Run Time Calculation for AUTO-CODE Users

December 2001 - Requires version 4.2 of AUTO-CODE

Introduction

This document describes the Run Time Calculations available for AUTO-CODE users. These consist of the output file NCCRUN.DAT, the control points used by the calculation feature, and the operation of the run time calculation report generator supplied with AUTO-CODE.

If you have customized your AUTO-CODE to make use of the previous run time data file, NCGDST.$$$ you will need to update that part of the interface as the $$$ file is no longer created during the translation phase.

If you would like to take advantage of these new features in an existing interface, the code that must be added can be found in the most current example interfaces supplied with AUTO-CODE. There are specific control points, detailed in this document, which must be inserted for the run time calculator to function properly. Code examples are also provided in this document for inserting the control points into your application.

New posts created by AUTO-CODE/KCI after December 2001 will include the control points needed for reporting purposes. Since the data messaging system is text based, alternative uses may be made of the control point contents for specific requests.

Note that the report generator uses the file NCCRUN.DAT that contains calculated distance and tool usage information related to the last NC program created by the AUTO-CODE translator.

The remainder of this document contains the following items.

Sub-Title	Contents
What are Run Time Calculation Tools	Description of run time calculations that can be performed by CAD/CAM systems and their applications.
Using the Run Time Calculator	Usage of the program modules for output of the run time data.
NCCRUN.DAT File Format	Detailed explanation of the NCCRUN.DAT file created each time the translator is run. For those wishing to customize the report generator.
Control Points	List of control points used for the NCCRUN.DAT output and subsequent reports generated. For those wishing to customize an interface to provide support for the run time calculator.
Interface Updates	Example Visual LISP code that outputs the control points needed for the run time calculator to function best.

What are Run Time Calculation Tools?

Run time calculation tools can be used to estimate the time that will be required to cut a part. Some run time calculations simply provide distance traveled by the tool while others have more sophisticated mechanisms for determining the actual run time. Obviously these simulations are just simulations and may not always reflect the real world. Tool breaks, difficult to handle materials, delivery problems, and many more time consuming activities cannot be accurately predicted thus any calculation should be considered just an educated guess.

The information obtained from a run time calculator is typically the average velocity of the cutting tool multiplied by the distance traversed. Thus if a tool is cutting at an average speed of 20 inches per minute and must cut a path that is 200 inches long, the time required is 10 minutes (200 divided by 20). But this time is only for the cutting activity and does not include any setup time, off load time, rapid movement to and from the home position, tool change time, and so on.

Other information that can be of use is the distance of cutting for a specific tool. This value can be used for tool life maintenance. Tool life maintenance is where you keep track of how long a given tool has been used and based on the history of similar tools, you can schedule replacements or sharpening. Obviously this information is very shop and material specific. AUTO-CODE's run time calculator merely reports the distances

For run time calculations to work properly in AUTO-CODE you must define a part program and then generate it. Immediately after generating the output file the run time calculations are available for the program just created. When another program is created, the run time data for the previous program is overwritten unless saved using the AUTO-CODE Run Time Calculation tool.

Other limitations that run time calculation tools may face are related to the special cycles of a machine tool. Advanced pocket and rough cutting cycles, special taps, and other features of the machine tool will most likely not be recognized in their proper context. The only exception would be a run time calculator that is written specifically for the machine tool. There are simply too many variations for a general purpose tool to be effective in that regard.

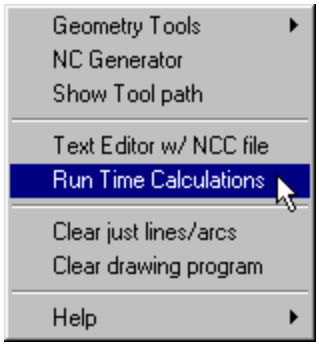
In AUTO-CODE, the Run Time Calculation Tool is a report generator. There are two parts to the tool. One that happens automatically as the system is run and another that operates under the guidance of the user.

As the AUTO-CODE translator runs it accumulates the total travel distance, tool usage, and saves project-related data (part number, program number, comments, and description). This data is output to a simple ASCII text file named NCCRUN.DAT. The data file is located in the AUTO-CODE program directory.

The second part of the run time calculator is under the control of the operator. An application named NCGPP.VLX has been supplied with AUTO-CODE. This application reads the NCCRUN.DAT file and created formatted reports in the drawing, to a text file, to a spread sheet, or in a dialog box on the screen.

Using the Run Time Calculator

Start the AUTO-CODE Run Time Calculator from the AUTO-CODE pull down menu immediately after creating a NC/CNC file. Each time the "Generate NC" button is used in AUTO-CODE, the run time data is overwritten unless saved by you first. You have several options for saving the data from the Run Time Calculator. Data can be written to the current drawing as text, output to an ASCII text file suitable for human reading, output to a dialog box for you to view the completed calculations, or sent to an Excel spreadsheet (you must have Microsoft Excel installed for this option to work).



Immediately after translation using the "Generate NC" button, select the menu option for the Run Time Calculator.

This will start the Run Time Calculations utility in AUTO-CODE. Note that this utility only works with the most recently created NC/CNC file and should only be run after running the NC Generator and selecting the "Generate NC" button to insure that the results are for the program currently being displayed.

A dialog box will be displayed showing the primary options for the Run Time Calculations output. The options for output are to display the results in a dialog box (the default), send the results to a text file, add the results as text objects in the current drawing, or send the data to an Excel spreadsheet. You should start by viewing the results in the dialog box to see if they appear proper. After you are happy with the results, generate the output in a more permanent format.

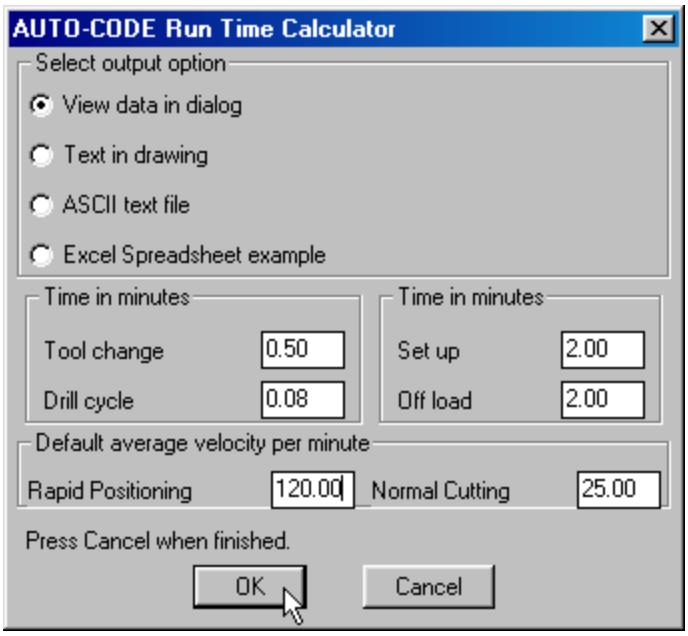


Figure 2: Primary Dialog Box

The primary dialog box of the AUTO-CODE Run Time Calculator is shown in figure 2. Six variables are provided that will modify the time report. The first two are the times required for performing a tool change and for a drill cycle. These values should be estimated conservatively and based on observation of the machine tool in action. If your process does not involve tool changes or drilling cycles, then these values will be ignored.

The set up and off load times will vary for the types of parts being created. If the material is very heavy or large and bulky, but results in many small pieces that can be removed quickly, the values will not be equal. The values to use should be based on observation and be conservative so that the resulting totals are more in line with reality.

The last two values to be supplied are the default average velocities. AUTO-CODE's run time calculator can work with variable feed rates, but only if proper values are supplied during the program creation. The values in these two dialog boxes override all other feed rates that may have originally been used. They are the last known feed values during translation by default. If your interface supplies feed rate values of the proper type, then do not modify these fields as any variable feed rate information will be lost.

When you are happy with your selections and input parameters, select OK to generate the output desired. When finished, select Cancel to exit the calculations utility dialog box and return to the AutoCAD command prompt. Figure 3 shows a sample output to the dialog box for a simple program.

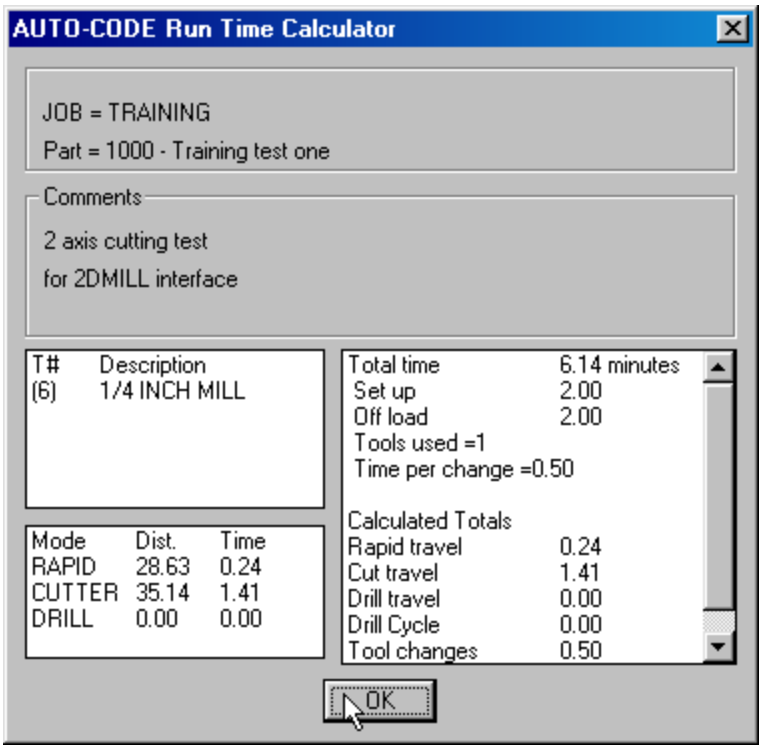


Figure 3: Sample output from Run Time Calculator to dialog box.

Using an Excel Spreadsheet

The output of the run time calculator can be directed to an Excel spreadsheet. When this option is selected the program will ask for the name of a template file to use. The template file is a spreadsheet that has been formatted to include defined names. The names are used to locate the cells where the data from the run time calculator is to be output.

A sample spreadsheet has been provided that can be manipulated to create the report you want. The spreadsheet, named **ACODEMO.XLS**, contains both text and variable names in predefined locations. You can move these to anyplace in the spreadsheet you desire using the normal cut and paste operations in the Excel spreadsheet editor. To relocate a named cell, first move the pointer to cell you wish to move, right click the pointer and select the "Cut" option, then move the pointer to the desired location, right click again and select "Paste".

You can create your own spreadsheet using the variable names from the following table to interface with the existing variable names. Select a cell where you want to insert a value from AUTO-CODE and then pick the "Insert" pull down menu. The "Name" fly out menu is selected next from which the "Define..." option is our target. This will result in a dialog box being displayed and you can type the name of the variable to be used. Note that names are case sensitive in this context thus you must use upper and lower case characters as they appear in the following table.

Variable Name	**Contents**
FileName	Name of output NC/CNC file
FileSize	Size of file in bytes
FileDate	Date and time file was created
PartNumber	Part number data associated with XN control point.
PartDesc	Part description data associated with DE control point
JobNumber	Job number data associated with the XJ control point.

ToolChanges	Number of tool changes
ToolChangeTime	Tool change time (per tool average)
SetupTime	Machine preparation time
OffloadTime	Machine completion time
MachineTime	Sum of machine movement times
TotalTime	Total of all times
CycleTime	Drill cycle time (per cycle)
DrillCycleTime	Sum of all drill cycle times
RapidTime	Time spent in rapid mode total
RapidDist	Distance traveled in rapid mode
CutTime	Time spent in cut mode total
CutDist	Distance traveled in cut mode
CycleDist	Distance traveled in cycle mode
CyclePosTime	Time spent positioning in cycle mode
Comment#	Comments (# is integer 1, 2, ...)
Tool#	Tool number, # is station number
Point#	Number of points hit by Tool#
Dist#	Distance traveled in cut mode by Tool#
Time#	Time of cutting by Tool#

Modifications to the Run Time Calculator

The source code for the Run Time Calculator is provided at the AUTO-CODE web site for those wishing to customize the program further to fit their unique needs. AUTO-CODE Mechanical and qualified consultants can also provide custom programming for a fee. You must know Visual LISP in order to do the customization work since the entire program is written using that language.

NCCRUN.DAT File Format

The NCCRUN.DAT file is formatted for easy access in Visual LISP. It is an ASCII text file with each line representing a set of data. There are two basic types of data lines in the file. Lines that start with a string value are assignments for a particular variable. Lines that start with a number are associated with tool usage.

Variable assignment lines fall into two categories. Those associated with the program in general have a single string following the variable name. Those associated with movement styles such as rapid point to point or cutting a contour have two numbers following.

Variable	**Description**
FILE	File name of NC output.
CFG	File name of the CFG used for translation.
JOB	Job name string as supplied by operator. From control point 'XJ'
PART	Part number string as supplied by operator. From control point 'XN'
DESC	Description string as supplied by operator. From control point 'XD'
FEEDRATE	Default feed rate (last used) for cutting. From control point 'FC'
RAPIDRATE	Default rapid feed rate (last used). From control point 'FR'
MACHDESC	Comment line from CFG file for translator.
COMMENT	Comment string as supplied by operator. Can have multiple comment lines in a program. Control point 'XC'
RAPID	Movement data record for rapid point to point travel.
CUTTER	Movement data record for cutting travel.
DRILL	Movement data record for drill or special cycle travel.
Number	Tooling data record.

Movement Record	**Usage**
1	Name of movement style (Rapid, Cutter, Drill...)
2	Total distance traveled in movement style
3	Calculated time for movement in that style
Tooling Data Record	**Usage**
1	Tool ID number (integer) from control point 'TC'
2	Description (string) from control point 'TT'
3	Distance in cutter mode
4	Calculated time for movement in cutter mode
5	Number of points visited (drill or punch hit points)

The source code for the AUTO-CODE run time calculator can be obtained at the AUTO-CODE web site. It is a Visual LISP program and associated dialog box file that can be compiled into a VLX file for easier distribution. The function READ_NCCRUN_FILE can be copied from the source code and used in other applications of your own design. This function will read the data file into a set of data lists, which can then be manipulated with ease in Visual LISP (see functions CheckMoves, CheckToolData, and ComputeTimeValues in the same project as the READ_NCCRUN_FILE function.)

The following examples show the type of data that is output to the NCCRUN.DAT file. The first example demonstrates the type of output you will see from a translation using an interface that does not contain any of the enhanced process planning control points. All the variable assignment data is blank (except for the file name and data obtained from the CFG file) and only the travel distances are reported for the two cutting modes.

Example 1: DAT file with no project data and no tooling.

```
"FILE" "C:\Program Files\AutoCAD 2000i\demo .NCC"
"CFG" "C:\Program Files\AutoCAD 2000i\NCG42/flame.cfg"
"JOB" ""
"PART" ""
"DESC" ""
"MACHINEDESC" "F:Flame cutter"
"FEEDRATE" 5.0
"RAPIDFEED" 100.0
"RAPID" 20.602 0.000000
"CUTTER" 16.76 0.000000
```

Adding the control points needed for the project information, the next example (#2) shows a more complete output file. In this case the feed rate for cutting was one inch per minute and the feed rate for rapid traverse was 20 IPM. Job and part number information was supplied during input of the header information and saved as special control points in the function NCG_RUN_FIRST for a custom interface.

Example 2: DAT file with project data but no tooling.

```
"FILE" "C:\Program Files\AutoCAD 2000i\demo .NCC"
"CFG" "C:\Program Files\AutoCAD 2000i\NCG42\wire.cfg"
"JOB" "1000"
"PART" "A-12"
"DESC" "TEST DIE"
"MACHINEDESC" "W:Wire EDM"
"COMMENT" "Wire Example"
"FEEDRATE" 1.0
"RAPIDFEED" 20.0
"RAPID" 20.602 1.0301
"CUTTER" 16.76 16.76
```

Example 3 is a DAT file for a translation that involved an interface with the enhanced control points for project data, feed rates, and tooling assignments. Data for the tool output is obtained from the tool change function NCG_CHANGE_TOOL in a custom interface. The tool ID number is required to be the unique tool station number while the description data can be anything desired.

Example 3: Filled out DAT file.

```
"FILE" "C:\Program Files\AutoCAD 2000i\DEMO3.NCC"
"CFG" "C:\Program Files\AutoCAD 2000i\NCG42\2dmill.cfg"
"JOB" "X-15"
"PART" "1000"
"DESC" "Simple Pocket Demonstration"
"COMMENT" "Drill guide holes first"
"COMMENT" "Then mill with half inch"
"COMMENT" "Finish with quarter inch"
"MACHINEDESC" "M:2d Mill"
"FEEDRATE" 30.0
"RAPIDFEED" 120.0
"RAPID" 58.650 0.488
"CUTTER" 511.285 17.163
"DRILL" 7.973 0.621
2 "1/2 INCH DRILL" 7.973 0.621 2
5 "1/2 INCH MILL" 369.850 14.408 328
6 "1/4 INCH MILL" 141.439 2.755 71
```

Control Points

The contents of the DAT file are supplied via control points from the user interface. The starting function (NC_START_UP) can be customized to include the codes needed for the job, part number, and so on. The function (NC_TOOL_CHANGE) is then used to supply the control codes for tool selections.

Variable assignments - the last one used in the file will be used in the actual report. That is, if two copies of the same control point exist, the last occurrence will be the only one used.

Code	Type	Value represents
"FA"	Real	Feed rate multiplier for arc speed (expressed as 0<x<=1.0) *
"FC"	Real	Feed rate cutting in units per minutes (IPM, MMPM) *
"FF"	Integer	Set the automatic feed rate output, for just calculations set to zero. *
"FR"	Real	Rapid speed in units per minute (IPM, MMPM) *
"FZ"	Real	Feed rate cutting in the Z direction in units per minute *
"TC"	Integer	Tool code number *
"TT"	String	Tool descriptive text
"XC"	String	Comment string, each occurrence added to a list
"XD"	String	Job description, only last encountered value is used
"XJ"	String	Job number or name, only last encountered value is used
"XN"	String	Part number or name (if part of an assembly), only last encountered value is used

* Note that the feed rate and tool code reference control points have been in the AUTO-CODE system for several releases but are included here again for reference.

Most of these control points are inserted during the STARTUP pattern or NCG_START_UP function. If the machine interface does not involve variable tools then the feed rate values should also be defined during

the start up process. They can also be placed in the AUTO_IN and AUTO_OUT patterns so that the most recent value from the machine settings is used each time.

The tool code, description, and feed rates are output during tool changes in the TOOLCHG pattern or NCG_TOOL_CHANGE function. If the TC and TT values are encountered during translation then tool detail records will be output to the NCCRUN.DAT file (as in example 3 above). There should only be one TC and TT pair for a tool change.

Interface Updates

The following shows how to update the 3DMILL interface of AUTO-CODE to include the process planning data. The functions NCG_START_UP and NCG_TOOL_CHANGE are modified to include the following new lines of code.

Additions to NCG_START_UP added just after making the variable assignments based on the input from the original insertion point. The symbols JN (Job Number), DE (job Description), PN (Part Number), CM1, CM2, CM3 (Comments 1 through 3), NCG:FD (Cut feed rate), NCG:FDR (rapid travel rate), NCG:FDZ (Z axis feed rate - optional) are all assumed to have been set to proper data values before this code segment runs.

```
     (NCG_INS NCG:KP "XJ" JN)
     (NCG_INS NCG:KP "XD" DE)
     (NCG_INS NCG:KP "XN" PN)
     (if (/= CM1 "")
       (NCG_INS NCG:KP "XC" CM1))
     (if (/= CM2 "")
       (NCG_INS NCG:KP "XC" CM2))
     (if (/= CM3 "")
       (NCG_INS NCG:KP "XC" CM3))
     ;;
     (NCG_INS NCG:KP "FF" 0)
     (NCG_INS NCG:KP "FC" NCG:FD)
     (NCG_INS NCG:KP "FR" NCG:FDR)
     (NCG_INS NCG:KP "FZ" NCG:FDZ)
     (NCG_INS NCG:KP "FA" 1.0)
```

Additions to the subroutine NCG_TOOL_ CHANGE, add at beginning of module.

```
  (NCG_INS NCG:KP "TC" NCG:CTOOL
  (NCG_INS NCG:KP "TT"
    (strcat NCG:DESC ;;tool description
            " DIA=" (rtos NCG:TSIZE) ;;Tool size
            " OFF=" NCG:TOFF ;;Offset register
            " LEN=" NCG:TLEN ;;Tool length
            ))
  (NCG_INS NCG:KP "FC" NCG:FD)
  (NCG_INS NCG:KP "FZ" NCG:FDZ)
  (NCG_INS NCG:KP "FR" NCG:FDR)
```

By adding these control points the process planning output file will be populated. If these control points do not exist in an AUTO-CODE drawing then the NCCRUN.DAT file will contain blank entries for this data.

More examples are available by looking at the sample interfaces provided with AUTO-CODE.