

Basic configuration of the CNC-Calc post processor using the Global variables section

One of the new features in CNC-Calc ver. 7 is the addition of a Post Processor. This makes it possible for the end user to format the created NC program.

The Post Processors are written in Java Script. Java script was selected because it made it possible to define variables with both global and local scopes. It also gives the end user the possibility to declare and define functions that can be used in formatting and calculation. The writer of the post processor has access to all standard mathematical functions defined in standard Java Script and all the standard features that this language contains.

CNC-Calc and the post processor cooperate to format and create the correct NC program. This cooperation is established with various functions that are called from CNC-Calc with the necessary parameters. The parameters are then used to calculate and output one or more lines of code in the NC program.

At the start of the Post Processor there is a section called Globals. This section contains variables that enables the end user to configure a post processor to handle the most common setup changes.

The following table lists these variables and describes how they affect the output:

Name	Normal Values	Description
decimalMark	',' Or ','	This variable defines the decimal mark that should be used when decimal values are shown.
linebreak	"\n"	Defines the character sequence that should be used to terminate the lines in the NC program.
variableDelimiter	" "	When more than one variable is shown in a single line the variableDelimiter defines how they should be separated.
tolerance	0.02	The tolerance is normally used in the user program to define the smallest entity that should be handled. If for instance a very small circular movement is made, the controller may mistake this as being a 360 degree movement.
showSequenceNumbers	true/false	If showSequenceNumbers is set to true, all blocks in the NC program will be formatted with a line number. This is used in the function writeBlock that can be modified by the programmer.
sequenceNumberStart	10	If showSequenceNumbers is set to true sequenceNumberStart will define the starting number used for the first block. This is used in the function writeBlock that can be modified by the programmer.

sequenceNumberIncrement	5	sequenceNumberIncrement is only used if showSequenceNumbers is set to true. The sequenceNumberIncrement defines the jump in block numbers between blocks. This is used in the function writeBlock that can be modified by the programmer.
useRadius	true/false	Circular moves are normally written using the R (radius) or I, J, and K values (center coordinates). When useRadius is set to true circular moves it will use the R value. This is used in the function onCircular that can be modified by the programmer.
absoluteArcCenter	true/false	If useRadius is set to false, the circular moves will use the I, J, and K values. The value of absoluteArcCenter determines whether these values should be given as absolute center coordinates or relative to the start of the move. This is used in the function onCircular that can be modified by the programmer
xDiameterProg	true/false	This field is only in post processors for lathes. The value of xDiameterProg determines whether the X values should be output as diameter values. If it is set to true, the X values will be output as diameter, otherwise they will be output as radius.
iDiameterProg	true/false	This field is for post processing for lathes only. If useRadius is set to false, the circular moves will use the I, J, and K values. The value of iDiameterProg indicates how I values of circular moves should be formatted. If iDiameterProg is true, the I values will be given as diameter values, otherwise they will be given as radius. This is used in the function onCircular that can be modified by the programmer

In the Globals section it is thus possible to take one of the post processors included with CNC-Calc and make it comply with a given machine.

How to set up backplot to reflect the settings in the Globals section

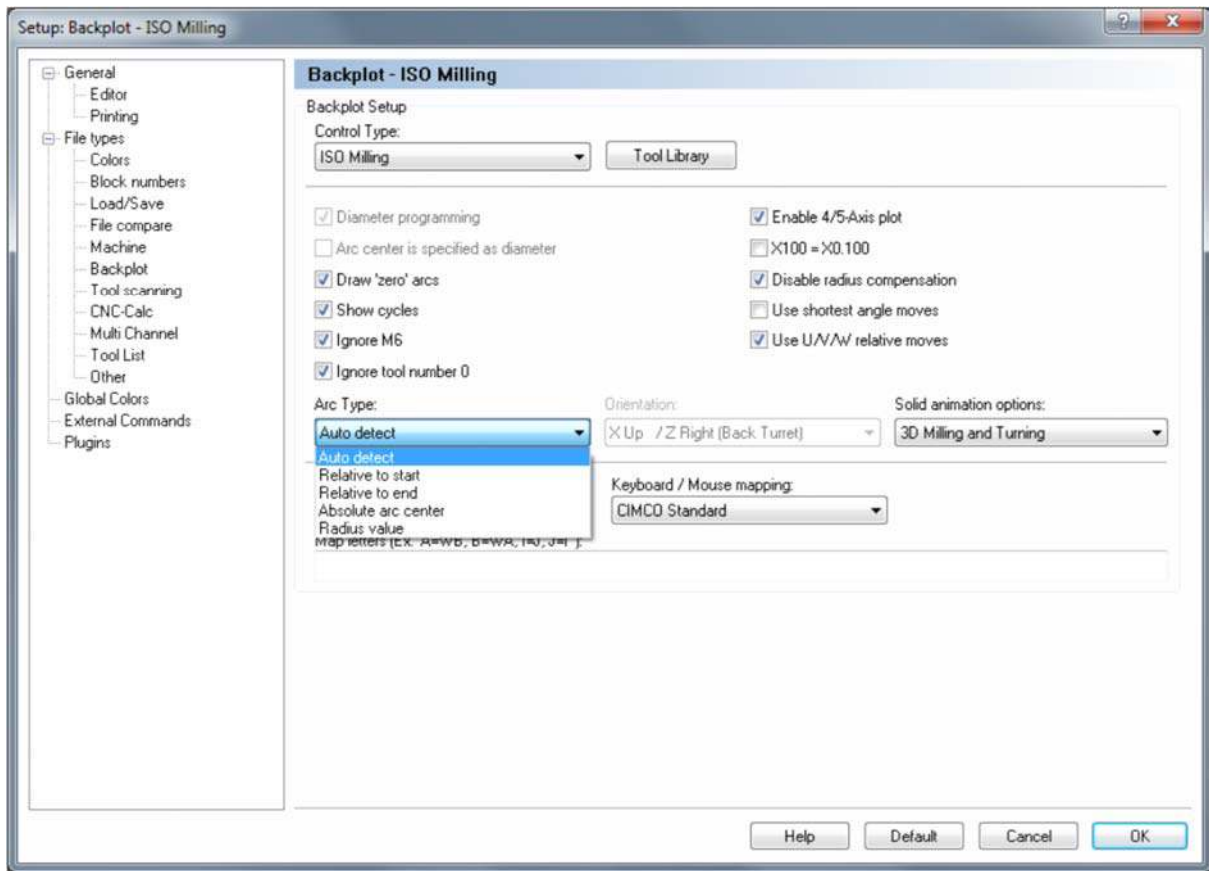


Figure 1 ISO Milling Setup Backplot

Figure 1 shows the Backplot Configuration for ISO Milling, and we will now see how we need to change the setup to reflect the variables in the Globals section.

The only part of the milling setup that can be influenced by the variables in the Globals section is the Arc Type dropdown. The Arc Type is the dropdown that is shown as dropped in Figure 1. The choice in this dropdown depends on the Globals variables **useRadius** and **absoluteArcCenter**, and in the table below the values of the variables and the corresponding Arc Type are shown.

Variable values	Correct Arc Type Selection
useRadius = true absoluteArcCenter = false	Radius Value
useRadius = true absoluteArcCenter = true	Radius Value
useRadius = false absoluteArcCenter = false	Relative to Start
useRadius = false absoluteArcCenter = true	Absolute Arc Center

In turning there are two more values in the Globals section that will influence the setup of the backplot. Figure 2 below shows the backplot configuration for ISO Turning.

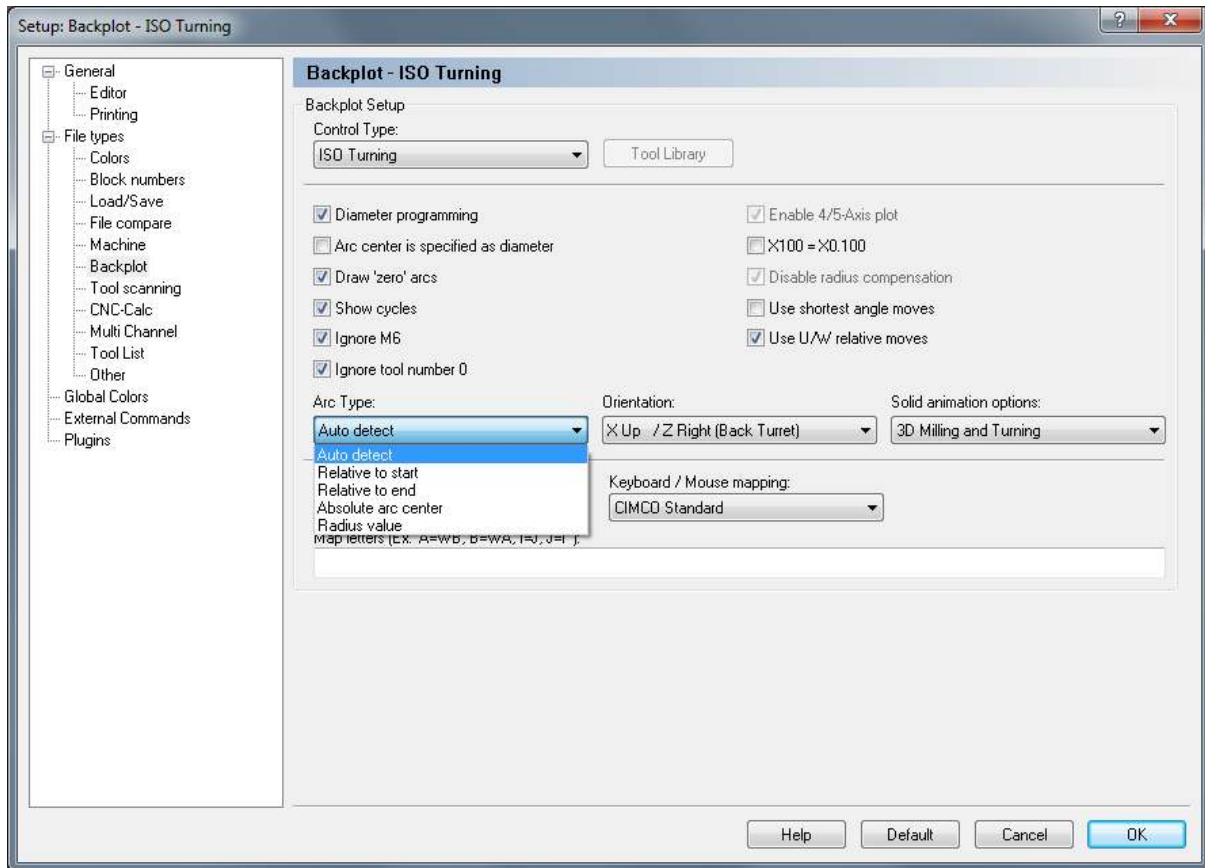


Figure 2 ISO Turning Backplot Setup

The ISO turning Arc Type should be selected from the Arc Type dropdown based on the Globals variables **useRadius** and **absoluteArcCenter**, exactly as described under milling. When looking at turning we also have to consider the possibilities regarding diameter programming. In the post processor, the diameter programming are handled by the Globals variables **xDiameterProg** and **iDiameterProg**. These variables correspond to the fields *Diameter programming* and *Arc Center is specified as diameter*. The following table shows how the values of the Globals variables correspond to the fields in the turning setup.

Variable Values	Correct Backplot Setup
xDiameterProg = false iDiameterProg = false	<i>Diameter programming</i> is unchecked <i>Arc Center is specified as diameter</i> is unchecked
xDiameterProg = false iDiameterProg = true	<i>Diameter programming</i> is unchecked <i>Arc Center is specified as diameter</i> is unchecked
xDiameterProg = true iDiameterProg = false	<i>Diameter programming</i> is checked <i>Arc Center is specified as diameter</i> is unchecked
xDiameterProg = true iDiameterProg = true	<i>Diameter programming</i> is checked <i>Arc Center is specified as diameter</i> is checked

If the Arc Type dropdown selection and the checkboxes *Diameter programming* and *Arc Center is specified as diameter* are checked respectively unchecked as described above, the backplot should show the correct tool paths.