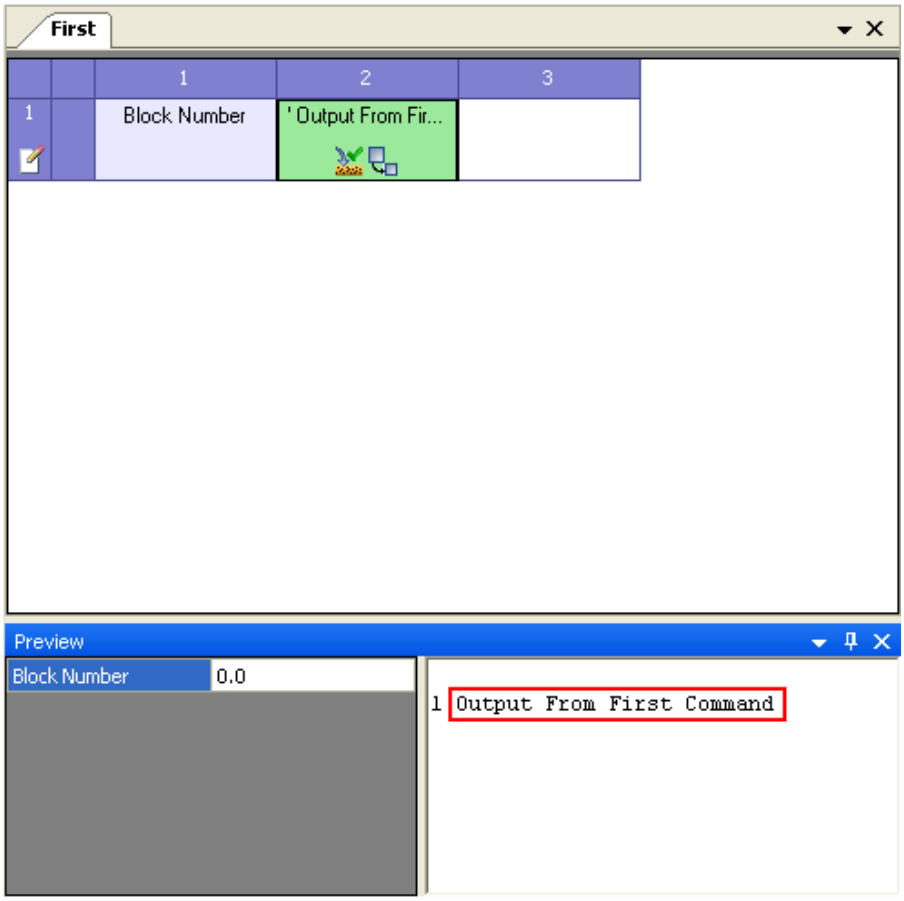
Examples of scripting principles

These examples have been simplified to illustrate what happens when a script function is attached to a command, and what happens when the script contains **Standard** and **AdvancedResponse** functions.

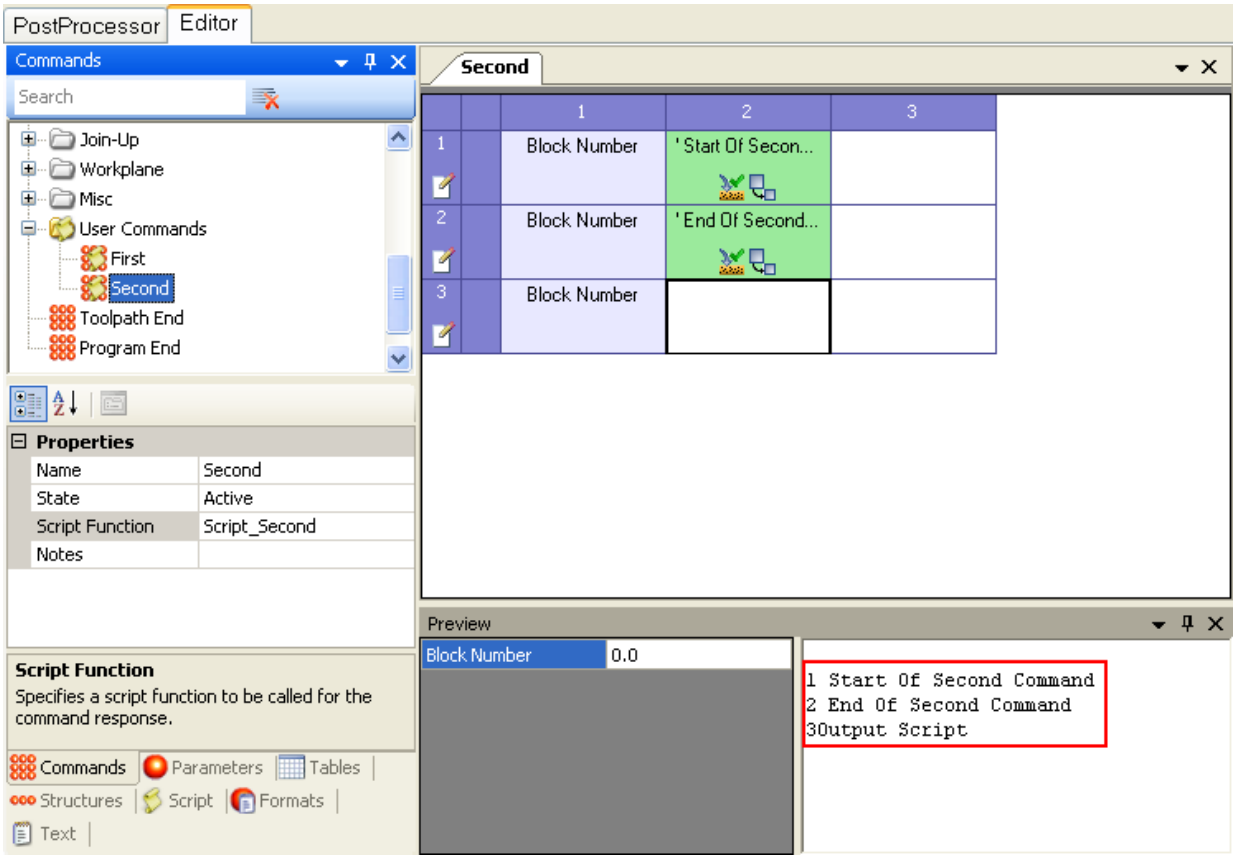
The following script functions generate simple text:



When **Delcam PostProcessor** processes the first command, which has no attached script, it processes the block items in the command and generates the NC output. This is the default behaviour and is equivalent to a script's **StandardResponse()**:



When **Delcam PostProcessor** processes the second command, the function ignores the standard output from the command block and uses the output from the script:

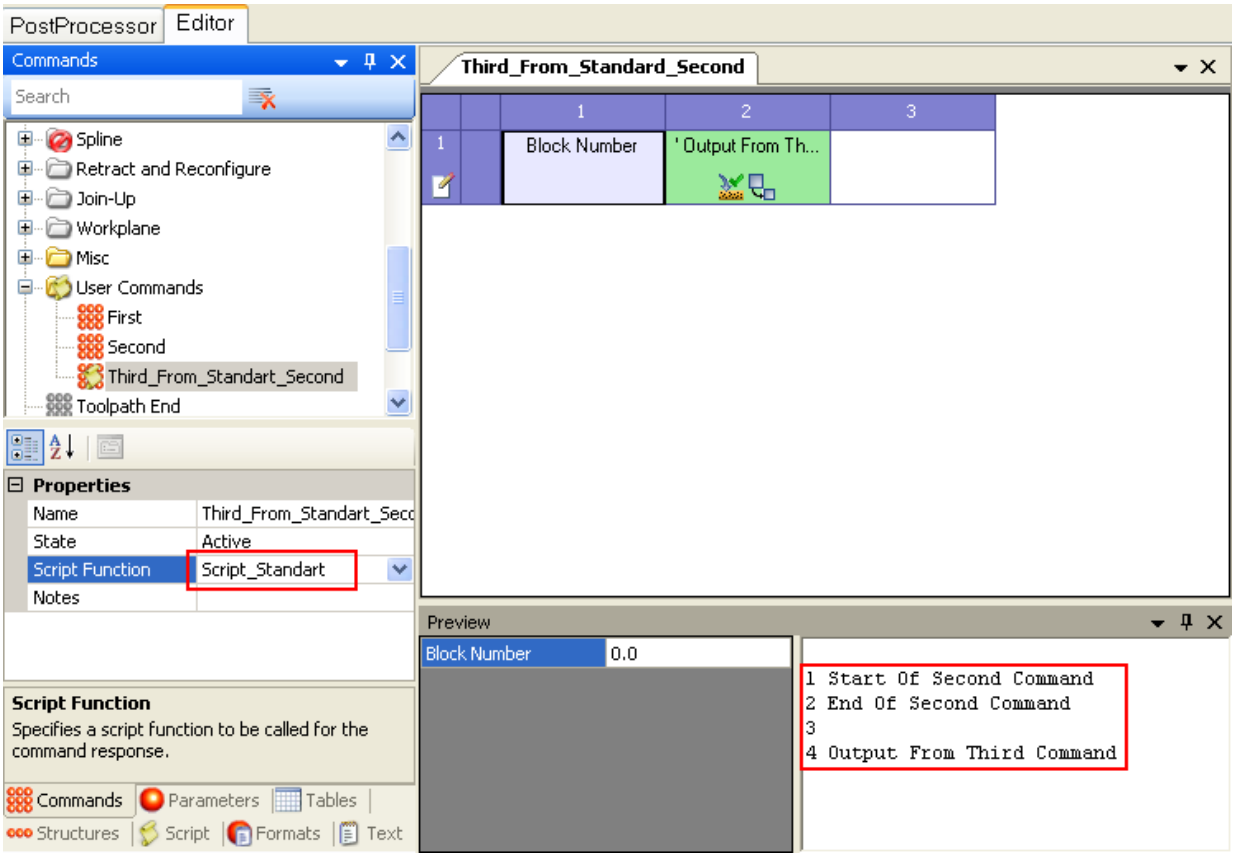


The script for the attached function, **Script_Second**, is:

```
function Script_Second()
  dim ret
  ret = StandardResponse & "Output Script"
  Script_Second = ret
end function
```

Because the first part of the value returned by function **Script_Second** contains a **StandardResponse()** with no argument, the first part of the output is the same as that of the command block (**Start of Second Command, End of Second Command**). The last part of the value returned (**Output Script**) corresponds to the last part of the script.

The command **Third_from_Standard_Second** uses a script function (**Script_Standard**) instead of its command block to generate its output:



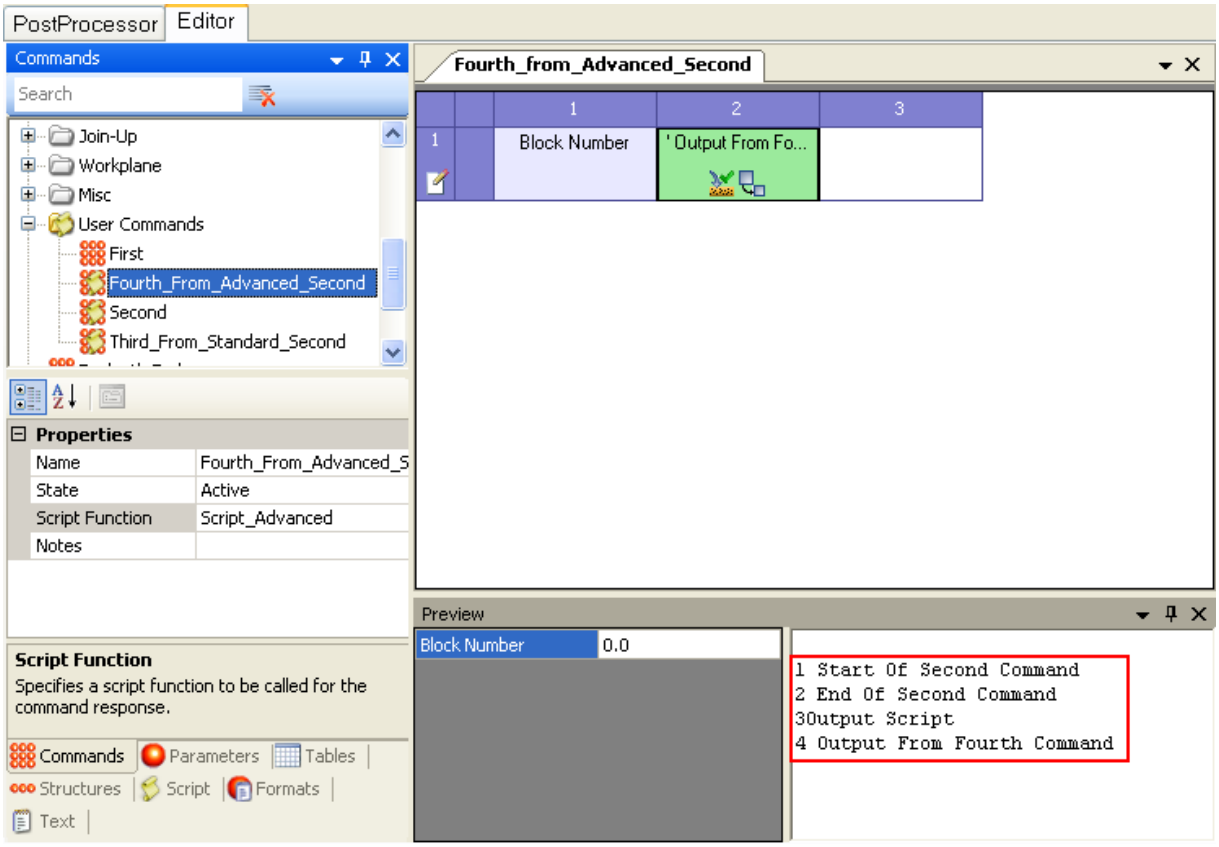
The script for the attached function, **Script_Standard**, is:

```
function Script_Standard()
  dim ret
  ret = StandardResponse("%b(Second)%") & StandardResponse
  Script_Standard = ret
end function
```

The first part of the output is from the block command output of the second command. The last part of the output (**Output from Third Command**) is derived from the current command block (**Third_from_Standard_Second**) because the

last part of the script function is **StandardResponse()** without an argument. This corresponds to the output from the calling command.

The command **Fourth_from_Advanced_Second** also uses a script function, **Script_Advanced**, instead of its command block to generate its output. However, the first part of the output is derived from the **AdvancedResponse()** (the scripted output) of the second command:



The script for the attached function, **Script_Advanced**, is:

```
function Script_Advanced()
  dim ret
  ret = AdvancedResponse("%b(Second)%") & StandardResponse
  Script_Advanced = ret
end function
```

Because the first part of the value returned by the function contains an **AdvancedResponse()** with an argument referencing the second command, the first part of the output is derived from the script attached to the command block for **Second**, that is, from **Script_Second** (**Start of Second Command, End of Second Command, Output Script**). Because the last part of the value returned contains a **StandardResponse()** with no argument, the last part of the output (**Output from Fourth Command**) is taken from the current command block.

Using the Script toolbar

The **Script** toolbar is displayed when you select a script in the **View** pane. It enables you to create function templates and perform **Cut**, **Copy**, **Paste**, **Delete**, **Select All** and **Find** functions on the script text.

Button	Description
	Adds atemplate for a new function to the end of the script.
	Click the down arrow and select an option from the list to add the template for the selected function to the script. You can also select a function by right-clicking the **Script** view and selecting an option from the context menu.
	Cuts the selected script fragment to the system clipboard.
	Copies the selected script fragment to the system clipboard.
	Pastes the contents of the system clipboard into the **Script** view.
	Deletes the selected script fragment.
	Displays the **Find toolbar**. Use it to search the script text and to control how the text is displayed.

Debugging script

Finding run-time problems is very difficult in complex script. But option file developers may setup breakpoints and execute script in "step" mode.

At first, you need to have installed any JIT debugger. The best one is shipped with Microsoft Visual Studio (including express editions), but you may install free Script Debugger for Windows (http://www.microsoft.com/downloads/details.aspx?FamilyID=2f465be0-94fd-4569-b3c4-dffdf19ccd99&DisplayLang=en).

Then, you need to allow script debugging. Create file called "TurnJITDebuggingOn.reg" with content:

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Windows Script\Settings]
"JITDebug"=dword:10000000
```

Now when you double clicked on it, will be applied changes into windows registry and debugging will be enabled.

Then, add "debugger" call to any required place in advanced function.

```
function complex_method()
{
  var res = "";

  // Some script code

  debugger; // Here script executing will be stopped

  // Continue script code

  res += StandardResponse();

  return res;
}
```

Then, start postprocessing. "debugger" will break executing your script function, and you will be able to debug. Debugger allows to see values of parameters and evaluate expressions. Also you may continue script execution step by step or to the end of script.

To disable JIT debugging you need create and apply file called "TurnJITDebuggingOff.reg" with content:

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Windows Script\Settings]
"JITDebug"=dword:00000000
```

Using the PostComparer

To select option files you want to compare:

1. Click the **Open Left File** button or select the **File > Open Left File** menu option.
2. In the **Open** dialog, select the option (***.opt**) file you want to display on the left of the **PostComparer** window, and click **Open**.
3. Click the **Open Right File** button or select the **File > Open Right File** menu option.
4. In the **Open** dialog, select the option (***.opt**) file you want to display on the right of the **PostComparer** window, and click **Open**.
5. You can now compare the option files.

- You can open recently viewed files by selecting the **File > Recent Left** and **File > Recent Right** menu options, or by clicking the **Recent Files** buttons above each pane.
- To swap the positions of the option files in the window, click the **Swap Files** button or select the **File > Swap Files** menu option.

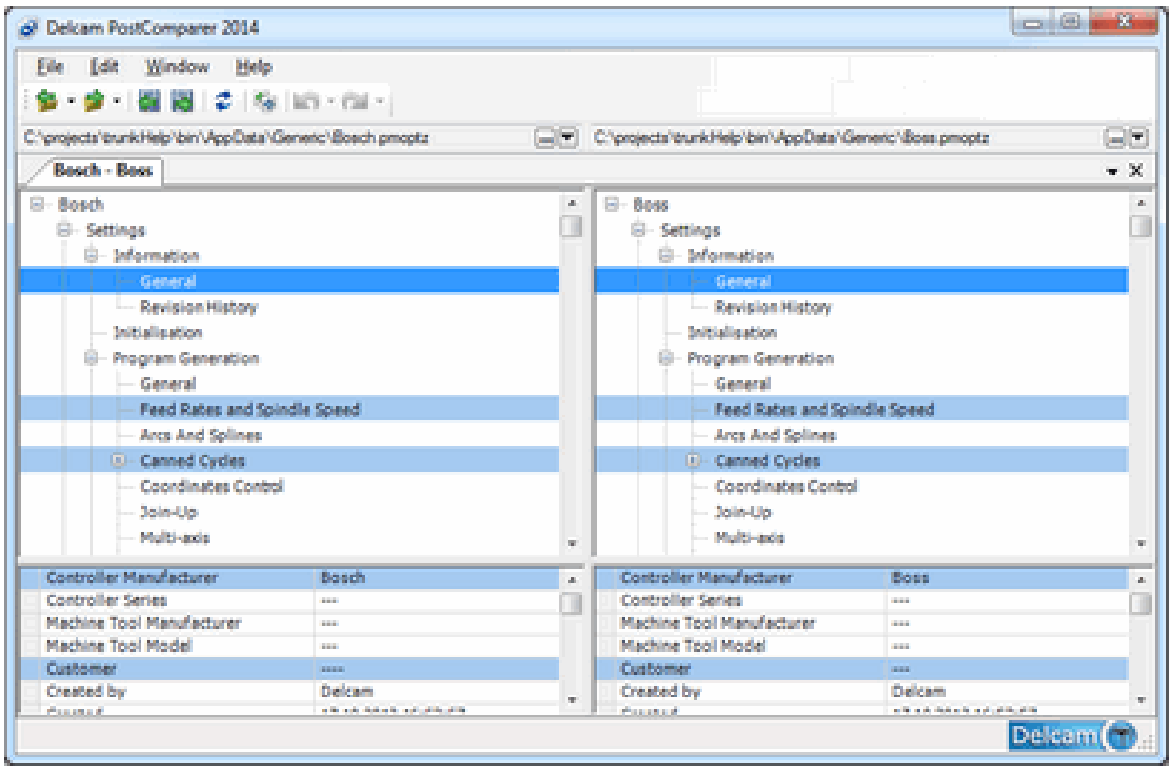
To compare the selected option files, click the **Compare** button or select the **File > Compare** menu option

View Differences

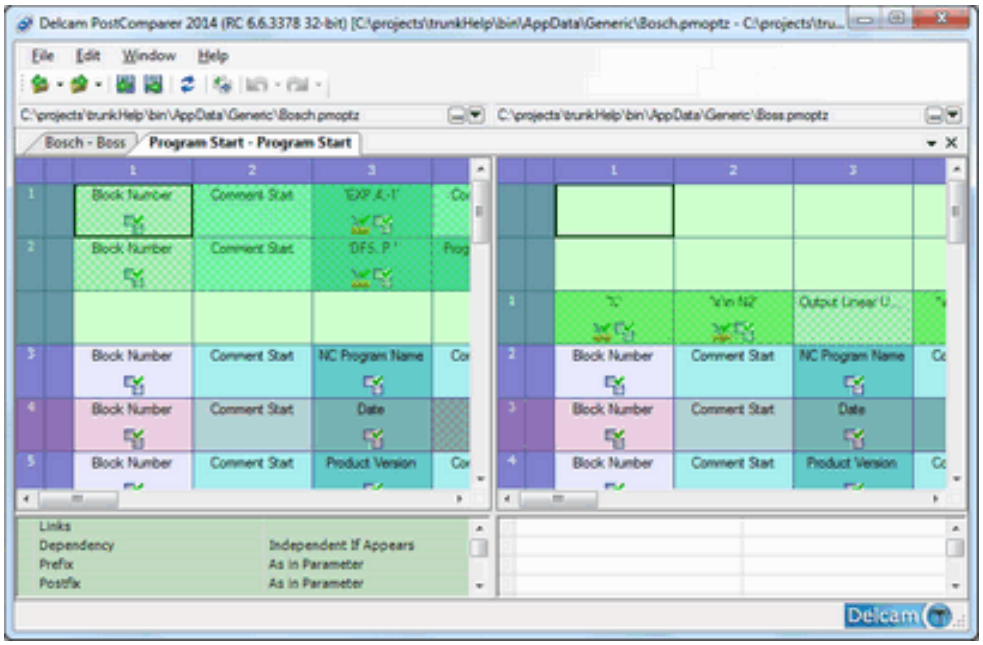
When the comparison is complete, each element is displayed with a background colour that indicates how it compares across the option files:

- White indicates the element is the same in both files.
- Pink indicates the element is different in each file.
- Green indicates the element appears in only one file.

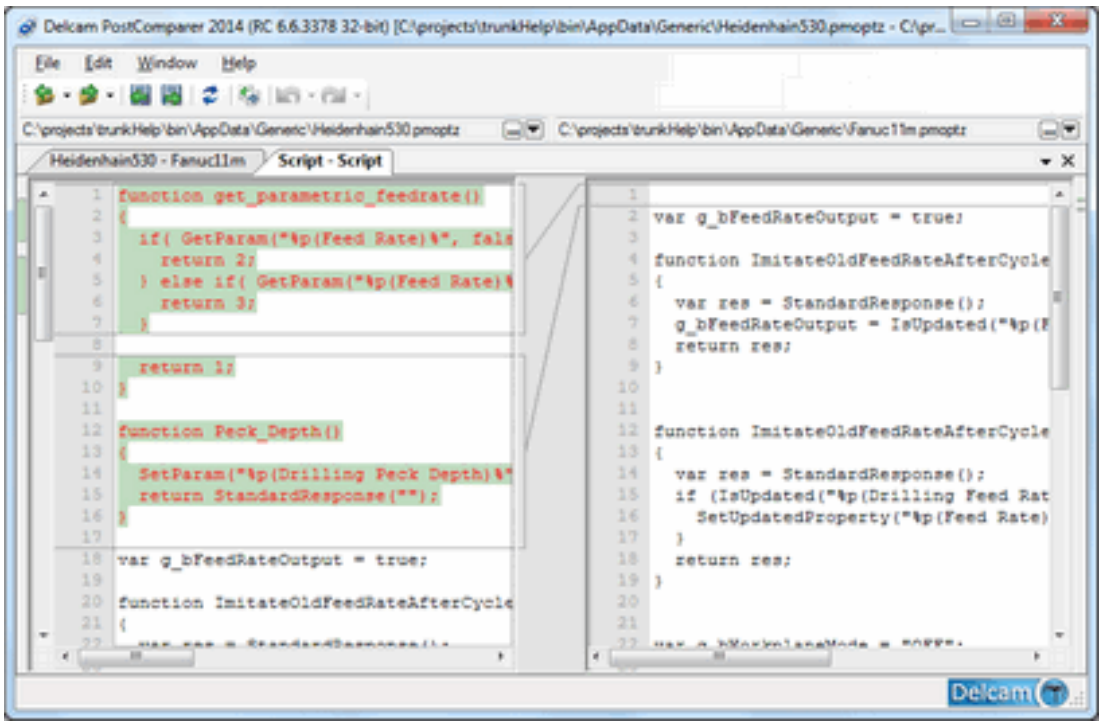
For example:



To compare a command, double-click its entry in the **PostComparer**. The command is displayed in a separate window with the differences highlighted. For example:

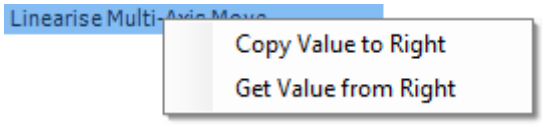


To compare a script, double-click its entry in the **PostComparer**. The text of the script is displayed in a separate window with the differences highlighted. For example:



Merge Features

Using context menu you may copy values of highlighted items between option files:



For complicated items, like commands, copying of whole item is not allowed. But you may copy all its properties separatelly, step by step.

PostComparer support undo/redo functionality.

All changes are made in the editor and eventually you need to save changes using toolbar buttons and .

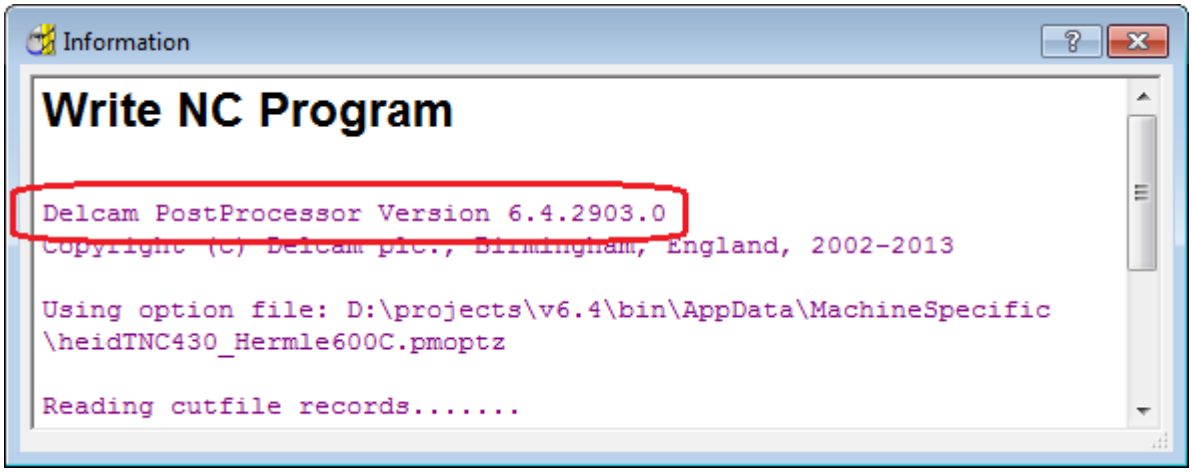
- When you close **PostComparer** with changed and unsaved option file, you will be prompted about saving it.
- Option file will be saved in the current file format and cannot be used with old versions of **PostProccesor**.

F.A.Q.

- **Which version will be used for postprocessing?**

Which version will be used for postprocessing?

CAM systems always uses only one **active** postprocessor. Its version will be printed in output window of CAM system.



For changing active version, please, use the guide Change active version

If you have individual CLDATA files you may call any version using Windows start menu - **All Programs** - **Delcam** - **PostProcessor** - **PostProcessor**.

How to

- **Change active version**

Change active version

1. In the Windows start menu - **All Programs** - **Delcam** - **PostProcessor**.
2. Choose desired version and run **Make PostProcessor the Current Version**.

This shortcut will start installer which sets active console and COM-server.

- On Windows 64-bit edition you may set two active versions, first for 32-bit and second for 64-bit applications.

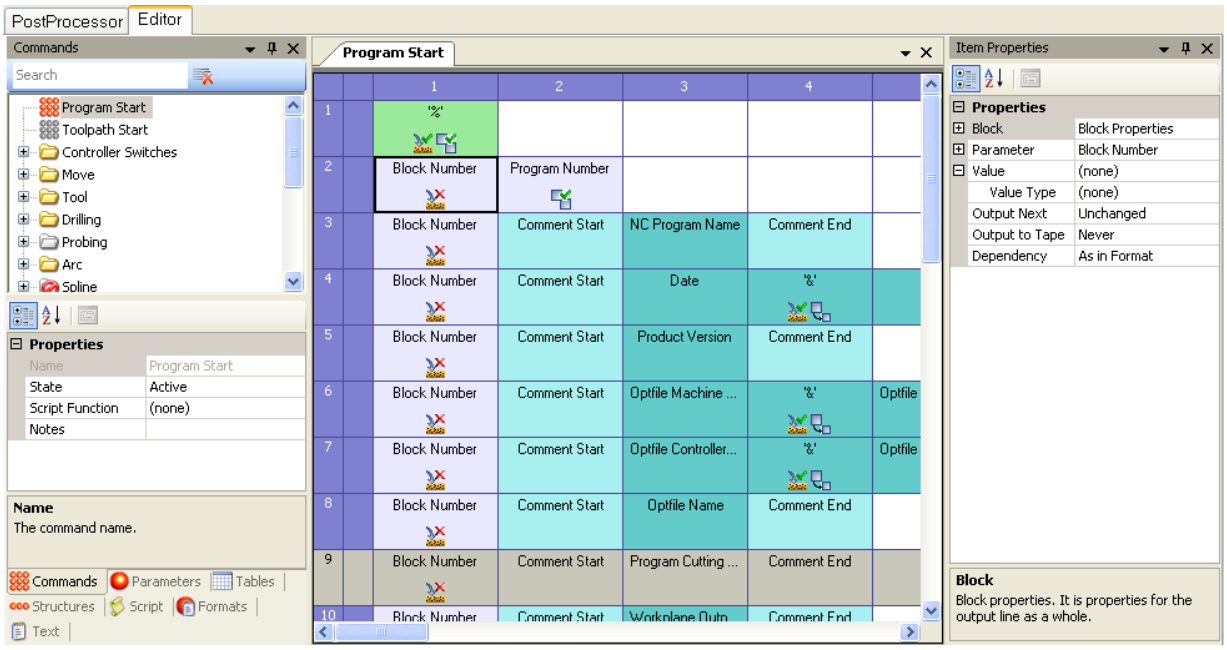
References

Command reference

This section lists the standard commands available with **Delcam PostProcessor**.

Program start

Program Start sets the initial values the NC machine needs to start running. The command is output at the beginning of every program, and must be activated and defined in each option file.



You can use the **Program Start** command to specify:

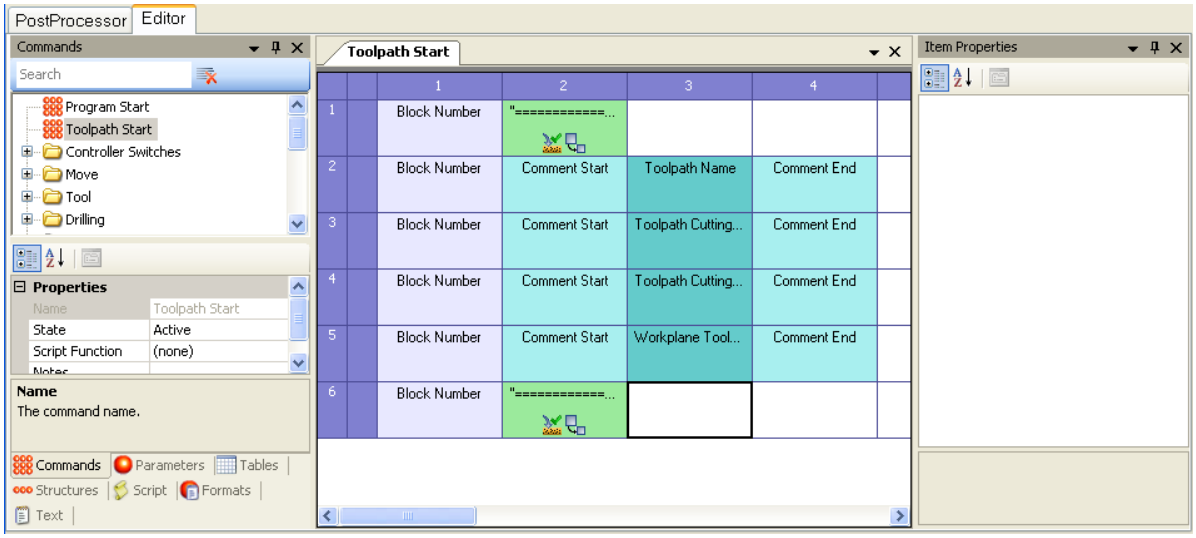
- Any special characters required by the machine controller. For example, Fanuc controllers require programs to start with the **%** character.
- Parameters that contain information about the program, such as the name of the NC output program, and the output units.
- Required home positions that are relative to the machine datum.

You can also use the command to output operator information, such as the tools and cutters required by the program.

You can control the way in which items are output to the NC program using tables.

Toolpath start

Toolpath Start initializes parameters before the start of every toolpath in the option file. It can also be used to output values and Comments.



You can also use the **Toolpath Start** command to output a list of tools using a **table**.

Controller switches

Cutter compensation on

Activate this command to switch on the **Cutter Compensation Mode** parameter, which specifies the offset from the cutter centre.

The **Cutter Compensation Mode** parameter can be embedded within other commands, so this command is not normally used.

Cutter compensation off

Activate this command to switch off the **Cutter Compensation Mode** parameter, which specifies the offset from the cutter centre.

The **Cutter Compensation Mode** parameter can be embedded within other commands, so this command is not normally used.

Coolant on

Activate this command to switch on the **Coolant Mode** parameter.

- The **Coolant Mode** parameter can be embedded within other commands, so this command is not normally used.

Coolant off

Activate this command to switch off the **Coolant Mode** parameter.

- The **Coolant Mode** parameter can be embedded within other commands, so this command is not normally used.

Set Multi-Axis On

The **Set Multi-Axis On** command is used by 5-axis toolpaths. It is also used by 3+2 toolpaths when workplanes are disabled or when the controller does not provide 3+2 support. Use the command to specify any conditions required only for multi-axis machining.

You must select a multi-axis machine profile in the **Coordinates Control** settings when processing multi-axis toolpaths.

Set Multi-Axis Off

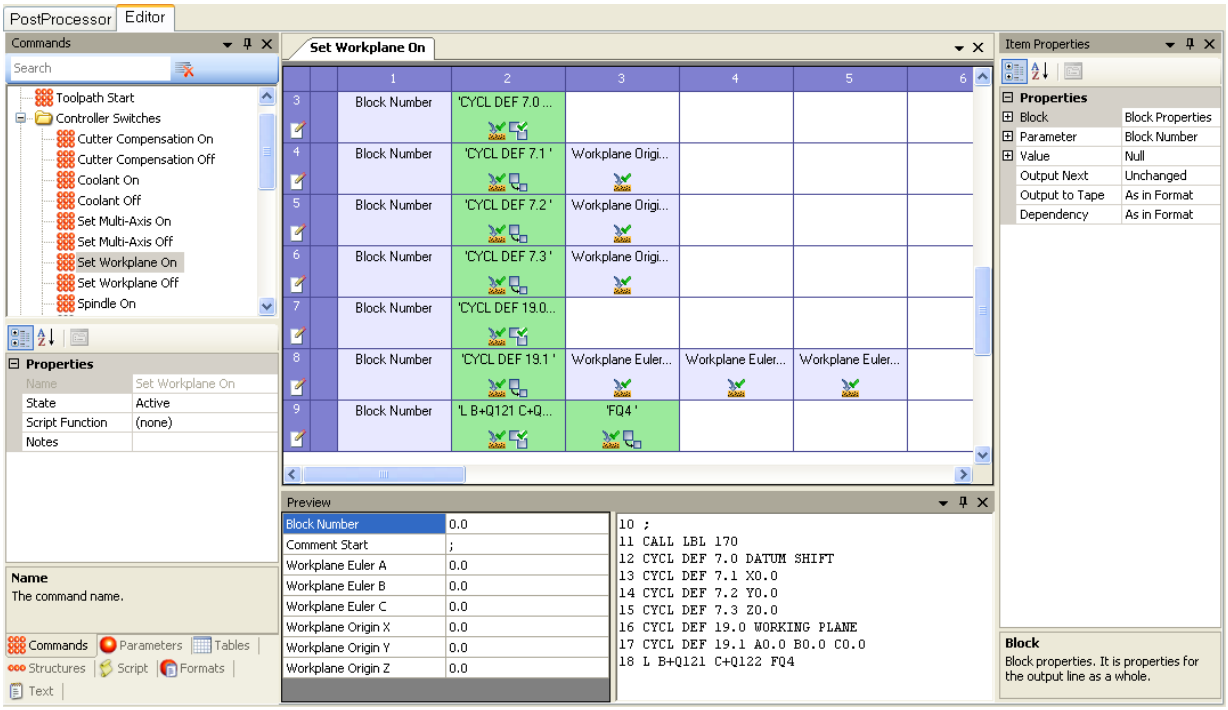
Use the **Set Multi-Axis Off** command to cancel any parameter settings specified for multi-axis working.

Set Workplane On

When the machine controller supports working in a local workplane, and you have specified a **Profile** of **Multiaxis machine with RTCP and 3+2 support** in **Coordinates Control**, you must activate the **Set Workplane On** command and specify the parameters required by the **Workplane Definition Method** used by the controller:

- If the controller requires machine angle coordinates to specify the tool location, you must set the **Workplane Origin** parameters and the azimuth and elevation values using machine angle parameters, such as **WP Machine B** and **WP Machine C**.
- If the controller requires Euler angles to specify the tool location, you must specify the **Workplane Origin** parameters and the **Workplane Euler** parameters.
  - The values of the **Workplane Euler** parameters depend on the **Euler Convention** specified in **Multi-Axis Configuration** settings.

You must select the correct **Euler Convention** for the target machine controller otherwise the values output to the NC program will be incorrect.



This example shows a **Set Workplace On** command for a Heidenhain controller. It uses Euler angles to specify the tool location.

Set Workplane Off

Use the **Set Workplane Off** command to specify the parameters required to cancel the local workplane specified in the **Set Workplane On** command so that the controller can revert to using the output workplane.

Spindle on

Activate this command to switch on the **Spindle Direction Mode** parameter.

The **Spindle Direction Mode** parameter can be embedded within other commands, so this command is not normally used.

Spindle off

Activate this command to switch off the **Spindle Direction Mode** parameter.

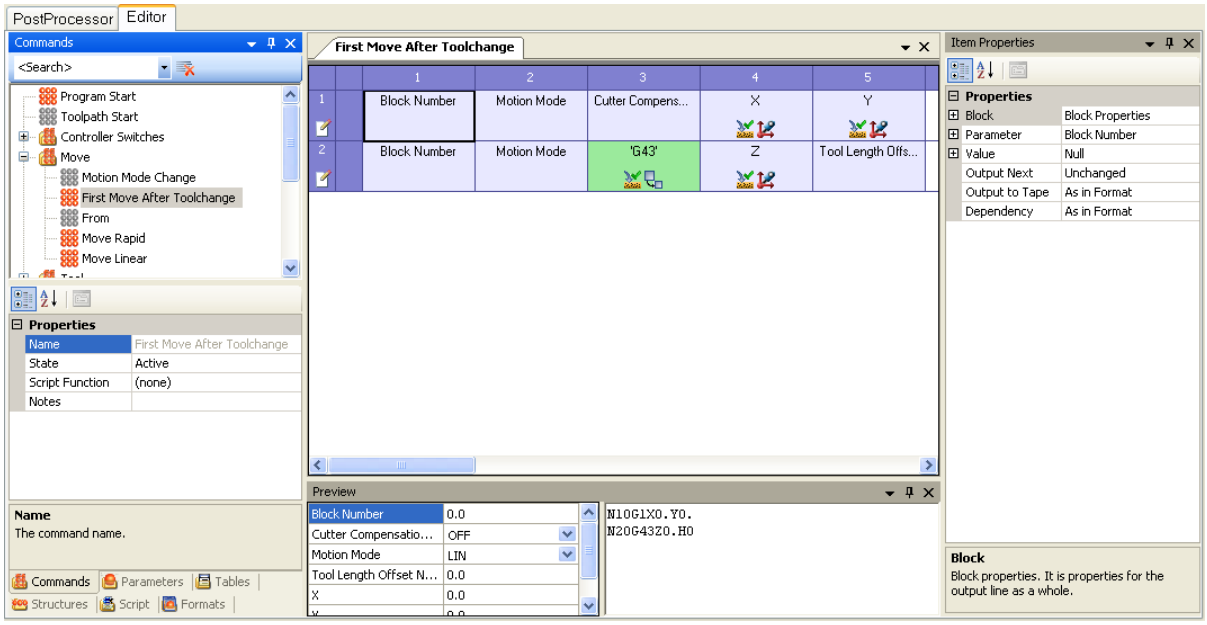
The **Spindle Direction Mode** parameter can be embedded within other commands, so this command is not normally used.

Move

First Move After Tool Change

The **First Move After Tool Change** specifies the initial movement within a toolpath following the loading of a tool. It typically sets the parameters that control the movement from the tool start position to the point where the tool begins work.

Example **First Move After Toolchange** command for a Fanuc controller:



For an explanation of commands, see Command Interface.

First Move After Tool Change enables you to specify special conditions for the initial movement of a tool. For example, in a 3-axis toolpath, the X and Y values can be specified in a separate block to the Z value so that the tool does not move in all three axes simultaneously. This enables you to avoid potential problems when moving the tool from its loading position.

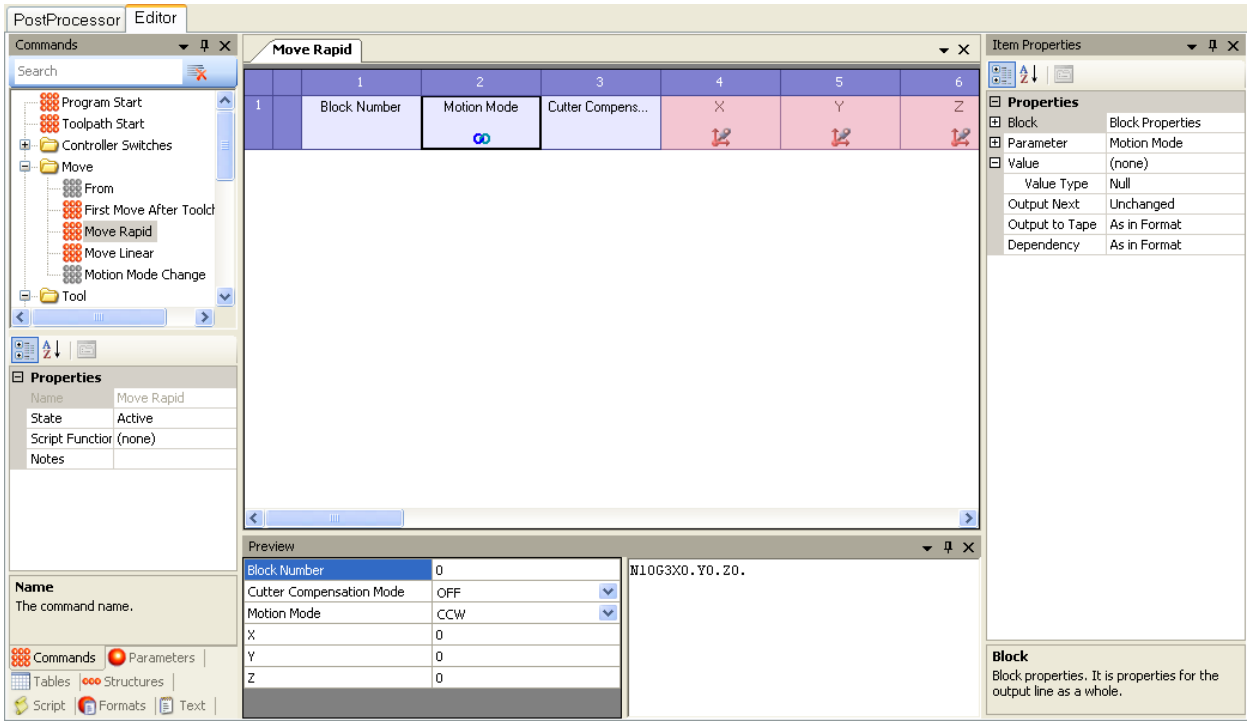
The command is also used to specify all required parameters that are not set in the **Initialisation Table** or **Program Start** command including:

- **Spindle Direction Mode** - This parameter must be specified before the first move so the mode can be set from the CLDATA file.
- **Coolant Mode** - This parameter must be specified after the spindle has been switched on and achieved the specified rotation rate, and before beginning a work movement. The **Mode** is then set from the CLDATA file.
    - On many machines, the spindle and coolant cannot be switched on simultaneously. To avoid this problem, specify the **Spindle Direction Mode** and **Coolant Mode** parameters in separate blocks.
- **Tool Length Compensation Mode On** - Set this parameter after a tool has left safe Z so the machine knows the tool length before it starts work.
- **Cutter Compensation Mode** - This parameter must be specified before beginning a work movement so the mode can be set from the CLDATA file.

You must define a value for each state in a group parameter before using it; nothing is written to the output file when a state is unset. To check the options for a group parameter, select the block item. The parameter properties is displayed in the **Item Properties**.

Move rapid

The **Move Rapid**command controls movements of the tool when it does not need to perform any cutting. Typical parameters used in this command include:



In this example, the output of the **Motion Mode** is required only when at least one of the X, Y or Z values is output. For more information, see **Linking block items**.

Linking enables you to make the output of one block item dependent on the output of one or more other items in the same block.

To create a link between block items:

1. Right-click the block item you want to link and select:

Positive if you want to output the block item only when the linked item is output.

Negative if you want to output the block item only when the linked item is not output.

The cursor changes to indicate the type of link you selected.

2. Click the block items you want to link to. A link icon is displayed in the first block item to show it is linked.

3. Right-click to complete the link.

To display the block items in a link at any time, click the link icon.

To break a link:

4. Right-click the block item you want to unlink and select **Unlink**.

5. Click the block items you want to unlink. The colour of the block item changes to show it is no longer linked.

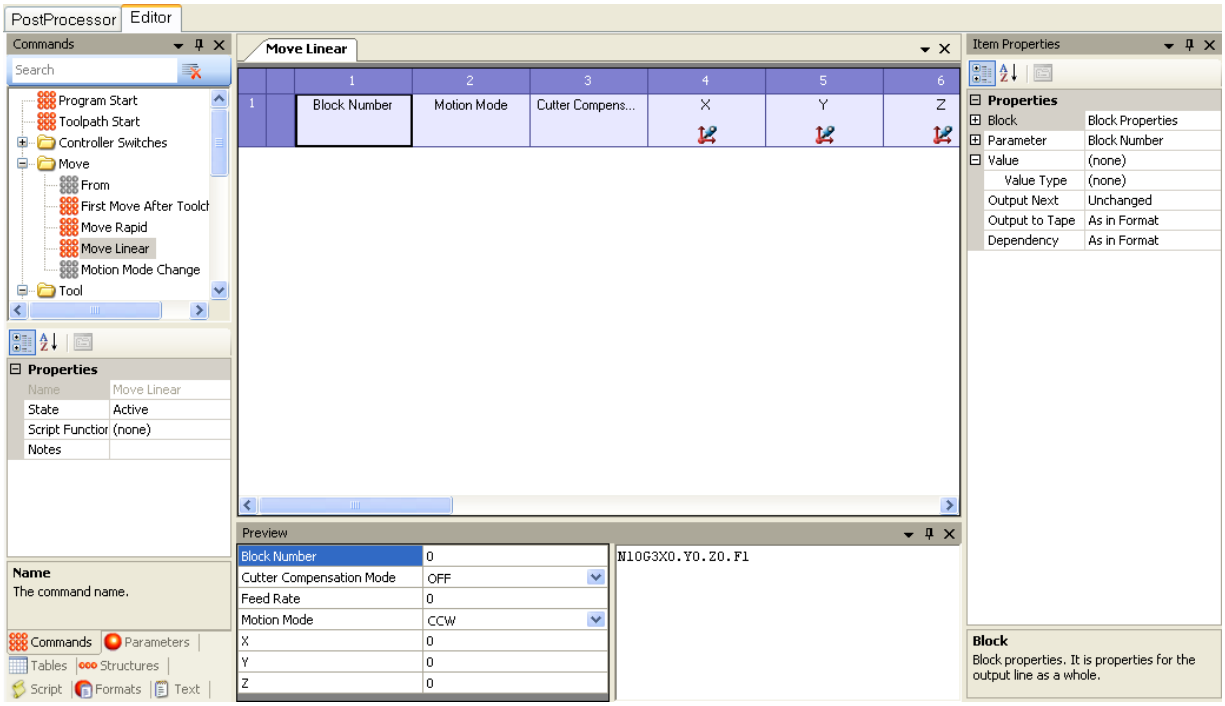
6. Right-click to stop unlinking.

The link icon is removed when all the block items are removed from the link.

- **Motion Mode** - This parameter enables the motion mode to be set from the CLDATA file.
- **Cutter Compensation Mode** - This parameter must be specified before beginning a move so the mode can be set from the CLDATA file.
- The point coordinates for each axis.
- **Tool Number** - The number is set from the corresponding tag in CLDATA. The prefix for tool number options is typically **T**.
- **Tool Length Offset Number** - The number is set from the corresponding tag in CLDATA. The prefix for tool length offset number options is typically **H**.
  - On most machines, the **Tool Number** parameter automatically sets the **Tool Length Offset Number** to the same value.
- **Coolant Mode** - This parameter must be specified after the spindle is switched on and has achieved the specified rotation rate, and before beginning a work movement. The **Mode** is then set from the CLDATA file.
- You must define a value for each state in a group parameter before using it; nothing is written to the output file when a state is unset. To check the options for a group parameter, select the block item. The parameter properties is displayed in the **Item Properties**.
- **Move Rapid** is never triggered by multi-axis toolpaths because the CLDATA file does not contain rapid moves.

Move linear

The **Move Linear** command controls movements of the tool. Typical parameters used in this command include:



- **Motion Mode** - This parameter enables the motion mode to be set from the CLDATA file.
- **Cutter Compensation Mode** - This parameter must be specified before beginning a move so the mode can be set from the CLDATA file.
- The point coordinates for each axis.
- **Feed Rate** - This parameter enables the feed rate to be set from the CLDATA file.

Motion mode change

Use the **Motion Mode Change** command to define the **Motion Mode** parameter for a change from **Move Rapid** to **Move Linear**, or from **Move Linear** to **Move Rapid**.

Rotation On

The **Rotation On** is executed before any sequence of moves, where rotary axes are changed. This command may be used for unlocking rotary axes when machine is support axes clamping.

This command is invoked each time when set of moving rotary axes changes.

Current states of rotary axes are stored in parameters from **Machine A Rotation** to **Machine T Rotation**. Each group parameter has two states:

- **ON** - axis is rotated in current move,
- **OFF** - there is no rotation.
- You may set clamping codes in parameters like **Machine A Rotation** and, using modality, output it in this command.

Rotation On can also be used for triggering inverse time feeds. The inverse time mode makes more sense when a rotation occurs. In that case the **Feed Mode** must be INVERSE_TIME and G93 (ISO) must be output in the NC program.

When no rotation occurs it's more convenient to use feed per minute or feed per revolution (G94 and G95 correspondingly). In that case the **Feed Mode** must be set to PER_MINUTE or PER_ROTATION in Rotation Off.

Rotation Off

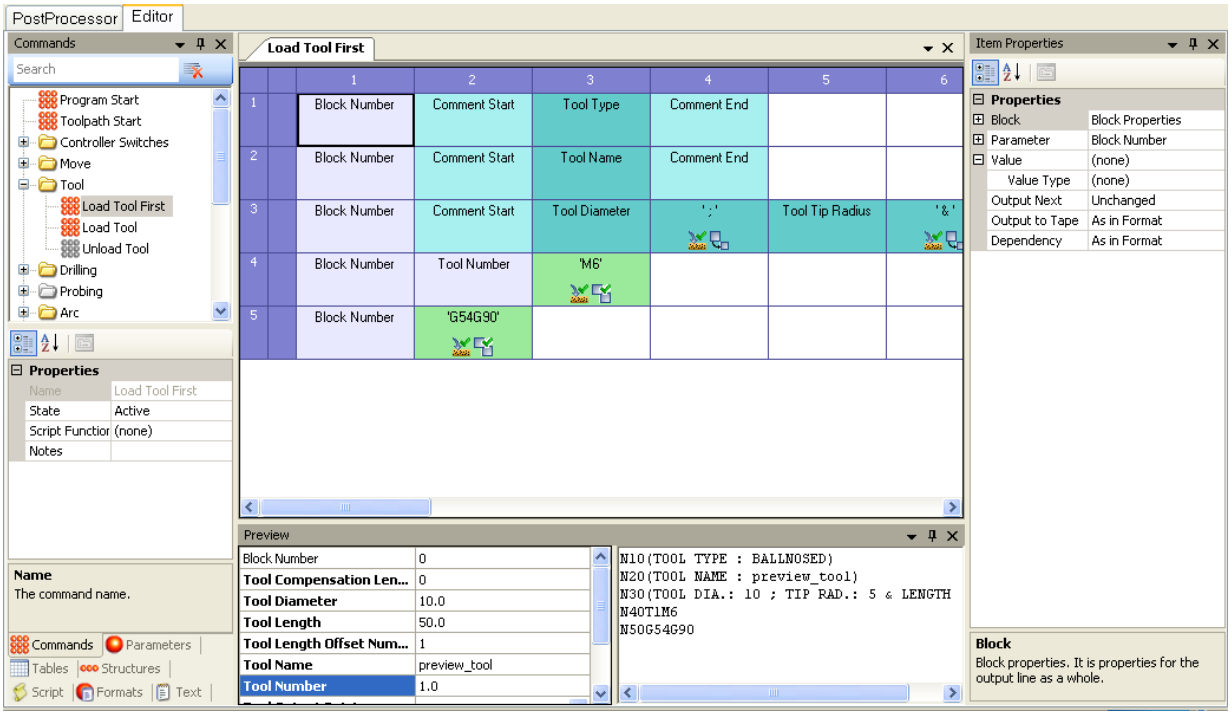
The **Rotation Off** is executed before any sequence of moves, where rotary axes are not changed (if rotary axes changed before). This command is used for unlocking rotary axes when machine supports axes clamping or disable inverse time feed mode.

Tool

Load tool first

The **Load Tool First** command is used the first time a tool is loaded. It normally appears in the program before the first move, and must be defined in each option file.

Example **Load Tool First** command for a Fanuc controller:



For an explanation of the symbols and settings, see Command Interface.

Typical parameters used in this command include:

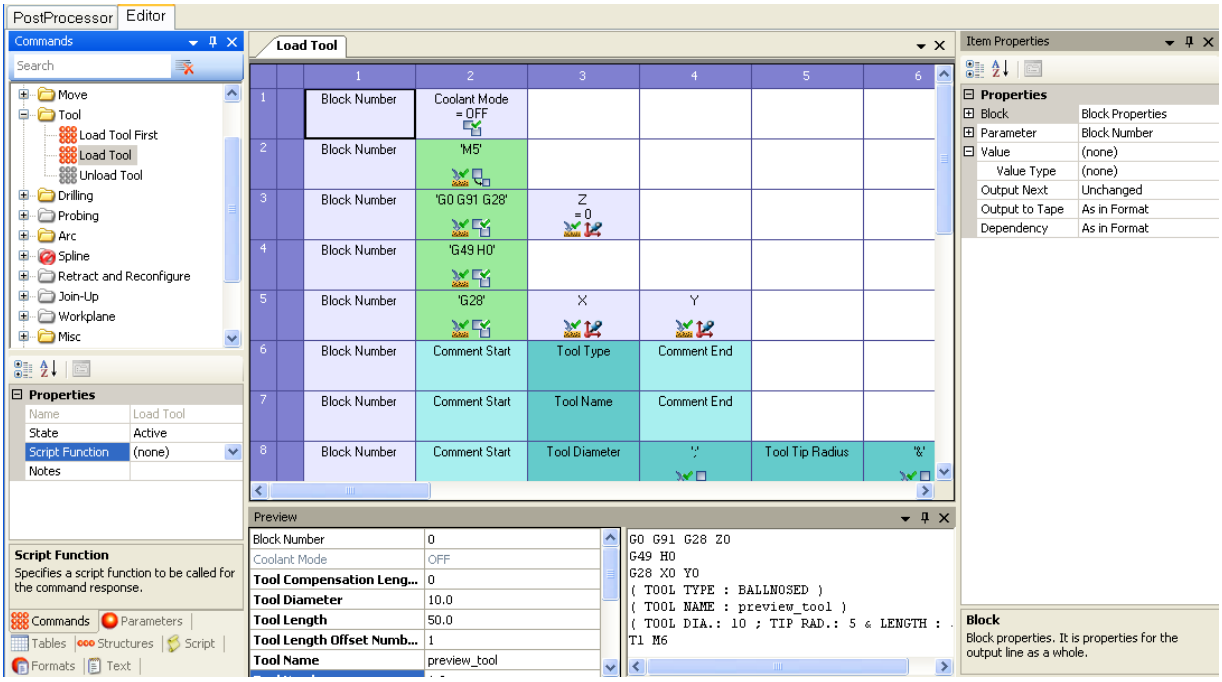
- **Tool Type**
- **Tool Name**
- **Tool Diameter**
- **Tool Tip Radius**
- **Tool Length**
- **Tool Number**

In most cases, the parameter value is set from the CLDATA file. However, you must check the machine prefix specified for each parameter is appropriate for the target controller. (To check a parameter, right-click the block item and select **Edit** from the context menu. The parameter is displayed in the **Add/Edit Parameter** dialog.)

- **Load Tool First** is used only once in the NC program. Subsequent tool loads are performed by the **Load Tool** command.

Load Tool

The **Load Tool** command is used when the machine needs to change the tool it is currently working with. The command is similar to the **Load Tool First** command, but it normally also includes instructions to retract the tool to a safe position and cancel the tool length compensation before changing the tool. The command must be defined in each option file.



In this example, blocks 1 - 4 retract the tool, cancel the tool length compensation and then move the tool in preparation for the tool change. Because blocks 5 -9 are the same as those used in the **Load Tool First** command in the Fauc_Generic option file, you can replace blocks 5 - 9 by making **Load Tool First** a sub-command of **Load Tool**. Alternatively, you could create a user command that includes blocks 5 - 9 and then embed the command in both load tool commands.

Typical parameters used in this command include:

- **Tool Type**
- **Tool Name**
- **Tool Diameter**
- **Tool Tip Radius**
- **Tool Length**
- **Tool Number**

In most cases, the parameter value is set from the CLDATA file. However, you must check the machine prefix specified for each parameter is appropriate for the target controller. (To check a parameter, right-click the block item and select **Edit** from the context menu. The parameter is displayed in the **Add/Edit Parameter** dialog.)

Drilling

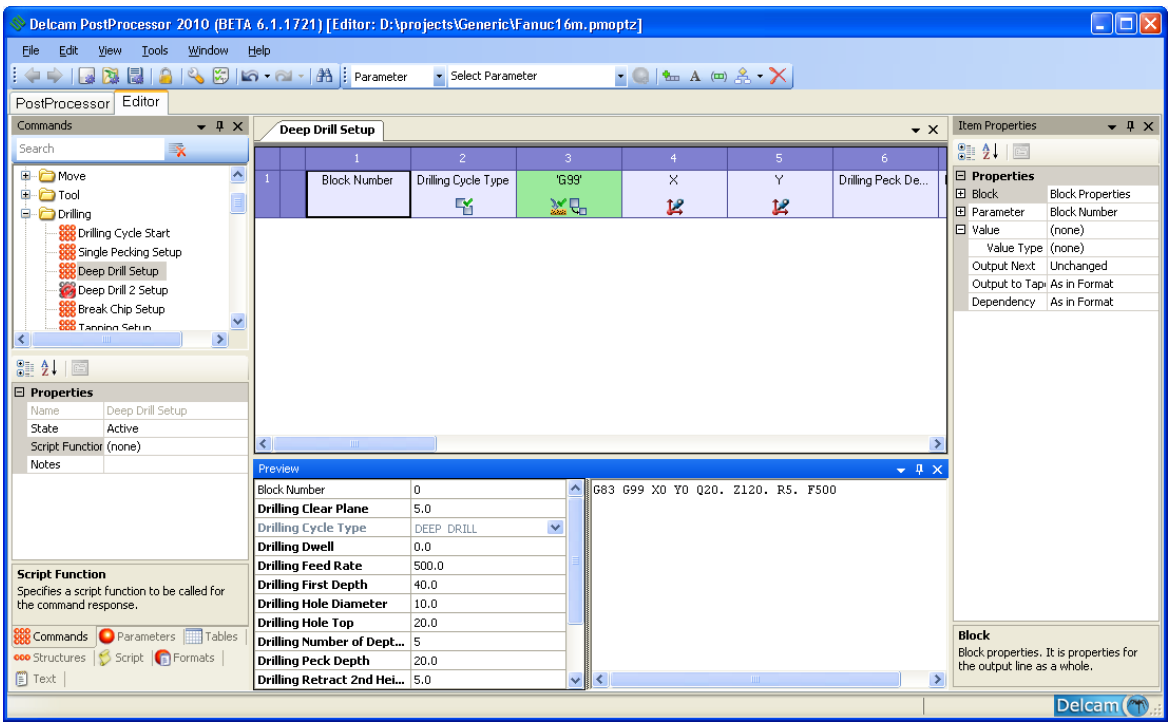
Delcam PostProcessor contains all the parameters and setup commands required to configure drilling cycles and thread milling. To ensure full compliance between CAM systems and the controller, you must define and activate all drilling cycle setup commands that may be used by the controller in the option file.

- Some parameters must be recalculated for specific controllers.
- In a drilling cycle, **Delcam PostProcessor** outputs commands in the following sequence:
- **Drilling Cycle Start** displays the specified parameter values at the start of every drilling cycle. This command enables you to display information that applies throughout the cycle, such as **Drilling Peck Depth**, **Drilling Number of Depths**, **Drilling Thread Pitch**, **Drilling Feed Rate**, **Drilling Clear Plane**, **Drilling Total Depth**, **Drilling First Depth**, and **Drilling Retract 2nd Height**.
- **Move in Cycle** specifies the hole co-ordinates and moves the tool between holes. The drilling cycle is then performed.
- **Drilling Cycle End** closes the sequence and should contain a **Drilling Cycle Type** of **Canned_Cycle_Off** and any other exit codes required by the controller.
- If your CAM system uses a drilling cycle not supported by Delcam PostProcessor, the NC program output may be incorrect.
- If a drilling cycle command specified in your CAM system is disabled or empty in the option file, Delcam PostProcessor does not display an error message when it creates the NC program. However, if a required command is disallowed, a warning message is displayed when the file is postprocessed.

Drilling cycle commands

You must define and activate all drilling cycle setup commands that may be used by the controller in the option file. These are:

- **Single Pecking** - drilling is performed in one operation.
- **Deep Drill** - drilling is performed in several stages (multiple peck).



- In this example, the **Drilling Cycle Type** parameter is used to set the machine code for the command from the CLDATA file. Fanuc code **G99** causes the tool to return to the retract plane, is output as a text string; and the **X, Y** coordinates give the position exactly above the first cycle's hole. The **Drilling Peck Depth**, **Drilling Total Depth**, **Drilling Clear Plane** and **Drilling Feed Rate** parameters are also output.

- **Deep Drill 2** - this command is used in the same way as the Deep Drill command. It is implemented only when the NC program needs to support two similar multiple-peck drilling cycles.
- **Break Chip** - drilling is performed in several stages (multiple peck) with a dwell at each peck.
- **Tapping** - machining is performed by a tapping attachment using the specified **Thread Pitch**.
- **Rigid Tapping** - tapping is performed by the machine. The command can specify a **Cycle Peck Depth** in addition to the **Thread Pitch**.
- **Ream** - the first of the boring cycles.
- **Counter Bore** - the second of the boring cycles.
- **Bore 3**, **Bore 4** and **Bore 5** - these correspond to the third, fourth and fifth type of boring cycles normally found on a machine tool. These operations vary across different machine tools.
- **Helical** - this enables the machine to bore out a large hole with a small tool. It is similar to trochoidal milling, except that trochoidal milling machines a slot (with no variation in Z), while helical milling drills a hole by machining down the Z axis.
- **Reverse Helical** - as for **Helical**, except that the hole is bored from bottom to top.
- **Helical Clockwise** - as for **Helical**, except that circular clockwise cuts are used, each one deeper than the last, until the material is fully penetrated.
- **Reverse Helical CW** - as for **Helical**, except that the hole is bored from bottom to top using circular counter-clockwise cuts.

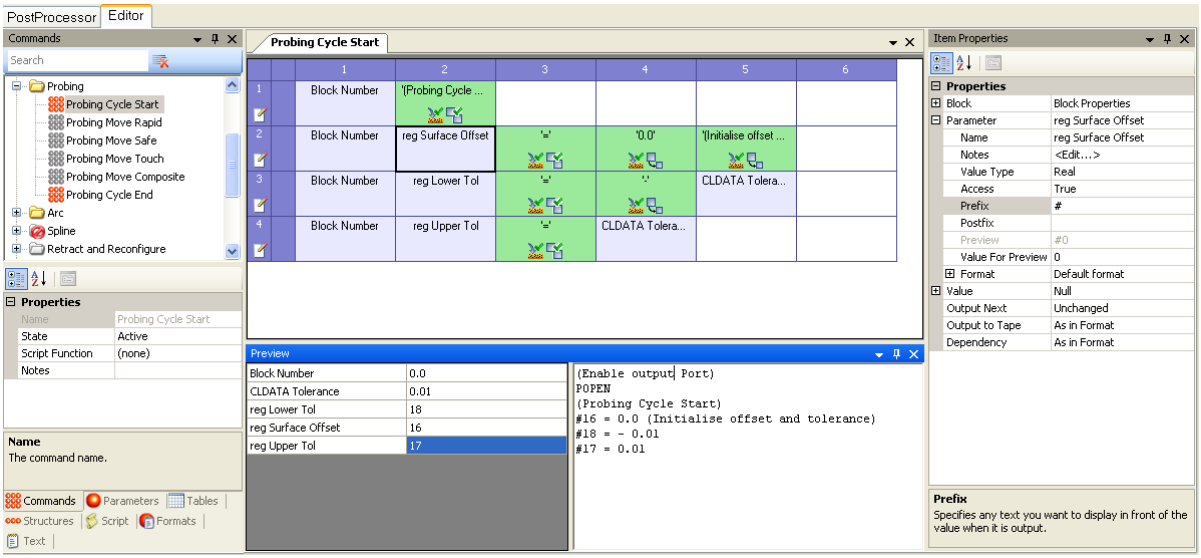
Probing

The **Probing** commands control the complex probing required in **On Machine Verification (OMV)** of parts.

> The probing commands are optimized for PowerINSPECT and this requires the machine controller to output measurements in a specific format. The probing option files use complex logic to support this, including extensive use of controller-specific macros, and you should not change the commands without expert knowledge of the target controller. Contact Delcam for support.

Probing cycle start

This **Probing Cycle Start** command starts the probing sequence. It is typically used to check that the tool is a probe and that is turned on. The command is also used to specify the parameters that are set only once, such as offsets and tolerances.



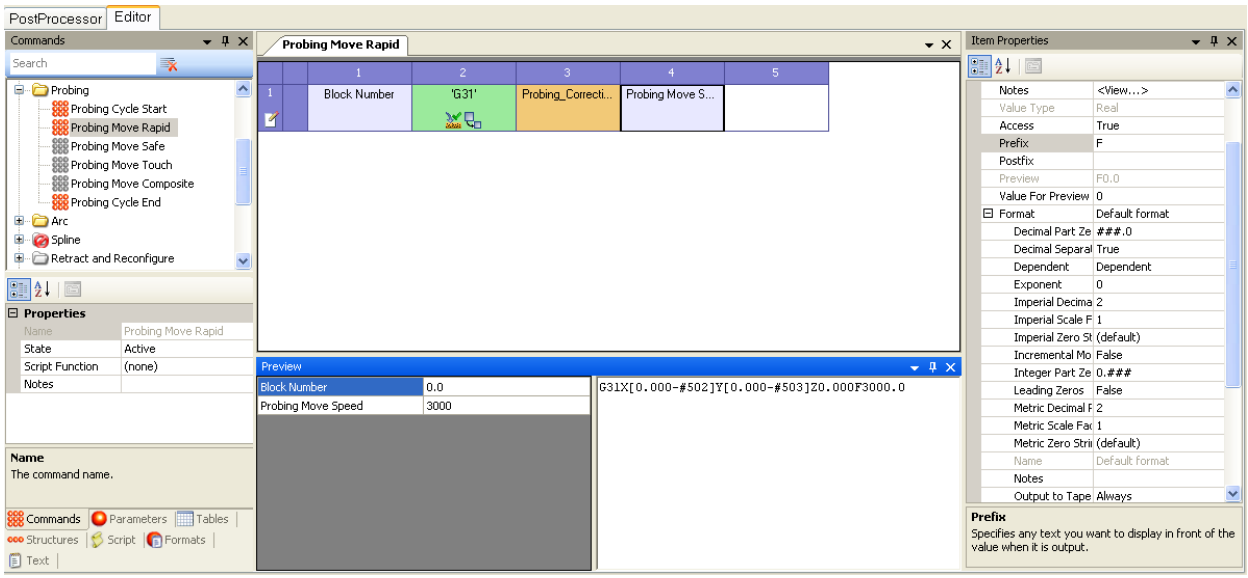
> You must set the appropriate machine prefix (**#** in the above example) for each setting in the **Parameter Properties**Editing parameters.

Probing move rapid

The **Probing Move Rapid** command controls moves when the probe is clear of the part so no collision checking is required.

In **Probing Move Rapid**, you must configure the **Motion Mode** parameter manually (by, for example, assigning, **RAP** to it when the **Probing Move Rapid** command starts, and then **LIN** to it when a different type of probing move starts.

The following example is for a machine using a Fanuc controller, where a specific speed setting is used:



A Fanuc controller uses a **G31** command to move the probe and touch a surface. For example, if you program a move of one centimeter in **G31**, the machine moves one centimeter as in **G01** (a straight-line feed move). However, in **G31**, the machine stops if the probe touches something.

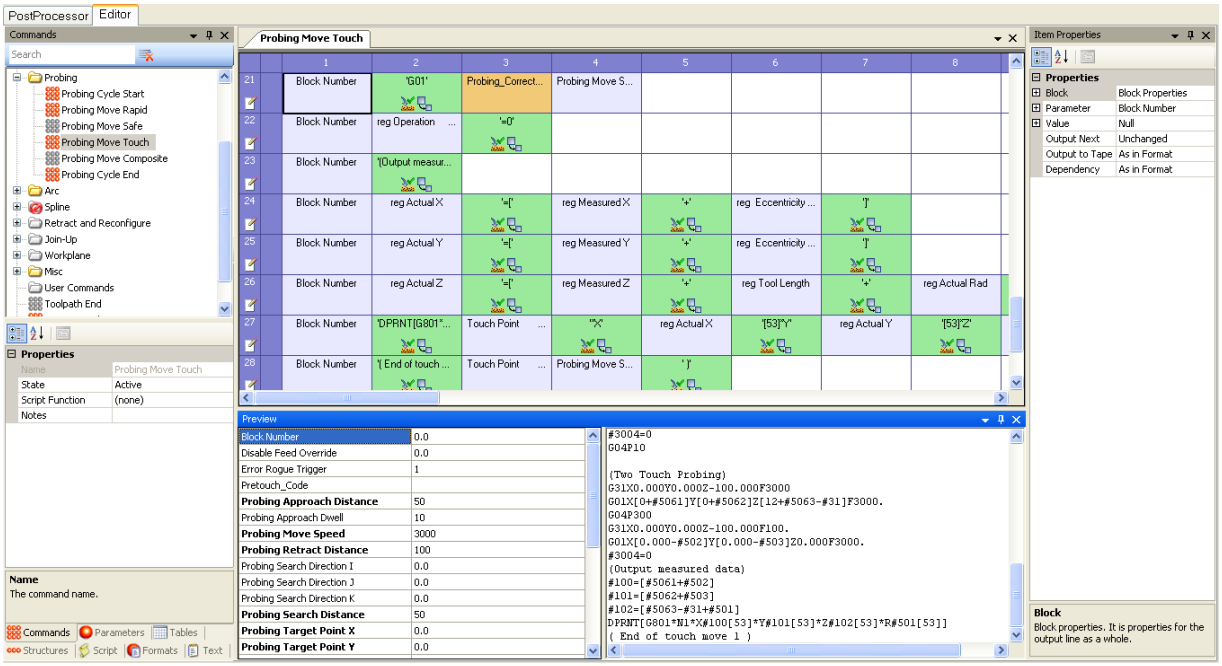
Probing move safe

The **Probing Move Safe** command is used to move the probe when it is close to the part immediately prior to making a measurement. It is similar to the **Probing Move Rapid** command except that it is normally configured to check for accidental collisions with the part.

Probing Move Safe is typically followed by a **Probing Move Touch** command.

Probing move touch

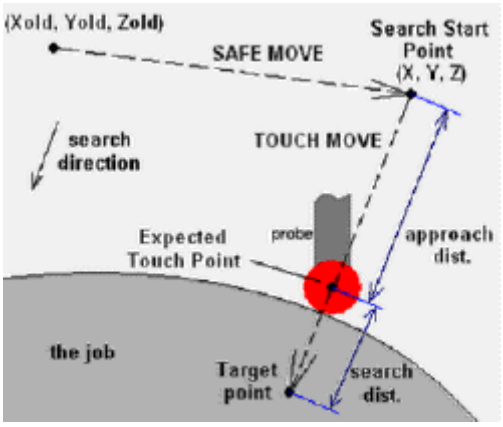
The **Probing Move Touch** command specifies how the probe moves to the surface of the part and the recording of measurements. It must also specify how to deal with the failure to make an expected touch.



The block items in yellow (**Probing_Target_Point** and **Probing_Corrected_Point**) are **Structures**. To modify a structure, right-click the block item and select **Go to Structure** from the context menu.

Probing move composite

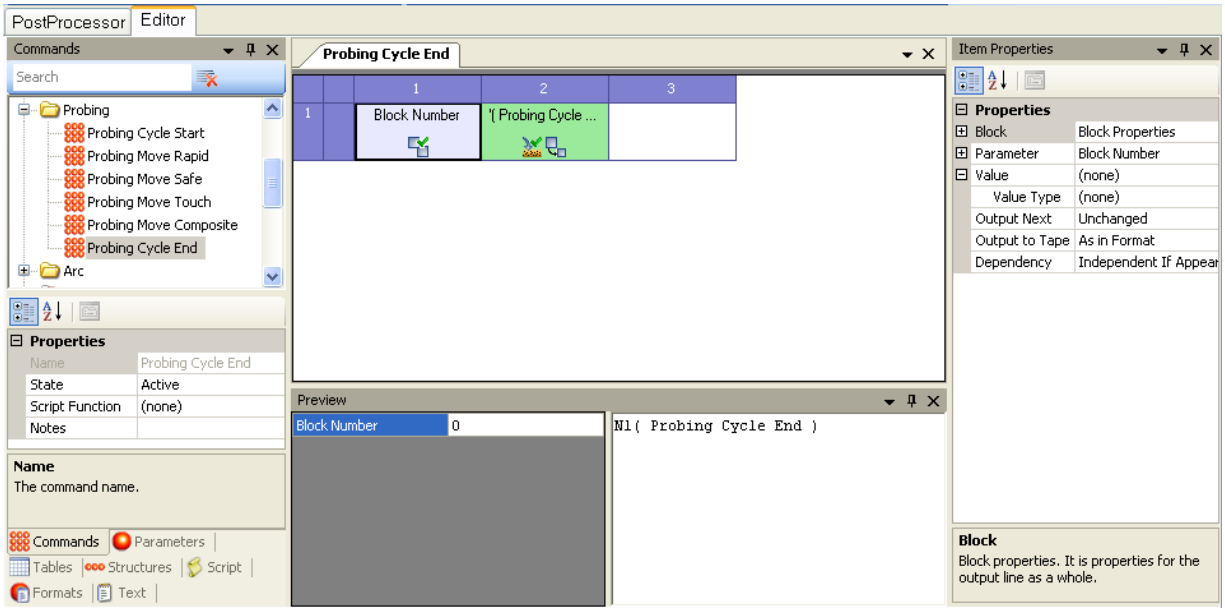
This **Probing Move Composite** command enables you to combine adjacent **Probing Move Safe** and **Probing Move Touch** commands into a single probing step. To create **Probing Move Composite** commands, the **Probing - Is Composite Move Used?** parameter must be set to **On** in the Initialisation table and the **Probing Move Composite** command must be active.



By default, Delcam PostProcessor generates an explicit sequence of **Probing Safe Move** and **Probing Touch Move** commands, so this command is not required.

Probing cycle end

The **Probing Cycle End** command specifies the end of the probing cycle. It can be used to turn off the probe is this is not done automatically. The block items used in the command are controller dependant. For example, Fanuc controllers use a text string.



Arc

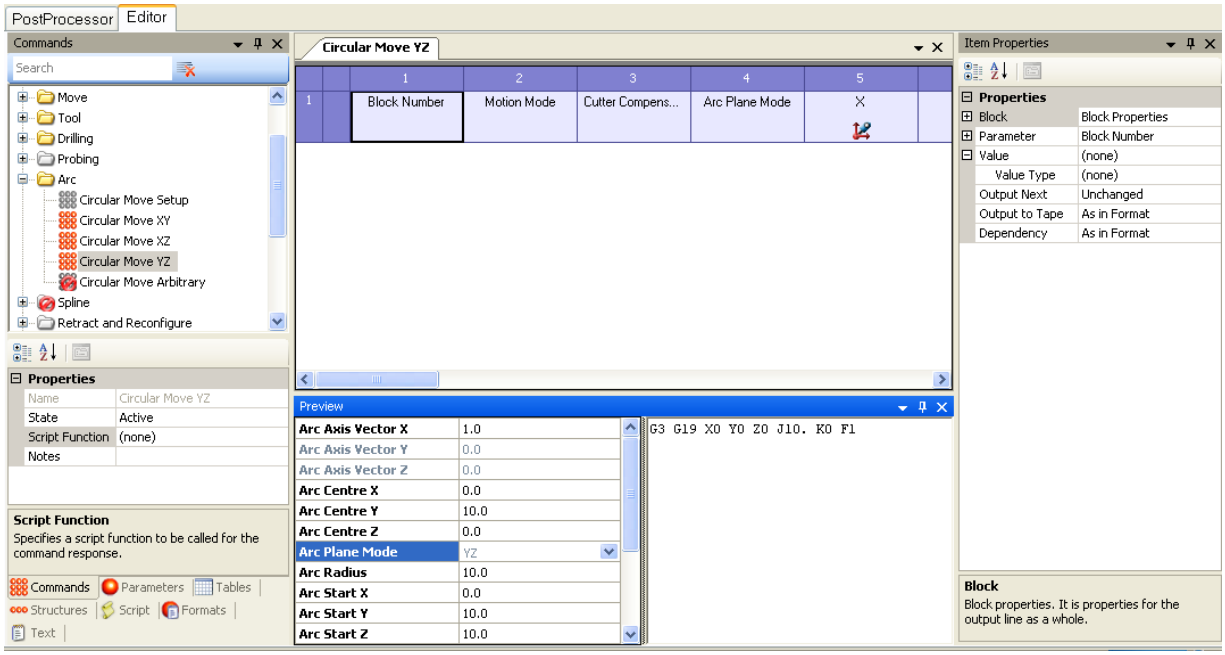
Arc commands control circular moves. The appropriate circular move is called when it is encountered in the CLDATA file:

- **Circular Move YZ**
- **Circular Move XY**
- **Circular Move XZ**

In addition, the **Circular Move Setup** command can, optionally, be used to set up any preparatory text and parameters for arc moves.

- For the settings and parameters relating to **Arc** commands, refer to Arc Interpolation is Supported and Arc Settings.

All circular moves have a similar command structure (using the parameters specified in Arc Settings). The following example demonstrates the **Circular Move YZ** command:

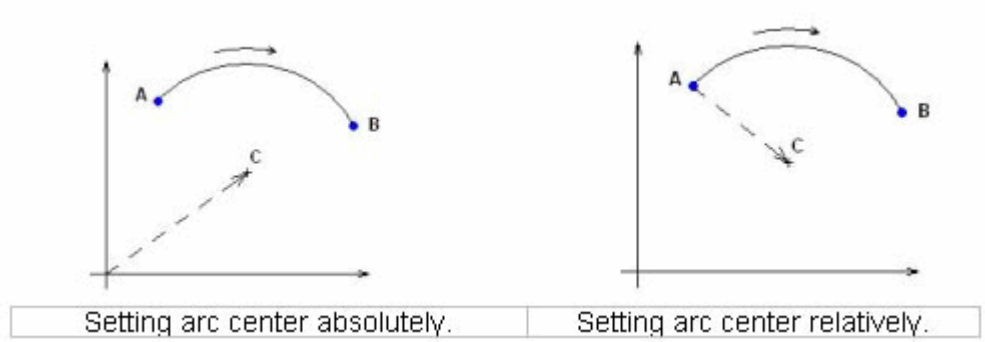


Arc settings

All circular move commands use the **Arc** parameters. However, because the layout method varies between controllers, the method by which an arc is represented must be defined when Arc Interpolation is Supported.

The arc centre coordinates, **Arc Centre X**, **Arc Centre Y** and **Arc Centre Z**, can be defined:

- In absolute coordinates
- Relative/incremental to the arc start point



Setting arc center absolutely.	Setting arc center relatively.

Arc Start X, **Arc Start Y**, **Arc Start Z** define the arc start point. This point is the same as the end point of a previous move, so it is not normally used in NC output.

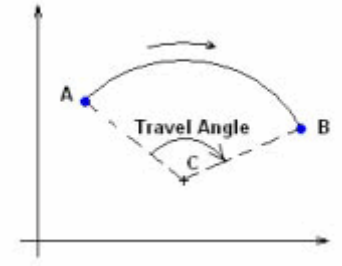
X, **Y**, **Z** define the arc end point when they are used in arc interpolation.

Arc Plane Mode sets the plane identification code for a controller. **Delcam PostProcessor** calculates the correct value of the parameter (the actual arc plane). It is only necessary to add this parameter to arc interpolation blocks to have it written out.

Arc Axis Vector X, **Arc Axis Vector Y**, **Arc Axis Vector Z** contain normalised vectors of a normal to an arc plane. They are used with machine controllers that support axes in arbitrary planes. They are not usually needed for arc interpolation output.

Arc Radius contains the arc radius value.

Arc Travel Angle contains the arc angle. The arc angle is the angle between vectors directing from the arc centre to the arc start and end points. So, the arc angle represents a path of a tool as it moves along the arc. The value is positive when the direction is counterclockwise and negative when the direction is clockwise.



The **Arc Travel Angle** value can be output to an NC program.

Motion Mode is used for both arcs and **Move Rapid** and **Move Linear** commands. The **Mode** is set from the corresponding tag in CLDATA. You must set the appropriate machine value for each **Mode** in the **Parameter Properties** (to display the properties, select the block **Motion Mode** in a command. The parameter properties will be displayed in Item Properties).

Cutter Compensation Mode add this parameter before beginning a work movement. The **Mode** is set from the corresponding tag in CLDATA. You must set the appropriate machine value for each **Mode** in the **Parameter Properties** (to

display the properties, select the block **Cutter Compensation Mode** in a command. The parameter properties will be displayed in Item Properties).

Feed Rate sets the feed rate from the corresponding tag in CLDATA. You must specify the appropriate prefix in the **Parameter Properties** (to display the properties, select the block **Feed Rate** in a command. The parameter properties will be displayed in Item Properties).

Spline start

This command initializes any required parameters before the start of any spline-points sequence.

Spline move

Refer to Spline Configuration.

Misc

Feed rate set

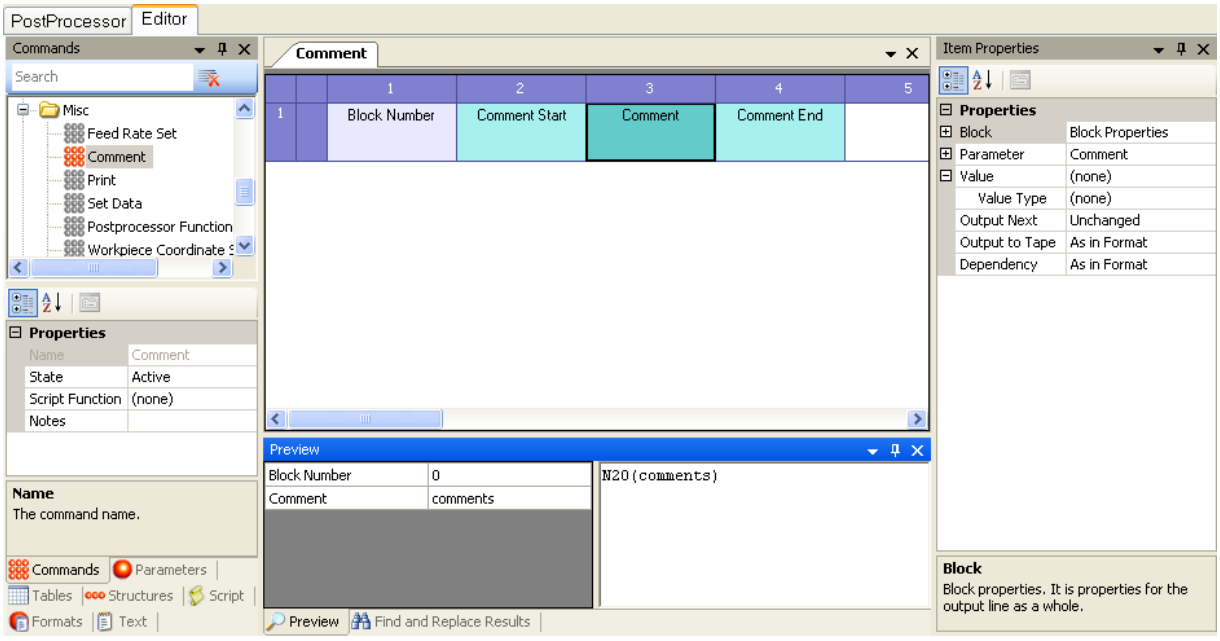
The **Feed Rate Set** command can be used to output a feed rate change when it is specified in the CLDATA file.

> Feed rates are usually specified using the **Feed Rate** parameter, which can be set automatically by CLDATA file. Because the **Feed Rate** parameter can be embedded within other commands, this command is not normally used.

Comment

Use the **Comment** command to pass comments from the CLDATA file through to the NC program. The comment is displayed by the machine controller, but it has no effect on the program.

> The **Comment** command outputs all comments in the CLDATA file including those generated automatically by the CAM system.



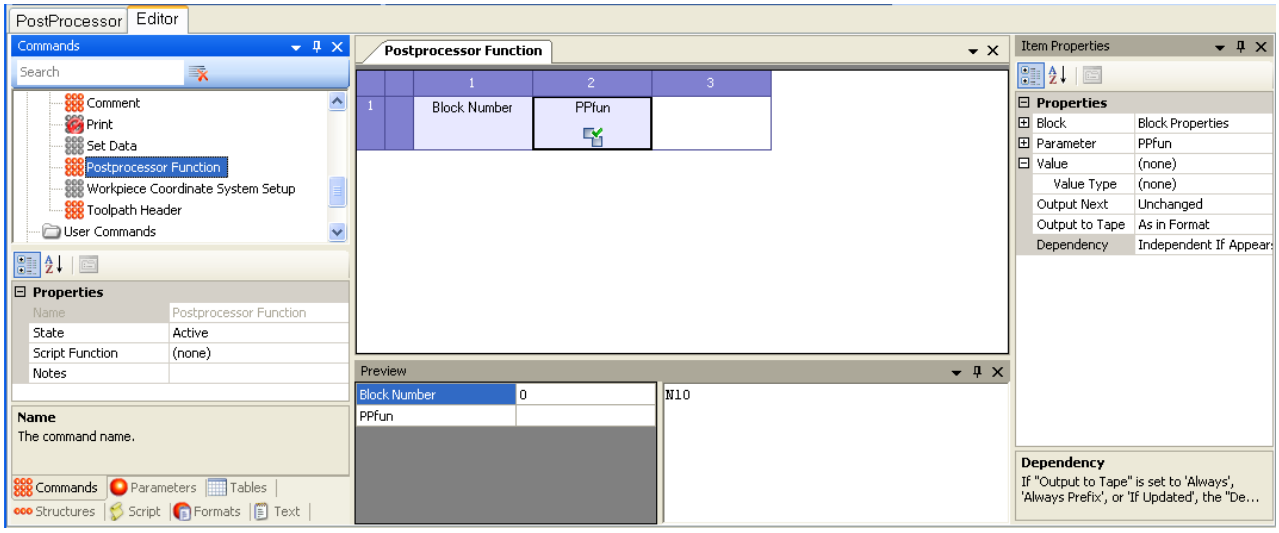
Comments must be identified by the comment commands that are specific to the target controller. For example, on Fanuc controllers, comments must be enclosed by bracket characters.

To add comment characters to items in the Comment command:

1. Select the block items.
2. Click the **Insert Comment** button in the toolbar, or right-click the selected items, and select the **Comment Selection**Block Item context menu menu option.

Postprocessor function

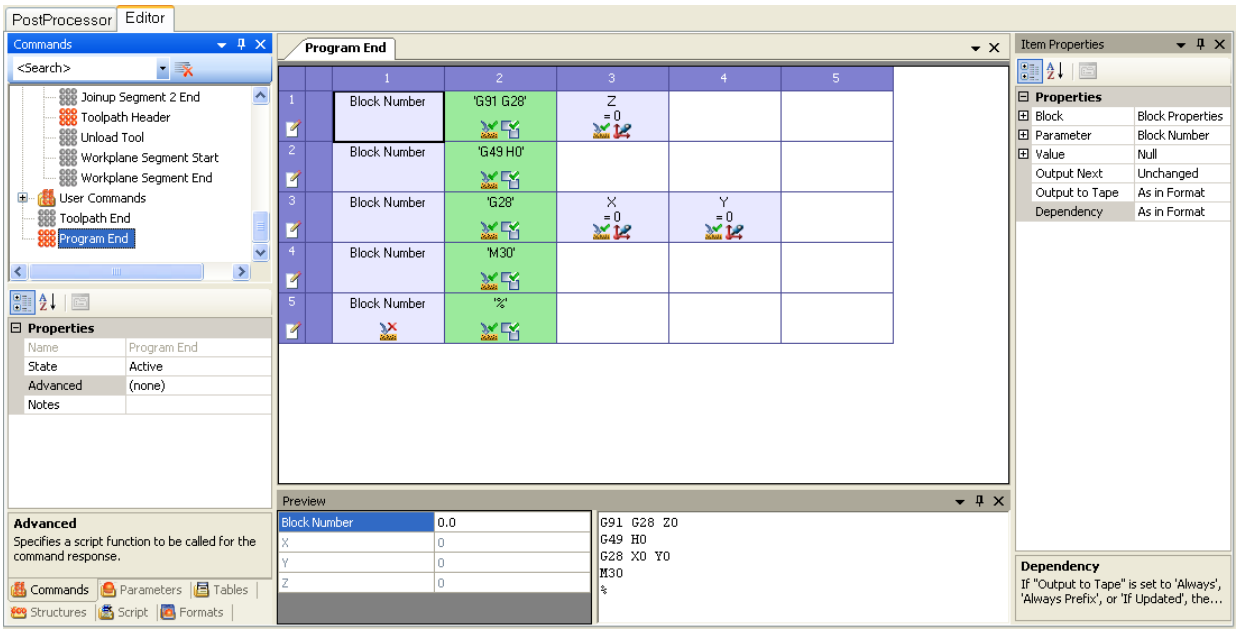
Use the **Postprocessor function** command to output commands that are not supported by Delcam PostProcessor, such as Print, Set Data and WPCS setup. It can be used to add machine-specific code to the NC program file by setting the **Value** of the function. For example, it can be used to add command line information that has been entered directly from the CAM system. However, this will make the option file machine specific. You can create a non-specific option file using script functions.



Program End

The main function of the **Program End** command is to mark the end of the postprocessing.

It is the final command in every program and must be set in each option file.



The following **Program End** settings must be set:

- Termination options, such as the positioning of the tool at the safe Z, and switching off the coolant and cutter compensation.
- The special identifying characters for the machine (for example, % for a machine using a Fanuc controller, as in the above screenshot).

Built-in parameters

This list contains the parameters supplied with Delcam PostProcessor.

Legend

Numeric Parameter

String Parameter

Group Parameter

Duration Parameter

Time Parameter

Date Parameter

- You must specify a machine value for each option in a group parameter in the **Parameter Properties** or **Item Properties** view before using the parameter; nothing is written to the output file when a parameter is set to an undefined **State**.

Parameter	Folder	Incremental	Description
Arc Axis Vector X	Arc	false	The X coordinate of the vector normal to the plane of an assigned arc.
Arc Axis Vector Y	Arc	false	The Y coordinate of the vector normal to the plane of an assigned arc.
Arc Axis Vector Z	Arc	false	The Z coordinate of the vector normal to the plane of an assigned arc.
Arc Centre X	Arc	true	The X coordinate of the centre of an arc move. This often requires the associated format to be set to Incremental Mode.
Arc Centre Y	Arc	true	The Y coordinate of the centre of an arc move. This often requires the associated format to be set to Incremental Mode.
Arc Centre Z	Arc	true	The Z coordinate of the centre of an arc move. This often requires the associated format to be set to Incremental Mode.
Arc End Angle	Arc	true	The end arc angle.
Arc Is Full Circle	Arc	N/A	
Arc Linearisation X	Special	false	
Arc Linearisation Y	Special	false	
Arc Linearisation Z	Special	false	
Arc Middle Point X	Arc	true	The X coordinate of an arc middle point.
Arc Middle Point Y	Arc	true	The Y coordinate of an arc middle point.

Parameter	Folder	Incremental	Description
Arc Middle Point Z	Arc	true	The Z coordinate of an arc middle point.
Arc Radius	Arc	false	The radius of an assigned arc.
Arc Start Angle	Arc	false	The start arc angle.
Arc Start X	Arc	true	The X coordinate of an initial arc point.
Arc Start Y	Arc	true	The Y coordinate of an initial arc point.
Arc Start Z	Arc	true	The Z coordinate of an initial arc point.
Arc Travel Angle	Arc	false	The angle between the vector from the centre of an arc to its start point and from the centre of the arc to the arc's finish point.
Axis Mode	Move	N/A	Current axis mode depending on Multiaxis ON/OFF state and multiaxis existance in the program.
Block Max X	Workpiece	false	The upper block limit along the X axis. Together with Block Min X, this defines the extent of the PowerMILL material block along the X axis.
Block Max Y	Workpiece	false	The upper block limit along the Y axis. Together with Block Min Y, this defines the extent of the PowerMILL material block along the Y axis.
Block Max Z	Workpiece	false	The upper block limit along the Z axis. Together with Block Min Z, this defines the extent of the PowerMILL material block along the Z axis.
Block Min X	Workpiece	false	The lower block limit along the X axis. Together with Block Max X, this defines the extent of the PowerMILL material block along the X axis.
Block Min Y	Workpiece	false	The lower block limit along the Y axis. Together with Block Max Y,

Parameter	Folder	Incremental	Description
			this defines the extent of the PowerMILL material block along the Y axis.
Block Min Z	Workpiece	false	The lower block limit along the Z axis. Together with Block Max Z, this defines the extent of the PowerMILL material block along the Z axis.
Block Number	Traceability	false	Number of current frame. The variable uses for itself calculation two global variables Number of start block and Block increment.
CAD WP Matrix 00	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAD WP Matrix 01	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAD WP Matrix 02	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAD WP Matrix 10	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAD WP Matrix 11	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAD WP Matrix 12	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAD WP Matrix 20	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAD WP Matrix 21	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAD WP Matrix 22	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates

Parameter	Folder	Incremental	Description
			relative to the world coordinates.
CAD WP Origin X	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAD WP Origin Y	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAD WP Origin Z	Workplanes in World CS	false	The elements of a rotation matrix defining the global coordinates relative to the world coordinates.
CAM System Version Number	Program	false	The version of the CAM System used to create the cutfile.
CLDATA Tolerance	Toolpath	false	The tolerance for an input toolpath in CLDATA. The toolpath cannot be moved away from the model by more than this value. For PowerMILL, it is set to the machining tolerance, so Tolerance = (INTOL + OUTTOL)*0.5.
Comment	Traceability	N/A	A comment string and one which does not therefore form part of the commands in the tapefile.
Contact Normal X	Move	false	The X coordinate of the contact normal.
Contact Normal Y	Move	false	The Y coordinate of the contact normal.
Contact Normal Z	Move	false	The Z coordinate of the contact normal.
Contact Point X	Move	false	The X coordinate of the tool contact point.
Contact Point Y	Move	false	The Y coordinate of the tool contact point.
Contact Point Z	Move	false	The Z coordinate of the tool contact point.

Parameter	Folder	Incremental	Description
Controller Command String	Traceability	N/A	A string containing the command for a particular machine tool controller.
Coolant Mode	Controller Switches	N/A	The Coolant Mode (ON / OFF / FLOOD / MIST / TAPPING / VACUUM / THROUGH / AIR / DOUBLE).
Current User	Traceability	N/A	The Windows login name of the current user.
Cutter Compensation Mode	Controller Switches	N/A	The Cutter Compensation Mode (ON / OFF / LEFT / RIGHT).
Cutting Direction	Toolpath	N/A	The cutting direction of the toolpath.
Cutting Rate	Toolpath	false	The rate (or speed) that the material moves past the cutting edge of the tool. See also Feed Rate.
Cycle Hole Top PMill	Special	false	The height of a cycle hole top measured from the last Z position.
Date	Traceability	N/A	The current date in the default format mm/dd/yy.
Diameter Mode	Controller Switches	N/A	
Drilling Chamfer Diameter	Canned Cycles	false	Chamfer diameter for drilling.
Drilling Clear Plane	Canned Cycles	false	The height of a cycle lead move's plane.
Drilling Cycle Type	Canned Cycles	N/A	The current type of drilling cycle.
Drilling Draft Angle	Canned Cycles	false	Output the defined draft angle on drilling cycle form.
Drilling Dwell	Canned Cycles	false	The length of a cycle dwell.

Parameter	Folder	Incremental	Description
Drilling Expanded Cycle Mode	Canned Cycles	N/A	Drilling : Expanded Cycle Mode.
Drilling Feed Rate	Canned Cycles	false	The feed rate for a cycle. See also Feed Rate.
Drilling Feed Reduction Distance End	Canned Cycles	false	Distance for which feed is to be reduced at the bottom of the drill.
Drilling Feed Reduction Distance Start	Canned Cycles	false	Distance for which feed is to be reduced at the bottom of the drill.
Drilling Feed Reduction End	Canned Cycles	false	Proportion by which feed is to be reduced at the bottom of the drill.
Drilling Feed Reduction Start	Canned Cycles	false	Proportion by which feed is to be reduced at the bottom of the drill.
Drilling Final Pass	Canned Cycles	false	If there is a final pass
Drilling Final Pass Location	Canned Cycles	false	If there is a final pass
Drilling First Depth	Canned Cycles	false	The depth of the first peck of a drilling cycle.
Drilling Hole Depth	Canned Cycles	false	The depth of the actual hole.
Drilling Hole Diameter	Canned Cycles	false	The diameter of the hole associated with a cycle.
Drilling Hole Top	Canned Cycles	false	Height of hole top measured from last Z position.
Drilling Lead Distance	Canned Cycles	false	Distance from start of lead move to it's end
Drilling Minimum Peck	Canned Cycles	false	Smallest peck if each peck is to be smaller than the last.
Drilling	Canned	N/A	The number of pecks in a cycle.

Parameter	Folder	Incremental	Description
Number of Depths	Cycles		
Drilling Overlap Angle	Canned Cycles	false	The factor by which a profile overlaps in profile drilling.
Drilling Peck Decrement	Canned Cycles	false	Peck decrement if each peck is to be smaller than the last.
Drilling Peck Depth	Canned Cycles	false	The depth of a cycle peck.
Drilling Rapid Retract	Canned Cycles	N/A	If ON then drilling cycle retracts at rapid.
Drilling Retract 2nd Height	Canned Cycles	false	The retract distance after the drill.
Drilling Retract Factor	Canned Cycles	false	The factor of the peck used to retract for break chip drilling.
Drilling Retract Feed Factor	Canned Cycles	false	The retract may be done at a different feedrate from cutting.
Drilling Retract Mode	Canned Cycles	N/A	Drilling : if 1 retract to safe Z (G98) otherwise to the clear plane (G99).
Drilling Secondary Multiaxis Clear Plane	Special	false	Secondary clear plane value for multiaxis toolpaths.
Drilling Start Offset	Canned Cycles	false	Incremental offset from the hole top.
Drilling Step Over	Canned Cycles	false	Step over for spiral
Drilling Sub Peck	Canned Cycles	false	Subsidiary peck (for each peck several sub pecks are performed).
Drilling Thread Pitch	Canned Cycles	false	This is calculated inside the Postprocessor from values in the output (otherwise, it needs to be set).

Parameter	Folder	Incremental	Description
Drilling Total Depth	Canned Cycles	false	The total depth of the cycle hole.
Drilling User Parameter	Canned Cycles	false	Allows you to set a user parameter for a drilling cycle.
Dumb Tool Length	Traceability	false	Allows you to specify the length of the tool.
Dwell	Traceability	false	The length of the dwell.
Euler A	Move	false	Euler Angle to represented multiaxis moves.
Euler B	Move	false	Euler Angle to represented multiaxis moves.
Euler C	Move	false	Euler Angle to represented multiaxis moves.
Feed Mode	Controller Switches	N/A	Sets Feed Rate representation mode: mm(inch) per minute, mm(inch) per one tool rotation or inverse time.
Feed Rate	Move	false	The feed rate is the distance (in millimetres, for metric units) that a cutting tool advances per revolution. The Plunge Rate value is used instead if the cutfile is of type Plunging.
Feed Rate Type	Move	N/A	The Feed Rate Type (UNDEFINED / CUTTING / RAPID / PLUNGE / SKIM).
Fixture Offset Type	Toolpath	N/A	Fixture offset in current toolpath.
Fixture Offset X	Toolpath	false	X Value of Fixture Offset
Fixture Offset Y	Toolpath	false	Y Value of Fixture Offset
Fixture Offset Z	Toolpath	false	Z Value of Fixture Offset

Parameter	Folder	Incremental	Description
From X	Program	true	The X coordinate of the first programmed point, not appropriate for use in multi axis option files.
From Y	Program	true	The Y coordinate of the first programmed point, not appropriate for use in multi axis option files.
From Z	Program	true	The Z coordinate of the first programmed point, not appropriate for use in multi axis option files.
Input Linear Units	Program	N/A	The units used to input linear units (MM / INCH).
Machine A	Move	true	Coordinate of Machine A axis.
Machine A Roll Over	Move	N/A	Is set to ON in a move command when A axis Roll Over happens.
Machine A Rotation	Move	N/A	Is set to ON in a move command when Machine A changed.
Machine B	Move	true	Coordinate of Machine B axis.
Machine B Roll Over	Move	N/A	Is set to ON in a move command when B axis Roll Over happens.
Machine B Rotation	Move	N/A	Is set to ON in a move command when Machine B changed.
Machine C	Move	true	Coordinate of Machine C axis.
Machine C Roll Over	Move	N/A	Is set to ON in a move command when C axis Roll Over happens.
Machine C Rotation	Move	N/A	Is set to ON in a move command when Machine C changed.
Machine D	Move	true	Coordinate of Machine D rotary axis.
Machine D Roll Over	Move	N/A	Is set to ON in a move command when D axis Roll Over happens.
Machine D Rotation	Move	N/A	Is set to ON in a move command when Machine D changed.

Parameter	Folder	Incremental	Description
Machine E	Move	true	Coordinate of Machine E rotary axis.
Machine E Roll Over	Move	N/A	Is set to ON in a move command when E axis Roll Over happens.
Machine E Rotation	Move	N/A	Is set to ON in a move command when Machine E changed.
Machine F	Move	true	Coordinate of Machine F rotary axis.
Machine F Roll Over	Move	N/A	Is set to ON in a move command when F axis Roll Over happens.
Machine F Rotation	Move	N/A	Is set to ON in a move command when Machine F changed.
Machine G	Move	true	Coordinate of Machine G rotary axis.
Machine G Roll Over	Move	N/A	Is set to ON in a move command when G axis Roll Over happens.
Machine G Rotation	Move	N/A	Is set to ON in a move command when Machine G changed.
Machine H	Move	true	Coordinate of Machine H rotary axis.
Machine H Roll Over	Move	N/A	Is set to ON in a move command when H axis Roll Over happens.
Machine H Rotation	Move	N/A	Is set to ON in a move command when Machine H changed.
Machine L	Move	true	Coordinate of Machine L rotary axis.
Machine L Roll Over	Move	N/A	Is set to ON in a move command when L axis Roll Over happens.
Machine L Rotation	Move	N/A	Is set to ON in a move command when Machine L changed.
Machine M	Move	true	Coordinate of Machine M rotary axis.
Machine M Roll Over	Move	N/A	Is set to ON in a move command when M axis Roll Over happens.

Parameter	Folder	Incremental	Description
Machine M Rotation	Move	N/A	Is set to ON in a move command when Machine M changed.
Machine N	Move	true	Coordinate of Machine N rotary axis.
Machine N Roll Over	Move	N/A	Is set to ON in a move command when N axis Roll Over happens.
Machine N Rotation	Move	N/A	Is set to ON in a move command when Machine N changed.
Machine O	Move	true	Coordinate of Machine O rotary axis.
Machine O Roll Over	Move	N/A	Is set to ON in a move command when O axis Roll Over happens.
Machine O Rotation	Move	N/A	Is set to ON in a move command when Machine O changed.
Machine P	Move	true	Coordinate of Machine P rotary axis.
Machine P Roll Over	Move	N/A	Is set to ON in a move command when P axis Roll Over happens.
Machine P Rotation	Move	N/A	Is set to ON in a move command when Machine P changed.
Machine Q	Move	true	Coordinate of Machine Q rotary axis.
Machine Q Roll Over	Move	N/A	Is set to ON in a move command when Q axis Roll Over happens.
Machine Q Rotation	Move	N/A	Is set to ON in a move command when Machine Q changed.
Machine R	Move	true	Coordinate of Machine R rotary axis.
Machine R Roll Over	Move	N/A	Is set to ON in a move command when R axis Roll Over happens.
Machine R Rotation	Move	N/A	Is set to ON in a move command when Machine R changed.
Machine S	Move	true	Coordinate of Machine S rotary axis.

Parameter	Folder	Incremental	Description
Machine S Roll Over	Move	N/A	Is set to ON in a move command when S axis Roll Over happens.
Machine S Rotation	Move	N/A	Is set to ON in a move command when Machine S changed.
Machine T	Move	true	Coordinate of Machine T axis.
Machine T Roll Over	Move	N/A	Is set to ON in a move command when T axis Roll Over happens.
Machine T Rotation	Move	N/A	Is set to ON in a move command when Machine T changed.
Machine U	Move	true	Coordinate of Machine U linear axis.
Machine V	Move	true	Coordinate of Machine V linear axis.
Machine W	Move	true	Coordinate of Machine W linear axis.
Machine X	Move	true	Coordinate of Machine X axis.
Machine Y	Move	true	Coordinate of Machine Y axis.
Machine Z	Move	true	Coordinate of Machine Z axis.
Max Cutting Rate	Option File Settings	false	The maximum rate of cutting. See also Feed Rate.
Max Rate	Option File Settings	false	The maximum Feed Rate.
Motion Mode	Controller Switches	N/A	The Motion Mode (LIN / RAP / CW / CCW / SPL / CYC).
Move Type	Move	N/A	The Move Type (UNDEFINED / LINK / APPROACH / RETRACT / LEAD IN / LEAD OUT / JOINUP).
NC Program Name	Program	N/A	The name of the associated NC Program.
NC Program Path	Program	N/A	Full path to the NC Program.
NC Program	Option File Settings	false	The tolerance of the NC Program used for multi-axis and arc

Parameter	Folder	Incremental	Description
Tolerance			linearisation.
Optfile Author	Option File Info	N/A	The author of the option file.
Optfile Controller Manufacturer	Option File Info	N/A	The manufacturer of the machine controller associated with the option file.
Optfile Controller Series	Option File Info	N/A	The series and version numbers of the machine controller associated with the option file.
Optfile Created Date	Option File Info	N/A	The date on which the option file was created.
Optfile Customer	Option File Info	N/A	The customer for whom the option file is intended.
Optfile Last Modified Date	Option File Info	N/A	The date on which the option file was last modified.
Optfile Machine Tool Manufacturer	Option File Info	N/A	The manufacturer of the machine tool associated with the option file.
Optfile Machine Tool Model	Option File Info	N/A	The model of the machine tool associated with the option file.
Optfile Name	Option File Info	N/A	The name of the option file.
Optfile Special Note	Option File Info	N/A	For example ISO, Advanced Function, Vertical, etc.
Optfile Version	Option File Info	N/A	The version of **Delcam PostProcessor** that was used to create the option file.
Orientation Vector X	Move	false	X coordinate of the current orientation vector.
Orientation Vector Y	Move	false	Y coordinate of the current orientation vector.
Orientation Vector Z	Move	false	Z coordinate of the current orientation vector.

Parameter	Folder	Incremental	Description
Output Angular Units	Option File Settings	N/A	From a PowerMILL cutfile, the output format of the angular units (DEG / RAD).
Output Linear Units	Option File Settings	N/A	From a PowerMILL cutfile, the output format of the linear units (MM / INCH).
Output Point Info	Move	N/A	Contains information about the properties of the point coordinates currently held in the X, Y and Z parameters.
Output Point Mode	Move	N/A	Contains information about the mode of the point coordinates currently held in the X, Y and Z parameters.
Output Rotation Matrix 00	Workplane	false	The elements of a rotation matrix defining the local coordinates relative to the global coordinates.
Output Rotation Matrix 01	Workplane	false	The elements of a rotation matrix defining the local coordinates relative to the global coordinates.
Output Rotation Matrix 02	Workplane	false	The elements of a rotation matrix defining the local coordinates relative to the global coordinates.
Output Rotation Matrix 10	Workplane	false	The elements of a rotation matrix defining the local coordinates relative to the global coordinates.
Output Rotation Matrix 11	Workplane	false	The elements of a rotation matrix defining the local coordinates relative to the global coordinates.
Output Rotation Matrix 12	Workplane	false	The elements of a rotation matrix defining the local coordinates relative to the global coordinates.
Output Rotation Matrix 20	Workplane	false	The elements of a rotation matrix defining the local coordinates relative to the global coordinates.
Output Rotation	Workplane	false	The elements of a rotation matrix defining the local coordinates

Parameter	Folder	Incremental	Description
Matrix 21			relative to the global coordinates.
Output Rotation Matrix 22	Workplane	false	The elements of a rotation matrix defining the local coordinates relative to the global coordinates.
Part Name	Traceability	N/A	The name of the part to be machined.
Permanent Comment	Special	N/A	
Plane Mode	Controller Switches	N/A	From a PowerMILL cutfile, this is the tool axis (XY, XZ or YZ) used as the current plane for arc interpolation.
Plunge Rate	Toolpath	false	The Plunge Rate from a PowerMILL cutfile is used as the Feed Rate if the cutfile is of type Plunging.
Polar Angle	Move	false	Angle to represent polar coordinate.
Polar Centre X	Move	false	Define centre of polar representation (see Polar Angle, Polar Radius).
Polar Centre Y	Move	false	Define centre of polar representation (see Polar Angle, Polar Radius).
Polar Centre Z	Move	false	Define centre of polar representation (see Polar Angle, Polar Radius).
Polar Radius	Move	false	Radius to represent polar coordinate.
PPfun	Traceability	N/A	User function that can be set from within PowerMILL or PowerINSPECT.
Probing Approach Distance	Probing	false	The approach distance used in a PowerINSPECT probe path.
Probing Expected Touch Point X	Probing	true	The X coordinate of the point that is expected to be touched next in a PowerINSPECT probe path.

Parameter	Folder	Incremental	Description
Probing Expected Touch Point Y	Probing	true	The Y coordinate of the point that is expected to be touched next in a PowerINSPECT probe path.
Probing Expected Touch Point Z	Probing	true	The Z coordinate of the point that is expected to be touched next in a PowerINSPECT probe path.
Probing Is Composite Move Redirection	Special	N/A	
Probing Is Composite Move Used	Probing	N/A	Set to ON for a composite move in PowerINSPECT. Otherwise, set to OFF.
Probing Move Speed	Probing	false	The speed of the move in PowerINSPECT.
Probing Retract Distance	Probing	false	The distance that the probe is retracted in PowerINSPECT.
Probing Search Direction I	Probing	false	The X coordinate of the probing search direction in PowerINSPECT.
Probing Search Direction J	Probing	false	The Y coordinate of the probing search direction in PowerINSPECT.
Probing Search Direction K	Probing	false	The Z coordinate of the probing search direction in PowerINSPECT.
Probing Search Distance	Probing	false	The length of the probing search in PowerINSPECT.
Probing Stylus Direction X	Probing	false	The X coordinate of the Stylus Direction.
Probing Stylus Direction Y	Probing	false	The Y coordinate of the Stylus Direction.

Parameter	Folder	Incremental	Description
Probing Stylus Direction Z	Probing	false	The Z coordinate of the Stylus Direction.
Probing Target Point X	Probing	true	The X coordinate of the probing target point in PowerINSPECT.
Probing Target Point Y	Probing	true	The Y coordinate of the probing target point in PowerINSPECT.
Probing Target Point Z	Probing	true	The Z coordinate of the probing target point in PowerINSPECT.
Probing Touch Speed	Probing	false	The speed of the probing touches in PowerINSPECT.
Product Version	Traceability	N/A	The current version of **Delcam PostProcessor**.
Program Cutting Duration	Program	N/A	The duration of the NC Program.
Program Number	Program	N/A	The number of the NC Program.
Program X Max	Program	false	Maximal X of the program.
Program X Min	Program	false	Minimal X of the program.
Program Y Max	Program	false	Maximal Y of the program.
Program Y Min	Program	false	Minimal Y of the program.
Program Z Max	Program	false	Maximal Z of the program.
Program Z Min	Program	false	Minimal Z of the program.
Project Name	Program	N/A	The name of the project generated in CLDATA.

Parameter	Folder	Incremental	Description
Quaternion W	Move	false	Quaternion to represented multiaxis moves.
Quaternion X	Move	false	Quaternion to represented multiaxis moves.
Quaternion Y	Move	false	Quaternion to represented multiaxis moves.
Quaternion Z	Move	false	Quaternion to represented multiaxis moves.
RR Origin X	Retract and Reconfigure	true	Retract & Reconfigure origin point.
RR Origin Y	Retract and Reconfigure	true	Retract & Reconfigure origin point.
RR Origin Z	Retract and Reconfigure	true	Retract & Reconfigure origin point.
RR Plunge X	Retract and Reconfigure	true	Retract & Reconfigure plunge point.
RR Plunge Y	Retract and Reconfigure	true	Retract & Reconfigure plunge point.
RR Plunge Z	Retract and Reconfigure	true	Retract & Reconfigure plunge point.
RR Safe X	Retract and Reconfigure	true	Retract & Reconfigure safe point.
RR Safe Y	Retract and Reconfigure	true	Retract & Reconfigure safe point.
RR Safe Z	Retract and Reconfigure	true	Retract & Reconfigure safe point.
RR Withdrawal Distance	Retract and Reconfigure	false	Retract & Reconfigure withdrawal distance.
RTCP Mode	Controller Switches	N/A	The Rotation Tool Compensation Point (ON / OFF). If it is set to OFF, **Delcam PostProcessor** will calculate the Pivot Point from the tip of the tool.

Parameter	Folder	Incremental	Description
Skim Distance	Toolpath	false	The skim distance
Skim Rate	Toolpath	false	The skim rate.
Spindle Mode	Controller Switches	N/A	The Spindle Rotation Mode (CW / CCW / OFF).
Spindle Speed	Move	false	The spindle speed.
Spline K0x	Spline	false	Spline coefficient "Spline K0x".
Spline K0y	Spline	false	Spline coefficient "Spline K0y".
Spline K0z	Spline	false	Spline coefficient "Spline K0z".
Spline K1x	Spline	false	Spline coefficient "Spline K1x" (MillPlus X51).
Spline K1y	Spline	false	Spline coefficient "Spline K1y" (MillPlus Y51).
Spline K1z	Spline	false	Spline coefficient "Spline K1z" (MillPlus Z51).
Spline K2x	Spline	false	Spline coefficient "Spline K2x" (MillPlus X52).
Spline K2y	Spline	false	Spline coefficient "Spline K2y" (MillPlus Y52).
Spline K2z	Spline	false	Spline coefficient "Spline K2z" (MillPlus Z52).
Spline K3x	Spline	false	Spline coefficient "Spline K3x" (MillPlus X53).
Spline K3y	Spline	false	Spline coefficient "Spline K3y" (MillPlus Y53).
Spline K3z	Spline	false	Spline coefficient "Spline K3z" (MillPlus Z53).
Spline Knot 1	Spline	false	Spline Knot Vector 1-th coefficient.

Parameter	Folder	Incremental	Description
Spline Knot 2	Spline	false	Spline Knot Vector 2-th coefficient.
Spline Knot 3	Spline	false	Spline Knot Vector 3-th coefficient.
Spline Knot 4	Spline	false	Spline Knot Vector 4-th coefficient.
Spline Knot 5	Spline	false	Spline Knot Vector 5-th coefficient.
Spline Knot 6	Spline	false	Spline Knot Vector 6-th coefficient.
Spline Knot 7	Spline	false	Spline Knot Vector 7-th coefficient.
Spline Knot 8	Spline	false	Spline Knot Vector 8-th coefficient.
Spline Mode	Spline	N/A	For example, the mode representing Cubic Polynomial Splines.
Spline Order	Spline	N/A	The order of the spline coefficients.
Split Segment Number	Traceability	N/A	Number of split segment.
Thickness	Toolpath	false	The thickness (stock offset) left on the model.
Thread Milling Allowance	Canned Cycles	false	Thread Milling Allowance.
Thread Milling Cuts Number	Canned Cycles	false	Thread Milling Cuts Number.
Thread Milling Lead Angle	Canned Cycles	false	Thread Milling Lead Angle.
Thread Milling Turns	Canned Cycles	false	Thread Milling Turns Number.

Parameter	Folder	Incremental	Description
Number			
Time	Traceability	N/A	Contains current time.
Time - Day	Traceability	N/A	The current number days into the current month (in the range 1-31).
Time - Hour	Traceability	N/A	The current number of hours into the current day (in the range 0-23).
Time - Min	Traceability	N/A	The current number of minutes into the current hour (in the range 0-59).
Time - Month	Traceability	N/A	Contains current month as a number 1-12.
Time - Sec	Traceability	N/A	The current number of seconds into the current minute (in the range 0-59).
Time - Year	Traceability	N/A	The last two digits of the current year.
Tool Compensation Length	Tool	false	The compensation length is only useful for 3-axis and is used to transform the whole toolpath in Z by that length.
Tool Compensation Radius	Tool	false	The tool radius compensation value.
Tool Cutting Duration	Tool	N/A	The length of cutting time for the tool.
Tool Cutting Length	Tool	false	The cutting length of the tool.
Tool Diameter	Tool	false	The diameter of the tool.
Tool Flutes Number	Tool	false	The number of flutes.
Tool Gauge Length	Tool	false	The tool length inclusive of holder length.

Parameter	Folder	Incremental	Description
Tool Index	Tool	N/A	Index, sequence number of the Tool.
Tool Length	Tool	false	The length is the total length of the cutter and holder assembly when it is mounted in the machine. The length is measured from the tip of the tool to the Gauge Face, which is the ground face of the spindle.
Tool Length Compensation Mode	Controller Switches	N/A	The tool length compensation value.
Tool Length Offset Number	Tool	false	The tool length offset number in the machine table.
Tool Life	Traceability	false	Seconds of tool working.
Tool Name	Tool	N/A	The name of the tool.
Tool Name Next	Tool	N/A	The name of the next tool.
Tool Number	Tool	false	The number of the tool.
Tool Number Next	Tool	false	The number of the next tool.
Tool Output Point	Tool	N/A	The tool output point (Tip / Centre).
Tool Overhang	Tool	false	The value of the tool overhang required to avoid collisions.
Tool Radius Offset Number	Tool	false	The tool radius offset number in the machine table.
Tool Taper Angle	Tool	false	The tool taper angle.
Tool Tip Radius	Tool	false	The radius of the tool tip.
Tool Tip Radius X	Tool	false	The X value of the tool tip radius.

Parameter	Folder	Incremental	Description
Tool Tip Radius Y	Tool	false	The Y value of the tool tip radius.
Tool Type	Tool	N/A	The type of tool.
Tool Vector From X	Program	false	The X coordinate of the first programmed tool vector.
Tool Vector From Y	Program	false	The Y coordinate of the first programmed tool vector.
Tool Vector From Z	Program	false	The Z coordinate of the first programmed tool vector.
Tool Vector X	Move	false	The X coordinate of the current tool vector.
Tool Vector Y	Move	false	The Y coordinate of the current tool vector.
Tool Vector Z	Move	false	The Z coordinate of the current tool vector.
Toolpath Axis Mode	Move	N/A	Axis mode of a current toolpath.
Toolpath Axis Mode Commanded	Move	N/A	Axis mode of a current toolpath commanded from CLDATA.
Toolpath Cutting Duration	Toolpath	N/A	The length of time required to cut the toolpath.
Toolpath Cutting Strategy	Toolpath	N/A	The cutting strategy used within PowerMILL.
Toolpath Index	Toolpath	N/A	Index, sequence number of the Toolpath.
Toolpath Length	Toolpath	false	The length of the toolpath.
Toolpath Name	Toolpath	N/A	The name of the toolpath.

Parameter	Folder	Incremental	Description
Toolpath Strategy Subtype	Toolpath	N/A	Strategy type (e.g. single peck for drilling)
Toolpath Type	Toolpath	N/A	The type of the toolpath.
Toolpath X Max	Toolpath	false	Maximal X in current toolpath.
Toolpath X Min	Toolpath	false	Minimal X in current toolpath.
Toolpath Y Max	Toolpath	false	Maximal Y in current toolpath.
Toolpath Y Min	Toolpath	false	Minimal Y in current toolpath.
Toolpath Z Max	Toolpath	false	Maximal Z in current toolpath.
Toolpath Z Min	Toolpath	false	Minimal Z in current toolpath.
User Defined Parameter Name	Program	N/A	User Defined Parameter Name
User Defined Parameter Value	Program	N/A	User Defined Parameter Value
Variable Feed Rate	Toolpath	N/A	The toolpath following this record contains some feedrates which are not plunge,cutting or rapid
Workpiece Coordinate System Number	Workpiece	N/A	The number of the Workpiece Coordinate System. In other words, the number of the datum shift (usually, it is set using G54-59 for Fanuc).
Workplane Definition Mode	Option File Settings	N/A	Euler or Machine angles.

Parameter	Folder	Incremental	Description
Workplane Euler A	Workplane	false	The Euler angle A taken from the rotation matrix of the local coordinates. The angular position can be described by three Euler angles or two Machine angles, depending on how the controller is set up.
Workplane Euler B	Workplane	false	The Euler angle B taken from the rotation matrix of the local coordinates.
Workplane Euler C	Workplane	false	The Euler angle C taken from the rotation matrix of the local coordinates.
Workplane Origin X	Workplane	true	The displacement of the set local origin relative to the global origin for the X coordinate. Used in Workplane Transformation for 3+2 enabled controllers.
Workplane Origin Y	Workplane	true	The displacement of the set local origin relative to the global origin for the Y coordinate.
Workplane Origin Z	Workplane	true	The displacement of the set local origin relative to the global origin for the Z coordinate.
Workplane Output Name	Workplane	N/A	The name of the workplane used to output the local coordinates.
Workplane Toolpath Name	Workplane	N/A	The name of the toolpath output in local coordinates.
Workplane Transformation Mode	Controller Switches	N/A	This can be controlled manually using Set Workplane ON and OFF.
WP Machine A	Move	true	Coordinate of WP Machine A axis.
WP Machine B	Move	true	Coordinate of WP Machine B axis.
WP Machine C	Move	true	Coordinate of WP Machine C axis.

Parameter	Folder	Incremental	Description
WP Machine D	Move	true	Coordinate of WP Machine D axis.
WP Machine E	Move	true	Coordinate of WP Machine E axis.
WP Machine F	Move	true	Coordinate of WP Machine F axis.
WP Machine G	Move	true	Coordinate of WP Machine G axis.
WP Machine H	Move	true	Coordinate of WP Machine H axis.
WP Machine L	Move	true	Coordinate of WP Machine L rotary axis.
WP Machine M	Move	true	Coordinate of WP Machine M rotary axis.
WP Machine N	Move	true	Coordinate of WP Machine N rotary axis.
WP Machine O	Move	true	Coordinate of WP Machine O rotary axis.
WP Machine P	Move	true	Coordinate of WP Machine P rotary axis.
WP Machine Q	Move	true	Coordinate of WP Machine Q rotary axis.
WP Machine R	Move	true	Coordinate of WP Machine R rotary axis.
WP Machine S	Move	true	Coordinate of WP Machine S rotary axis.
WP Machine T	Move	true	Coordinate of WP Machine T axis.
WP Machine U	Move	true	Coordinate of WP Machine U linear axis.
WP Machine V	Move	true	Coordinate of WP Machine V linear axis.

Parameter	Folder	Incremental	Description
WP Machine W	Move	true	Coordinate of WP Machine W axis.
WP Machine X	Move	true	Coordinate of WP Machine X axis.
WP Machine Y	Move	true	Coordinate of WP Machine Y axis.
WP Machine Z	Move	true	Coordinate of WP Machine Z axis.
WP mxs X	Workplane	false	The workplane X coordinate for multi-axis representation.
WP mxs Y	Workplane	false	The workplane Y coordinate for multi-axis representation.
WP mxs Z	Workplane	false	The workplane Z coordinate for multi-axis representation.
WP Safe Z	Workplane	false	The workplane safe height.
WP Start Z	Workplane	false	The workplane start height.
X	Move	true	The current X coordinate of the tool.
Y	Move	true	The current Y coordinate of the tool.
Z	Move	true	The current Z coordinate of the tool.
Zero Tool Length	Traceability	N/A	On Table - Table machines it is useful to set the tool length to 0 by setting this parameter to ON.

Built-in functions

These are the script functions that come as standard with **Delcam PostProcessor**.

ActiveCommandId

ActiveCommandId()

Returns the ID of command that call a script.

Example (JScript)

```
function test()
{
  return ActiveCommandId;
}
```

ActiveCommandName

ActiveCommandName()

Returns the name of command that call a script.

Example (JScript)

```
function test()
{
  return ActiveCommandName;
}
```

AdvancedResponse

AdvancedResponse(*[command]*)

Executes the script function associated with the referenced command and returns its value. If called command has no assigned script, **AdvancedResponse** return NC output of referenced command.

> 💡 Refer to Scripting and Examples of Scripting principles for the difference between **StandardResponse()** and **AdvancedResponse()**.

Example (VBScript)

In the following example of a **Probing Move Composite**, the value for **ptm** is derived from the **AdvancedResponse()** for **Probing Move Touch** (the script function associated with the **Probing Move Touch** command):

```
Function ProbingCompositeMove
  ' A Probing Move Composite is a Probing Move Rapid to the Approach Point
  ' followed by a Probing Move Touch to the Probing Target Point
  dim prm, ptm, ret
  prm = StandardResponse("%b(Probing Move Rapid)%")
  ' Note AdvancedResponse so script is run
  ptm = AdvancedResponse("%b(Probing Move Touch)%")
  ret = StandardResponse & prm & ptm
  ProbingCompositeMove = ret
end Function
```

InitialSplittingPartNumber

InitialSplittingPartNumber()

Get/set the number of first splittig part.

Example (JScript)

```
function ProgramStart()
{
  InitialSplittingPartNumber = 100;
  return StandardResponse();
}
```

GetCoordinate

GetCoordinate(*parameter, representation_id*)

This function returns the last value of the coordinate parameter that was output to the NC Program file in the specified coordinate system. The function returns a value type of real.

> 💡 If the parameter has not been output to the NC Program file, the function returns the parameter value specified in the Initialisation settings.

Arguments

parameter - Parameter name string

representation_id - The coordinate system for which the parameter value is to be returned. Select:

0 to return the value in the NC-program output system. (This option is the equivalent of the **GetParam()** function.)

1 to return the value in the Model Coordinate System.

2 to return the value in the Machine Coordinate System with multi-axis transformation applied, but excluding tool length compensation.

3 to return the value in the Local Workplane.

4 to return the value in the CAD Workplane.

Examples (JScript)

```
// Returns current X coordinate in Machine system.
x_coord = GetCoordinate("%p(X)%", 2);

// Returns current X coordinate in NC-output system.
x_coord = GetCoordinate("%p(X)%", 0);
x_coord = GetParam("%p(X)%");
```

GetDynamicAxisInitial

GetDynamicAxisInitial*(axis_name)*

This function returns the current initial rotary angle for the specified axis.

Arguments

axis_name - The axis name string.

GetDynamicAxisMax

GetDynamicAxisMin*(axis_name)*

This function returns the current maximum rotary angle for the specified axis.

Arguments

axis_name - The axis name string.

GetDynamicAxisMin

GetDynamicAxisMin*(axis_name)*

This function returns the current minimum rotary angle for the specified axis.

Arguments

axis_name - The axis name string

GetParam

GetParam(*parameter* [,*rounding*])

This function returns the value of the specified parameter. The value type depends on the parameter type.

Arguments

parameter - Parameter name string

rounding - Boolean value

The *rounding* argument is optional:

- The **false** state returns the unrounded value as it is stored in **Delcam PostProcessor**.
- The **true** state rounds a **Real** value to the number of digits specified in the format used by the parameter.

The *rounding* argument does not affect **Integer** or **String** parameters.

Examples (JScript)

```
comment = GetParam("%p(Comment)%");
// Returns string value of "Comment" parameter.
x_coord = GetParam("%p(X)%");
// or
x_coord = GetParam("%p(X)%",true);
// Returns value of "X" parameter, but rounded using the number of digits specified in the output format used by "X".
x_coord = GetParam("%p(X)%",false);
// Returns the unrounded value of "X" parameter ("X" is of type REAL).
frame_number = GetParam("%p(Block Number)%");
// Returns the integer value for the current block number.
```

- Refer also to the example given for **StandardResponse()**.

GetParamPrevValue

GetParamPrevValue(*parameter* [,*rounding*])

The function returns the previous value of a specified parameter. The value type depends on the parameter type.

Arguments

parameter - Parameter name string

rounding - Boolean value

The rounding argument is optional. It sets the rounding property for the return of a **Real** value. If you use the **false** state, **GetParamPrevValue()** returns the unrounded value as it is stored in **Delcam PostProcessor**. If you use the **true** state, **GetParamPrevValue()** rounds a **Real** value to the number of digits specified in the format used by the parameter. The *rounding* argument does not affect **Integer** or **String** parameters.

Examples (JScript)

```
  comment = GetParamPrevValue("%p(Workplane Output Name)%");
  // Returns a string containing the name of the previous workplane used.

  x_coord = GetParamPrevValue("%p(X)%");
  // or
  x_coord = GetParamPrevValue("%p(X)%", true);
  // Returns the rounded REAL value of the X coordinate from the previous move.

  x_coord = GetParamPrevValue("%p(X)%", false);
  // Returns the unrounded REAL value of the X coordinate from the previous move.

  frame_number = GetParamPrevValue("%p(Block Number)%");
  // Returns the previous block number.

// --------------------------------
function Move()
// how to avoid repeating moves like:
// N100 X10 Y20 Z30 B50 C60
// N110 X10 Y20 Z30 B50 C60
// if the coordinates are not controlled by the UPDATE flag.
//
// Just use this function for all motion commands.
// If all the coordinates are the same as in the previous move
// then a command response will not be processed.
//
// Note that ROUNDED values are used.
//
{
  if (GetParamPrevValue("%p(X)%", true) != GetParam("%p(X)%", true) ||
      GetParamPrevValue("%p(Y)%", true) != GetParam("%p(Y)%", true) ||
      GetParamPrevValue("%p(Z)%", true) != GetParam("%p(Z)%", true) ||
      GetParamPrevValue("%p(Machine B)%", true) != GetParam("%p(Machine
B)%" ,true) ||
      GetParamPrevValue("%p(Machine C)%", true) != GetParam("%p(Machine
C)%", true))
  {
    return StandardResponse();
  }
  return "";
}
// --------------------------------
```

MessageBox

MessageBox*(text [, flags])*

The **MessageBox** function creates, displays, and operates a message box. The message box contains an script-defined message, along with any combination of predefined icons and push buttons.

The value must be within the limit specified within the CLDATA file. If the machine is already outside the specified maximum, or if the specified axis does not exist, an error is returned.

Arguments

text - String that contains the message to be displayed.

flags - Specifies the contents and behavior of the dialog box. This parameter can be a combination of flags from the following groups of flags.

- **ppOk** - The message box contains one push button: **OK**. This is the default.
- **ppOkCancel** - The message box contains two push buttons: **OK** and **Cancel**.
- **ppYesNo** - The message box contains two push buttons: **Yes** and **No**.
- **ppYesNoCancel** - The message box contains three push buttons: **Yes**, **No** and **Cancel**.
- **ppRetryCancel** - The message box contains two push buttons: **Retry** and **Cancel**.
- **ppAbortRertyIgnore** - The message box contains three push buttons: **Abort**, **Retry**, and **Ignore**.
- **ppCancelTryContinue** - The message box contains three push buttons: **Cancel**, **Try Again**, **Continue**.
- **ppIconError** - A stop-sign icon appears in the message box.
- **ppIconWarning** - An exclamation-point icon appears in the message box.
- **ppIconAsterisk** - An icon consisting of a lowercase letter i in a circle appears in the message box.

Return Value

If a message box has a **Cancel** button, the function returns the **ppIdCancel** value if either the **ESC** key is pressed or the **Cancel** button is selected. If the message box has no **Cancel** button, pressing **ESC** has no effect.

If the function fails, the return value is zero.

If the function succeeds, the return value is one of the following values.

- **ppIdOk** - **OK** button was selected.
- **ppIdCancel** - **Cancel** button was selected.
- **ppIdAbort** - **Abort** button was selected.
- **ppIdRetry** - **Retry** button was selected.
- **ppIdIgnore** - **Ignore** button was selected.
- **ppIdYes** - **Yes** button was selected.
- **ppIdNo** - **No** button was selected.
- **ppIdTryAgain** - **Try Again** button was selected.
- **ppIdContinue** - **Continue** button was selected.

Examples (JScript)

```
MessageBox("Some information");
...
var result = "";
if (MessageBox("Do you want print tool table?", ppYesNo) == ppIdYes) {
  result += StandardResponse("%b(Generate Tool Table Command)%");
}
```

OutputExtension

Changes extension of output NC file.

Property value

text - String that contains the extension of output NC file.

> Extension must not contains following characters : \ / : * ? \ " < > |.

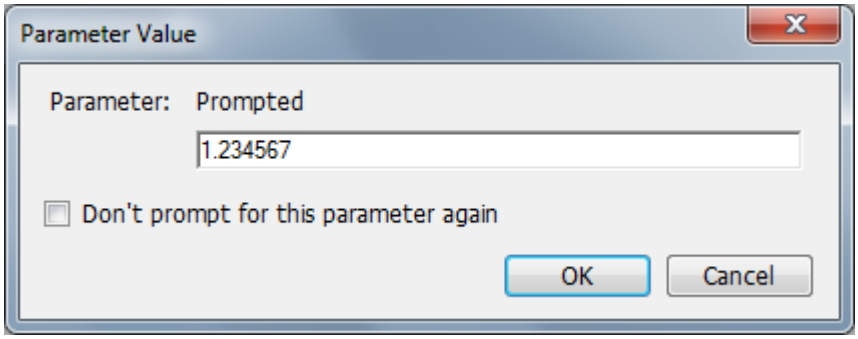
Examples (JScript)

```
// This script is called by the "Main" command.
if (GetParam("%p(Program Number)%") < 1000) {
  OutputExtension = "mpf";
}
```

PromptParam

PromptParam(*parameter[, value]*)

This function prompts the user to set the value of a parameter during postprocessing. When postprocessing takes place, **PromptParam** opens the **Parameter Value** dialog:



The user must enter the required **Value**, and then click **OK** before postprocessing can continue.

> **PromptParam** can be used to request a value when post-processing from within PowerMILL.

Arguments

parameter - The name of the parameter for which the value is to be verified.

value - Parameter value

The *value* argument is optional. It can be used to set a default value for the parameter. The **Type** depends on the parameter. Where possible, the **Value** entered by the user is converted to the type required by the parameter.

Examples

```
function User_Input() {
  //This function is called from the Start command
  var out_str = "";
  PromptParam("%p(Input Offset Value)%")
  out_str += StandardResponse("");
  return out_str;
}
```

ResetDynamicAxisLimits

ResetDynamicAxisLimits*(axis_name)*

Use this function to reset the rotary angle of the specified axis to the original value specified in the CLDATA file.

Arguments

axis_name - The axis name string

If no axis-name is specified, the function resets all axes to their original values.

ResetMachineCoordinates

ResetMachineCoodinates()

This function returns the rotary axes to their initial states as defined in the MTD-file or the option file. For example, use it to ensure all axes are reset when switching tool paths on machines that are configured to use two machine angles rather than Euler angles.

Arguments

None

Example

```
function reset_axes() {
  ResetMachineCoordinates();
  return StandardResponse("");
}
```

SetDynamicAxisInitial

SetDynamicAxisInitial*(axis_name, value)*

Use this function to set a new initial rotary angle for the specified axis. For example, this position will be used during retract and reconfigure or when you call ResetMachineCoordinates.

The value must be within the limit specified within the CLDATA file. If the specified initial position is outside , or if the specified axis does not exist, an error is returned.

Arguments

axis_name - The axis name string

value - The initial rotation for the specified axis in degrees

SetDynamicAxisMax

SetDynamicAxisMax*(axis_name, value)*

Use this function to set a new maximum rotary angle for the specified axis. For example, you can use this function when the target machine supports different axis limits for different tools.

The value must be within the limit specified within the CLDATA file. If the machine is already outside the specified maximum, or if the specified axis does not exist, an error is returned.

Arguments

axis_name - The axis name string

value - The maximum rotation for the specified axis in degrees

SetDynamicAxisMin

SetDynamicAxisMin*(axis_name, value)*

Use this function to set a new minimum rotary angle for the specified axis. For example, you can use this function when the target machine supports different axis limits for different tools.

The value must be within the limit specified within the CLDATA file. If the machine is already outside the specified minimum, or if the specified axis does not exist, an error is returned.

Arguments

axis_name - The axis name string

value - The minimum rotation for the specified axis in degrees

SetParam

SetParam(*parameter, value [, update flag]*)

This function allows you to set the values of the **Postprocessor's** internal parameters.

Arguments

parameter - The name of the parameter for which the value is to be set.

value - Parameter value.

This argument depends on the parameter type - where possible, the **Postprocessor** transforms the argument to the required type.

update flag

This argument takes one of the following integer values:

0 - Not changed (the default)

1 - Updated

2 - Not Updated

For values **1** and **2**, refer to **Output Mode** in **Initialisation Table**.

If the function is called with flag **0** (the default), then whether the parameter is considered to have been **Updated** depends on its value: if the value differs from the preceding value of the parameter, the parameter is assumed to be **Updated**; otherwise, it is not.

Example 1

```
SetParam("%p(Comment)%", "example");
SetParam("%p(Comment)%", 777);
SetParam("%p(Motion Mode)%", "RAP");
SetParam("%p(Spindle Speed)%", 5000);
SetParam("%p(Feed Rate)%", 20000, 2);
```

These are single line examples from different functions. Refer to example 2 for a complete Heidenhain machine function that illustrates the use of **SetParam** to set the **FQ** parameter according to feed rate type:

Example 2

In the following function, **FQ** is set to **1**, **2**, **3** or **4** depending on whether the feed rate set is a **Plunge Rate**, **Cutting Rate**, **Skim Rate** or **Max Rate**. The function can then be associated with the **Move Rapid, First Move after Toolchange**, **Circular Move YZ** ,**Circular Move XZ** and **Circular Move XY** commands.

JScript

```
function Parametric_Feedrates()
{
 var out_str = "";
 if (GetParam("%p(Feed Rate)%") == GetParam("%p(Plunge Rate)%"))
  SetParam("%p(FQ)%", 1);
 else if (GetParam("%p(Feed Rate)%") == GetParam("%p(Cutting Rate)%"))
```

```
    SetParam("%p(FQ)%", 2);
  else if (GetParam("%p(Feed Rate)%") == GetParam("%p(Skim Rate)%"))
    SetParam("%p(FQ)%", 3);
  else if (GetParam("%p(Feed Rate)%") == GetParam("%p(Max Rate)%"))
    SetParam("%p(FQ)%", 4);
  else
    SetParam("%p(FQ)%", 1);
  out_str += StandardResponse("");
  return out_str;
}
```

VBScript

```
Function Parametric_Feedrates
'
dim FQ
dim FR
FR = GetParam("%p(Feed Rate)%")
if FR = GetParam("%p(Plunge Rate)%") then
  FQ = "1"
elseif FR = GetParam("%p(Cutting Rate)%") then
  FQ = "2"
elseif FR = GetParam("%p(Skim Rate)%") then
  FQ = "3"
elseif FR = GetParam("%p(Max Rate)%") then
  FQ = "4"
end if
'
Call SetParam("%p(FQ)%", FQ)
Parametric_Feedrates = StandardResponse
'
end Function
```

Example 3

This function is to be associated with the **Move Linear** command. It first returns the value of parameter **Toolpath Axis Mode**. Another two parameters - **Workplane Transformation** and **Tool Length Compensation Mode** - are set according to the value of this **Mode**.

The **Move Linear** command block is appended to the output by **StandardResponse()** (with no calling argument) - therefore, the current command is called.

```
function Move_Linear_()
{
  var res = "";
  var s_axis_mode = GetParam("%p(Toolpath Axis Mode)%");
  if (s_axis_mode == "3+2" || s_axis_mode=="3axis")
  {
    SetParam("%p(Workplane Transformation)%", "ON");
    SetParam("%p(Tool Length Compensation Mode)%", "OFF");
  }
  else if (s_axis_mode=="5axis")
```

```
  {
    SetParam("%p(Workplane Transformation)%", "OFF");
    SetParam("%p(Tool Length Compensation Mode)%", "ON");
  }
  res += StandardResponse();
  return res;
}
```

StandardResponse

StandardResponse(*[command (optional)]*)

Postprocess block items of specified command and return it NC output (if no command is specified, output is returned from the current command).

- If you do not specify **StandardResponse()** on its own within the script function, the command block for the command to which it is attached is never output.
- Using **StandardResponse()** for current command (without specified command) is forbidden in functions which result is used for update block items value.
- Refer to Scripting and Examples of Scripting principles for the difference between **StandardResponse()** and **AdvancedResponse()**.

Example 1

The following is a function called **Parametric_Feedrates**, written in both **JScript** and **VBScript**. The function is called from the following commands: **Move Rapid, First Move after Toolchange**, **Circular Move YZ** ,**Circular Move XZ** and **Circular Move XY**, and outputs parametric feed rates for a Heidenhain machine using the **Qdef** function. The **StandardResponse()** function at the end of each script (**JScript** and **VBScript**) ensures the command block is output.

JScript

```
function Parametric_Feedrates()
{
  var FR = GetParam("%p(Feed Rate)%");
  var FQ;
  if (FR == GetParam("%p(Plunge Rate)%")) {
    FQ = 1;
  } else if (FR == GetParam("%p(Cutting Rate)%")) {
    FQ = 2;
  } else if (FR == GetParam("%p(Skim Rate)%")) {
    FQ = 3;
  } else if (FR == GetParam("%p(Max Rate)%")) {
    FQ = 4;
  } else {
    FQ = 1;
  }
  SetParam("%p(FQ)%", FQ);
  return StandardResponse("");
```

```
}
```

VBScript

```
Function Parametric_Feedrates
  dim FQ
  dim FR
  FR = GetParam("%p(Feed Rate)%")
  if FR = GetParam("%p(Plunge Rate)%") then
    FQ = "1"
  elseif FR = GetParam("%p(Cutting Rate)%") then
    FQ = "2"
  elseif FR = GetParam("%p(Skim Rate)%") then
    FQ = "3"
  elseif FR = GetParam("%p(Max Rate)%") then
    FQ = "4"
  end if
  Call SetParam("%p(FQ)", FQ)
  Parametric_Feedrates = StandardResponse
end Function
```

Example 2 (JScript)

This function is for association with the **Tool Change** command and its purpose is to cancel workplane tilting and zero point shift. The script variables that follow the **if** command define whether workplane tilting and zero point shift appear in the program. Where the variables are returned as **true**, the user defined commands **Cancel Tilt Plane** and **Cancel ZP Shift** are called to set the variables to **false** (using **StandardResponse()** with a calling argument).

Regardless of whether workplane tilting and zero point shift need to be cancelled, output from the **Tool Change** command blocks is appended to the preceding output (using **StandardResponse()** with no calling argument).

```
function Tool_change()
{
  var res = "";
  if (m_b_tilt_wp) {
    res += StandardResponse("%b(Cancel Tilt Plane)%");
    m_b_tilt_wp = false;
  }
  if (m_b_zp_shift) {
    res += StandardResponse("%b(Cancel ZP Shift)%");
    m_b_zp_shift = false;
  }
  res += StandardResponse();
  return res;
}
```

UserError

UserError(*error string*)

This function interrupts the postprocessing of a CLDATA file with an error (for example, if a condition is not met). When **Delcam PostProcessor** encounters this function during the processing of a command, it stops at the place the function is called, displaying a user-defined message in the **Output** window.

Arguments

error string - User defined error message

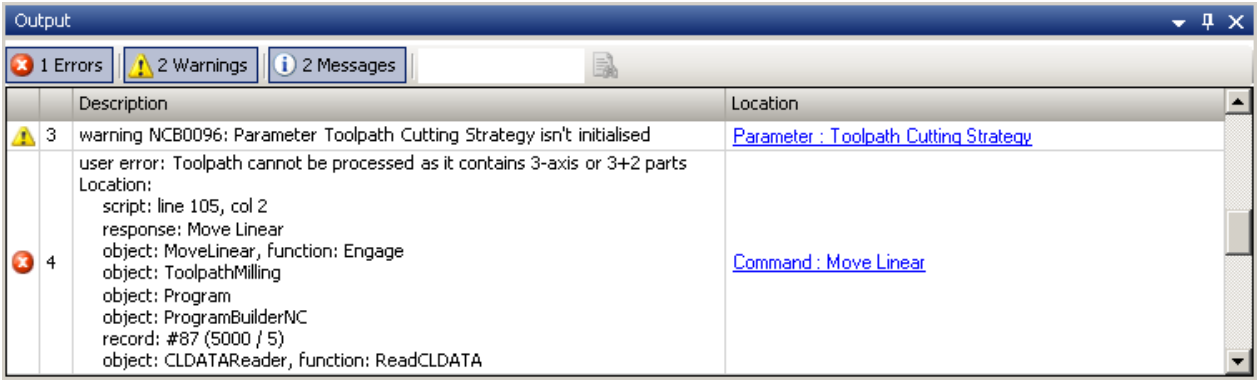
This is the error message displayed when the **UserError()** function is executed. It can contain text and other information, such as current parameter values.

Example 1 (JScript)

If you have an option file that handles only continuous 5-axis toolpaths, you can force it to interrupt processing when another mode is encountered in the toolpath by associating the following function with **Move Linear** and other relevant commands:

```
function Process_with_axis_mode_check()
{
  // Interrupt processing if toolpath is not in 5-axis mode
  if (GetParam("%p(Toolpath Axis Mode)%") != "5AXIS")
  {
    UserError("Toolpath cannot be processed as it contains 3-axis or 3+2 parts");
  }
  return StandardResponse();
}
```

For any mode other than continuous 5-axis, **Delcam PostProcessor** stops processing the CLDATA file and displays the following message when the associated commands are run:



The **StandardResponse()** function must be called **after** the possible execution of the **UserError()** function. Otherwise, incorrect NC code can be generated prior to the error message regardless of the axis mode.

The hyperlink associated with the named command allows users to jump to the command block.

Example 2 (JScript)

This example shows how to add a parameter value to the output string when generating a **UserError**.

```
function On_move()
{
  // Negative X values not allowed.
  if (GetParam("%p(X)%") < 0)
  {
    // Write down error and stop processing.
    UserError("X is negative: " + GetParam("%p(X)%"));
  }
  return StandardResponse();
}
```

If a negative **X** value is generated from the processing of a command, and the command calls the **on_move** function, then, if, for example, the command was a **Move Linear**, and **X** was returned as **-40**, **Delcam PostProcessor** stops processing and outputs the following message:

```
---------------------------------------------------------------
User error! X is negative: -40
Commands: Move Linear
Script location: line 20, pos 4
---------------------------------------------------------------
```

💡 Note that the current value of parameter **X** is present in the output.

Example 3 (JScript)

The **UserError()** function can allow a variety of data types to be returned:

```
function Error_message_test()
{
SetParam("%p(Delay)%", 123.456);
SetParam("%p(Comment)%", "my comment.");
UserError(" Test. String: " + GetParam("%p(Comment)") + " Real: " +
GetParam("%p(Delay)%") + " Number: " + 987.654 + " End Test.");
return "";
}
```

If this function is linked to the **Program Start** command, **Delcam PostProcessor** stops processing and outputs the following error in the **Output** window:

```
---------------------------------------------------------------
User error! Test. String: my comment. Real: 123.456 Number: 987.654 End Test.
Commands: Program Start
Script location: line 20, pos 4
---------------------------------------------------------------
```

Example 4 (JScript)

This example shows how the **UserError()** function can be used to handle drilling cycle setup. The conditions are:

- Multi-axis drilling cycles are not supported.
- **Helical** and **Reversed Helical** cycles are not supported.
- If the cycle setup block does not produce any output, processing stops and an error message is displayed.
- The **Break Chip** cycle has additional settings to be written into the NC program. Output of these settings is already generated by the **Break Chip Setup** command block.

The following function must be linked to all **Drilling Setup** commands, all of which must be **Active**:

```
function On_cycle_setup()
{
  var out_str = "";
  // 1. Throw error and stop processing if CLDATA contains multi-axis drilling cycle.
  if (GetParam("%p(Toolpath Axis Mode)%") == "5axis")
  {
    UserError(Multi-axis Drilling Cycle not allowed!!);
  }
  // 2. Throw error if unsupported cycles are processed.
  if (GetParam("%p(Drilling Cycle Type)%") == "HELICAL" ||
      GetParam("%p(Drilling Cycle Type)%") == "REVERSED_HELICAL")
  {
    UserError("Unsupported drilling cycle!");
  }
  // 3. Generate output for cycle setup. Throw error if it is empty
  var out_str += StandardResponse();
  if (out_str == "")
  {
    UserError("Drilling cycle setup block is empty!");
  }
  // 4. Add special code to cycle setup if "Break Chip" is used.
  if (GetParam("%p(Drilling Cycle Type)%") == "BREAK_CHIP")
  {
    // generate output for user-defined block
    out_str += StandardResponse("%b(Break Chip Setup)%");
  }
  return out_str;
}
```

What happens in each case:

1,2. If a condition is not satisfied in the first two cases, the corresponding error message appears. The name of the command where the error occurred is contained in the message as in the previous examples.

3. The **StandardResponse()** call initiates the processing of the current cycle setup block and generates output for the NC program. An error is generated if the string that is output from the block is empty.

- If error **1** or error **2** is generated, this part of the function is not processed and there is no output from case **3**).

4. This additional block is parsed if the **Break Chip Setup** command is processed.

- The "+=" operator is used to insert the additional output to **out_str**. If it is not used, output from the initial block is lost.

UserWarning

UserWarning(*warning string*)

This function generates a warning during the postprocessing of a CLDATA file (for example, when a specified condition is not met). When **Delcam PostProcessor** encounters this function during the processing of a command, it displays the user-defined warning in the **Output** window, before continuing with the postprocessing.

Arguments

warning string - User defined warning

This is the warning displayed in the **Output** window when the function is executed. It can contain text and parameter values.

Example 1 (JScript)

```
function On_load_tool()
{
  // Throw warning if tool length is excessive. Associate function with Load Tool First and Load Tool.
  if (GetParam("%p(Tool Length)%") > 100)
  {
    UserWarning("Tool length is excessive: " + GetParam("%p(Tool Length)%"));
  }
  return StandardResponse();
}
```

This function outputs a warning from the current command (for example, **Load Tool First** or **Load Tool**) when the length is in excess of **100**. It also outputs the tool length so the user can see the excess length:

```
---------------------------------------------------------------------
User warning! Tool length is excessive: 120
Commands: Load Tool First
---------------------------------------------------------------------
```

- A hyperlink is associated with the command so users can open the command block.
- After the message has been output, processing continues, and the **StandardResponse()** for the current command is always executed.

Example 2 (JScript)

The **UserWarning()** function can allow a variety of data types to be returned:

```
  function Warning_test()
  {
    // comment
    SetParam("%p(Delay)%", 123.456);
    SetParam("%p(Comment)%", "my comment.");
    UserWarning(" Test. String: " + GetParam("%p(Comment)%") + " Real: " + GetParam("%p(Delay)%") + " Number: " + 987.654 + " End Test.");
    return "";
```

```
}
```

If this function is linked to the **Program Start** command, then, when the cutfile is postprocessed, **Delcam PostProcessor** outputs the following warning in the **Output** window before completing the command:

```
----------------------------------------------------------------
User warning! Test. String: my comment. Real: 123.456 Number: 987.654 End Test.
Commands: Program Start
----------------------------------------------------------------
```

WorkplaneEuler

By default, Delcam PostProcessor uses the Euler convention defined in the **Multi-Axis Configuration** view to calculate Euler angles that are output to the NC program file. Use the **WorkplaneEuler** function to specify a different convention during processing.

To specify the convention for an Euler angle:

1. In the Script Function toolbar, click **Script Function** and select **WorkplaneEuler**.
2. In the **Angle** drop-down list, select the angle for which you want to redefine the Euler convention.
3. In the **Convention** list, select the new convention for the angle.
4. Click **Accept**.

Example (JScript)

```
WorkplaneEulerA("SXYZ")
```

Console command line

The syntax:

Console {optfile.pmopt} {input} [output] [parameters]

Keys:

optfile.pmopt	**Option File** for your machine (***.pmopt**)
input	A program file comprising ASCII or BINARY CLDATA (***.cut**) or XML-CLDATA (***.cxm**), or the name of a folder containing the files.
output	Specifies the location and name of the output file. For example: **D:\tmp\Test.tap** 💡 If no location is specified, the file is saved in the same location as the input CLDATA file. 💡 If no filename is specified, the output file is saved with the same name as the input file and with the output file extension specified in the **General** view. 💡 If a filename is specified with no extension, the output file is saved with the specified output file extension specified in the **General** view. 💡 If a filename is followed by a period (.) but no extension, the output file is saved with no extension.
parameters	You can use the following parameters: **-d** - directory mode (indicates that the **input** is from the cutfiles in a specific folder). **-t** - **output** as tapefile (indicates that the specified destination is one or more tapefiles). **performance:sax** - disable access to XML model (recommended). **performance:dom** - enable access to XML model. **-h** - show the help.

Command prompt examples

Example 1 - Postprocessing a single cutfile to the same directory

console D:\OptionFile\Fanuc.pmopt D:\CLDATA\rough_simple.cut

The cutfile **rough_simple.cut**, in the **CLDATA** folder, is postprocessed using the **Fanuc.pmopt** option file. The output file is saved in the **CLDATA** folder as **rough_simple.tap**.

Example 2 - Postprocessing a folder of cutfiles to the same directory

This example uses the **-d** argument to specify the directory from which the cutfiles are to be read.

console D:\OptionFile\Fanuc.pmopt -d D:\CLDATA

The cutfiles in the **CLDATA** folder are postprocessed using **Fanuc.pmopt**. An output file is created for each input file in the same directory. The output files are saved with the same names as the corresponding input files, but with an extension of **.tap**.

Example 3 - Postprocessing a single cutfile to a specific tapefile

This example uses the **-t** argument to specify the location and name of the output file.

console D:\OptionFile\Fanuc.pmopt D:\CLDATA\rough_simple.cut -t D:\Output\nc_file.tap

Example 4 - Postprocessing a folder of cutfiles to a specific tapefile directory

This example uses the **-t** argument to specify the location for the output files.

console D:\OptionFile\Fanuc.pmopt -d D:\CLDATA -t D:\Output

Each cutfile in the **CLDATA** folder is postprocessed using **Fanuc.pmopt**. The output files are created in the **Output** directory with the same names as the input files, but with an extension of **.tap**.

Errors

NCB0003

Error message

Minimum arc radius is greater than maximum arc radius.

Description

Possible set wrong arc configuration on Arcs and Splines page.

NCB0010

Error message

Unrecognised command name in advanced script (***value***)

NCB0011

Error message

Unrecognised parameter name in advanced script (***value***)

NCB0028

Error message

Cannot output undefined state value (***value***) from group parameter '***param***'

NCB0032

Error message

Arc linearisation tolerance value is too small

NCB0034

Error message

Wrong language identifier (***value***)

NCB0037

Error message

Cyclic nested structure

NCB0039

Error message

Cyclical links in block

NCB0045

Error message

Block number values are incorrect

NCB0054

Error message

Transformation to local workplane can't be used during continuous 5-Axis

NCB0058

Error message

Syntax error in the advanced script

NCB0059

Warning message

Type conversion has failed for '***param***'

NCB0083

Error message

Cannot define Probing Search Direction as Search Start Point coincides with Probing Target Point

NCB0084

Error message

Start block number can't be negative

NCB0085

Error message

Block increment can't be negative

NCB0086

Error message

Minimum Feed Rate can't be negative

NCB0087

Error message

Maximum Cutting Feed Rate can't be less than the Minimum Feed Rate

NCB0088

Error message

Maximum Rapid Feed Rate can't be less than the Maximum Cutting Feed Rate

NCB0089

Error message

All feed rates values are set to zero

NCB0090

Error message

Parameter '***param***' does not exist.

NCB0092

Error message

Output file isn't set.

NCB0093

Error message

Cannot use '***value***' object in advanced script. Enable access to XML model at run-time

NCB0095

Error message

Command disallowed***value value***

NCB0096

Warning message

Parameter '***param***' isn't initialised

NCB0098

Error message

Cyclical nesting of '***command***' command

NCB0099

Error message

Multi-Axis drilling cycles aren't enabled in the option file

NCB0100

Error message

Multi-Axis drilling cycle can't be processed without Automatic Coordinate Control (ACC)

NCB0101

Error message

Multi-Axis drilling cycle can't be processed. Turn ON Local Workplane for 3+2 trajectories

NCB0106

Warning message

Command inactive: ***command***.

NCB0107

Warning message

Can't change 'Read Only' parameter ***param***. Set 'Read/Write' access for current parameter.

NCB0108

Error message

Euler convention ***value*** doesn't exist.

NCB0109

Error message

Parameter '***param***' cannot be accessed as a coordinate as it has different type.

NCB0110

Warning message

Cannot set '***param***' as there is no corresponding axis in the machine kinematics.

NCB0116

Error message

Expansion of ***value*** cycles are not supported. Upgrade your CAM system.

NCB0125

Warning message

Move by Z with cutter compensation.

NCB0129

Error message

There is not string variable ***value*** in list.

NCB0130

Error message

String variables ***value*** start the loop.

NCB0132

Error message

Incorrect key name ***value*** in command SetStringVariables.

NCB0133

Error message

Script function ***value*** cannot call postprocessing of current command.

NCB0134

Error message

Parameter ***value*** is invalid. Expansion of ***value*** cycle cannot be processed.

NCB0135

Error message

'Tapping' and 'Rigid Tapping' cycles cannot be expanded.

NCB0199

Error message

User error.

NCB0207

Error message

Can't initialise ***value*** calculation engine. Machine kinematics is configured incorrectly

NCB0213

Error message

Toolvector can't be zero

NCB0219

Error message

Negative tool length is not allowed

NCB0227

Error message

Undefined Tool direction of aim position

NCB0229

Error message

Linear axis coordinate limit exceeded. ***value***.

NCB0230

Error message

Can't find linear machine coordinates to reach the aim tool position.

NCB0232

Error message

Rotary axis coordinate limit exceeded. ***value***.

NCB0233

Error message

Can't reach desired tool direction (***value***, ***value***, ***value***).

NCB0234

Error message

Can't reach desired tool direction without rotary axes (***value***, ***value***, ***value***).

NCB0235

Error message

Can't initialise ***value*** calculation engine for reduced result kinematics for the current machine state as axes is locked.

NCB0236

Error message

Retract and Reconfigure cannot be processed.

NCB0237

Error message

Can't reset axis limit ***value***=***value*** degrees during reconfiguration without a retract procedure as the interval between the limits is too narrow.

NCB0238

Error message

Impossible to use multiaxis linearization: tool directions of two neighbouring positions are anti-parallel.

NCB0239

Error message

Failed to process a singular position - NC Program Tolerance is too accurate.

NCB0240

Error message

Axis '***value***' limit exceeded: Value = ***value***; Min = ***value***; Max = ***value***.

NCB0241

Error message

Function ***value*** cannot set new dynamic limit for axis ***param*** : the value ***value*** is out of range (machine ***value***, dynamic ***value***). All data are in degrees.

NCB0242

Error message

Cannot use function ***value*** with argument '***param***' : This parameter is not linked to any machine axis of used kinematic model.

NCB0244

Error message

No Delcam PostProcessor licence (***value***)

NCB0245

Error message

No Delcam PostProcessor Multi-Axis licence (***value***)

NCB0247

Error message

Can not write output file '***value***'. The file is missed or write protected.

NCB0248

Warning message

Retract and Reconfigure has been applied at line ***value***. Please, verify correctness of the NC-program.

NCB0249

Error message

Active table object is undefined.

NCB0251

Warning message

During a connection move tool directions of two neighbouring positions are anti-parallel: machine behaviour during this move is unpredictable. Please, be careful! Start tool vector (***value***), end tool vector (***value***).

NCB0252

Warning message

Tool directions of two neighbouring positions are anti-parallel within one toolpath! machine behaviour during this move is unpredictable. Be very careful, THE MACHINE CAN BE SERIOUSLY DAMAGED! Start tool vector (***value***), end tool vector (***value***).

NCB0253

Error message

Licence error: ***value***

NCB0254

Warning message

New output extension '***value***' contains forbidden characters. Changing extension was ignored.

NCB0256

Warning message

Linearisation of Retract and Reconfigure moves has been failed.

NCB0257

Warning message

Block item (***value***:***value***) overlaps '***value***' text with '***value***'. Command: '***command***'.

NCB0258

Error message

'if(..)' condition marker expected

NCB0259

Error message

'end if' condition marker expected

NCB0617

Error message

Minimum value ***value*** for axis '***value***' is greater than maximum value ***value***.

NCB0618

Error message

Initial value ***value*** of axis '***value***' is out of range: [***value***, ***value***].

NCB0619

Error message

Axis vector for axis '***value***' has zero normal

NCB0621

Error message

Cannot load the machine kinematics. Wrong Orientation vector (***value***, ***value***, ***value***). It must be orthogonal to Tool vector (***value***, ***value***, ***value***).

NCB0641

Warning message

Feed Rate value (***value***) beyound the limit.

NCB0642

Warning message

Spindle Speed value (***value***) beyound the limit.

NCB0643

Warning message

Tool Length Compensation cannot be used for machines with rotary axes on a head for continuous multiaxis moves. It will be ignored.

NCB0644

Warning message

Local Workplane and RTCP are both switched ON. This is not normally allowed.

NCB0645

Warning message

Minimum Feed Rate value is zero.

NCB0646

Warning message

Maximum Cutting Feed Rate value is zero.

NCB0647

Warning message

%%pr(FM)%% is set to INVERSE_TIME but PostProcessor ommits its calculation when Workplane Transformation is ON.

NCB0648

Warning message

Parameter ***param*** has too small an output field (***value***) to contain the value of '***value***'

NCB0649

Warning message

param is 'ON' when 'Tool Tip' Machine Attach Point is set. Tool length is already compensated by selection of such attach point, so, Delcam PostProcessor will ignore 'ON' state of ***param*** parameter.

NCB0650

Warning message

Splines are not available in continuous Multi-Axis. Linearised moves used instead.

NCB0651

Warning message

Some characters cannot be translated in code page ***value*** and replaced by '?'

NCB0652

Warning message

Origin workplane cannot be reached by machine angles.

NCB0653

Warning message

Function '***value***' undefined.

NCB0654

Warning message

Floatig point value cannot be formatted.

NCB0655

Warning message

Used machine coordinate parameter is not linked to any machine axis.

NCB0656

Warning message

Input orientation vector is not orthogonal to tool vector

NCB0810

Warning message

Diverging tool vector (***value***; ***value***; ***value***) corrected by (***value***; ***value***; ***value***)

NCB0811

Warning message

Workplane adjusted in order to match tool vector and orientation.

NCB0814

Warning message

Parameter '***param***' cannot be recalculated.

NCB0815

Warning message

Value of 'Split Part Postfix' should contain '_%%part' mask.

NCB0816

Error message

Wrong default value for '***param***'

D0070

Warning message

Field is empty.

Description

Possible one from following fields is empty:

- Controller,
- Machine Tool,
- Decimal separator,

- Exponent String.

D0071

Warning message

Feed rate limit has zero value.

Description

All feed rate values should be greater then zero and **Maximum Cutting Feed Rate** should be equal or below **Maximum Rapid Feed Rate**.

Correct following values:

- Minimum Feedrate,
- Maximum Rapid Feed Rate,
- Maximum Cutting Feed Rate.

D0072

Warning message

Radius limit value is out of range.

Description

Correct following values:

- Minimum Radius,
- Minimum Arc Height,
- Maximum Radius.

D0075

Warning message

block start number can't be negative.

Description

Correct Number of Start Block, it should be greate then zero.

D0076

Warning message

Parameter is not initialised.

D0077

Warning message

Command must be enabled.

Description

As a minimum, the following commands should be activated and specified in each option file:

- **Program Start**
- **Load Tool First**
- **Load Tool**
- **Move Rapid**
- **Move Linear**
- **Circular Move YZ**, if the **Arcs** option is selected for Arcs and Splines.
- **Circular Move XZ**, if the **Arcs** option is selected for Arcs and Splines.
- **Circular Move XY**, if the **Arcs** option is selected for Arcs and Splines.
- **Spline Move**, if **Spline Configuration Output** is set to **Polynomial**.
- **Program End**

D0078

Warning message

command '{0}' must contain parameter '{1}'.

D0079

Warning message

only one format is defined.

Description

Usually option files uses different formatting styles for print values of paramenets, but this uses single format for all. You may need to create new format and assign it to desired parameters or just ignore this warning.

D0080

Warning message

Group parameter contain less than 2 states.

Description

Group parameter uses as switch and should two and greater states.

D0081

Warning message

State of group parameter isn't defined.

Description

Output value for state is not defined, that mean when **Group Parameter** will be in this state you will get empty output.

D0082

Error message

undefined error

D0083

Error message

'if(..)' condition marker expected

D0084

Warning message

Command 'name' isn't defined.

Description

In advanced script you want to get output of command, but it name is not defined. Possible, there is a misprint in name or you need to create new command.

D0085

Warning message

command ID is no longer valid.

D0086

Warning message

Description

In advanced script you want to assign new value for read-only parameter. Possible, you need to change access to parameter.

D0088

Warning message

function name() has incorrect argument.

D0089

Warning message

command must contain parameter

D0090

Warning message

parameter ID is no longer valid.

D0091

Warning message

parameter with name 'name' isn't defined.

Description

In advanced script you want to use parameter, but it name is not defined. Possible, there is a misprint in name or you need to create new parameter.

D0092

Warning message

block number values are incorrect.

Description

Block Number configuration should match following rules:

- **Number Of Start Block** should be greater than zero.
- **Maximum Block Number** should be greate than **Number Of Start Block**
- **Block Increment** should be greater than zero.

D0093

Warning message

block increment must be positive.

Description

Correct value of Block Increment, it should be greater than zero.

D0094

Warning message

'Feedrate Value' can not be less than 'Minimum'.

D0095

Error message

'end if' condition marker expected

D1001

Error message

Machine Axis: Direction cannot be Null.

D1002

Error message

Machine Axis: Maximum Limit cannot be less than Initial Value.

D1003

Error message

Machine Axis: Initial Value cannot be less than Minimum Limit.

D1004

Error message

Machine Axis: This Axis sould not be collinear to Initial Tool Vector.

D1005

Error message

Simplex solver is used for this kinematics: number of linear axes aren't equal 3.

D1006

Error message

Simplex solver is used for this kinematics: linear axes aren't orthogonal.

D1007

Error message

Simplex solver is used for this kinematics: linear axes aren't right-handed.

D1008

Error message

Simplex solver is used for this kinematics: rotary axes are collinear.

D1009

Error message

Head: Orientation vector is null.

D1010

Error message

Head: Orientation vector isn't orthogonal with initial tool vector.

D1011

Error message

Simplex solver is used for this kinematics: axes order in kinematics chain not standard.

D1012

Error message

Last Rotary Axis is collinear to Initial Tool Vector. The rotary angle can only be controlled by input Orientation Vector.

Appendix

Glossary

Auxiliary Option Files

Auxiliary option files allow to produce more than one option file in one go. This feature is especially useful for processing of main and sub-programs. The auxiliary files are absolutely independent on its master. However, they are stored inside the master option files.

In order to add an auxiliary option file, you may create it from scratch or attach an existing option file. For the first option, click "New Auxiliary Option File" in the context menu of the master option file in the Session Tree. Then enter the name of the file and follow to the Editor to fill it in. If you prefer to load an existing option file, click "Load Auxiliary Option File" and choose an appropriate one.

If you would like to change an auxiliary file, it must be activated first. To do that, double click on the file item in the Session Tree or click "Activate" from its context menu. After activation follow to the Editor and do the changes you need.

Inverse Time Feed Rate

Most NC programmers think of the F-register in a CNC controller as the method for specifying linear velocity. This is true for two- and three-axis linear motion, but when rotary motion is to be controlled, the F-register takes on a different meaning. When combined linear/rotary motion exists, most good CNC controllers require the inverse of the amount of time necessary to make the move, and since each move has a different distance, the corresponding time varies for each block as well. The exact reasoning behind using the inverse value rather than the direct time in minutes or seconds is simply a historical matter.

The constant used to calculate the inverse time code is normally 1 minute, such that the equation is:

F(code) = 1(minute) / (time = 3D distance/velocity)

The 3D distance of the move is calculated in model coordinate space at the NC control point, not in machine coordinate space and not necessarily at the tool tip. For example, a 5-inch move at 50 IPM takes 5 50ths of a minute, yielding an inverse time calculation of 1/.1 and an F-code of F10. The same 5-inch move at 700 IPM would be 1(minute) / (time = 5 / 700) or (1/(5/700)) or (1/.0071428) or F1400.168

Escape sequences

To include any of the following special characters in a string, use the associated escape sequence:

Character	ASCII Representation	ASCII Value	Delcam PostProcessor 6.0 Escape Sequence
New Line	NL (LF)	10 or 0x0a	\n
Horizontal Tab	HT	9	\t
Vertical tab	VT	11 or 0x0b	\v
Carriage Return	CR	13 or 0x0d	\r
Form Feed	FF	12 or 0x0c	\f
Back Slash	\	92 or 0x5c	\\
Question Mark	?	63 or 0x3f	\?
Single Quotation Mark	'	39 or 0x27	\'
Double Quotation Mark	"	34 or 0x22	\"
Hexadecimal Number	hh		\xhh

For example, type **Line 1\nLine 2** into a text string to generate the following output:

Line 1

Line 2

In previous versions of **Delcam PostProcessor**, the following escape sequences were used:

Previous Escape Sequence	Character Representation	Replacement
;cr	CR LF	\r\n
;amp	&	&
;apos	'	\' or '
;quot	"	\" or "
;lt	<	<
;gt	>	>
;bcktick	`	`
;bar	|	|
;;	;	;

When you open an existing option file that contains any of the above sequences, **Delcam PostProcessor** automatically updates them to the replacements shown.