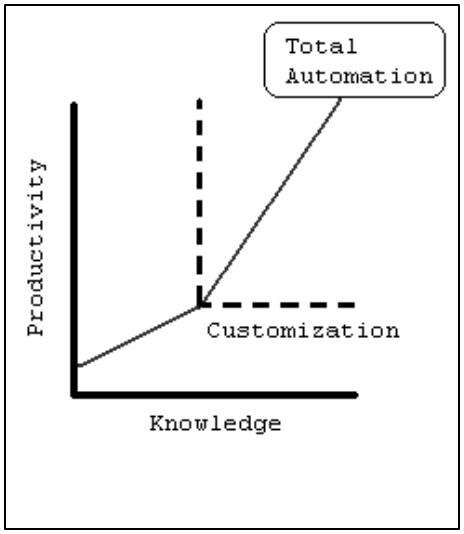
AUTO-CODE NC-Generator Concepts and Theory

This chapter discusses the AUTO-CODE™ software in relationship to the AutoCAD® graphics system. Because AUTO-CODE runs inside AutoCAD, the way you create drawings is critically important to the success of the system. Topics covered include how to set up a drawing for the machine tool interfaces supported by AUTO-CODE, the basic theory of operation, various methods to start the programming system, and a series of frequently asked questions with answers.

AutoCAD and AUTO-CODE

The most important concept to understand is that AUTO-CODE operates inside AutoCAD and follows many of the same philosophies. As such, the more you know about AutoCAD the better you will be at using AUTO-CODE. Certain objects (circles, lines, arcs, and so forth) from the AutoCAD system are utilized to define tool paths. Your ability to work with the AutoCAD objects involved will greatly influence how well you can use AUTO-CODE to get the NC/CNC programming task accomplished. At the same time, you do not need to be an AutoCAD expert to productively use AUTO-CODE. You only need to master a few concepts. However, the more you do learn about AutoCAD, the more you will be able to do - including customizing AUTO-CODE and AutoCAD to build an automated design-to-manufacturing system.

Like AutoCAD, the AUTO-CODE software assumes you know what you are doing. AutoCAD does not teach you how to draw engineering documents nor does it teach proper drafting techniques. It is still a very powerful tool for engineers and drafters. AUTO-CODE will not teach you how to program a machine tool. It is a powerful tool for machinists and mechanically inclined engineers to use AutoCAD graphics in the creation of machine tool programs.



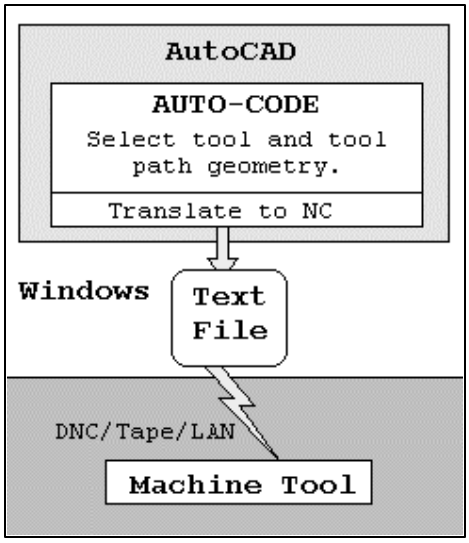
As shown in the figure at left, the more you know about AutoCAD and AUTO-CODE the more productive you will be. If your knowledge level increases to the point where you can customize the AutoCAD and AUTO-CODE systems, then the productivity can go through the roof with the ultimate goal being total automation.

Total automation means the drawing and NC/CNC programming operations take place automatically. There are several concepts that can be employed to achieve this goal with the most common being a parameter-based solution. Parameter-based solutions use variables to describe the size (and sometimes the shape) of the part. The parameter values can be obtained from any number of sources, including an ordering system, an engineering configuration system, a simple spreadsheet and more. Once the parameters are known, the graphics can be regenerated and an automated version of AUTO-CODE put to work on it. Since the drawing standards would be clearly defined, it is conceivable for the NC/CNC generator to work completely unattended. The only way to achieve total automation (or any portion of it) is through customization. AUTO-CODE provides all the tools needed for the NC/CNC code generation.

AUTO-CODE Environment

As said before, AUTO-CODE runs from within AutoCAD. The goal of AUTO-CODE is to transform graphic information in a drawing into an NC/CNC code file. The NC/CNC code file is a text file that can be viewed and changed in any word processor or text editor. The graphical information is in the form of an AutoCAD drawing which can be viewed and changed using AutoCAD. If the source graphics system is not AutoCAD, then the DXF or IGES file import options may be used to bring the external drawing data into AutoCAD.

Since not all of the engineering drawing information is relevant to the machining operation of a specific process, AUTO-CODE is interactive and relies on the operator to help select the proper geometry and define the proper sequence of events. This way, information like the title block and details can be left on the drawing for reference while the NC/CNC program is created.



The primary user interface is dialog box based and does not significantly alter any of the AutoCAD menus or other programming systems. The AUTO-CODE software was written using the Visual LISP language augmented by ObjectARX, which provides very tight integration with AutoCAD.

AUTO-CODE and AutoCAD both run inside Microsoft Windows (32-bit version). The Windows environment provides the tools needed to manipulate the files that are generated as a result of running AUTO-CODE.

After AUTO-CODE creates the text file containing the NC/CNC program, the next step is to get the program to the machine tool. In most cases it involves sending the NC/CNC program in the text file to the machine tool using some sort of communications utility. The options range from solutions such as using a local area network (LAN) to direct numeric control links (DNC) to paper tape punching to manually typing the program into the machine controller.

AUTO-CODE Steps

The first step in creating an NC/CNC program using AutoCAD and AUTO-CODE involves creating a picture. In general, the AutoCAD graphics are prepared using the normal methods available. There are some concepts to keep in mind, such as making sure the end points match up and the drawing is properly scaled.

When preparing for NC/CNC coding with AUTO-CODE, the continuous path areas should be joined into polyline objects. This is an important step to remember since polylines are the fastest way to set up a program using AUTO-CODE. Centerline offsets and other geometry needed for the machine tool path can then be added as well. The Geometry Tools provided with AUTO-CODE may also be of help in preparing your drawing before defining the NC/CNC program.

At this point, the drawing is ready to be used as a programming tool for the creation of NC/CNC code.

Staying inside AutoCAD, load the AUTO-CODE software and select the machine tool interface you want to use.

Next pick the first tool to use (for multiple tool machines) and you are all set to define the tool path.

Each machine tool interface varies in what it is expecting in the drawing. Virtually all of the AUTO-CODE interfaces use polyline objects in one fashion or another. For three-axis machine tool programming, drawing information can be presented in either 2D or 3D. Keep in mind that the third axis information will have to be supplied at some time during the programming process. If the part involves many elevation changes, then it is best to draw in 3D. Parts that have minimal changes in elevation can be drawn in 2D with the depth provided during program path definition.

AUTO-CODE Operations Overview

Most of AUTO-CODE is accessed through a series of dialog boxes that run inside AutoCAD. Selecting the NC Generator option from the AUTO-CODE 4.2 pull down menu starts the program. You can exit the AUTO-CODE interface at any time and restart it in the same drawing session by typing the command NCG (Numeric Control Generator) or by selecting NC Generator from the pull down menu. If you reload the drawing in the editor, reload the application and it will pick up where you left off the last time the drawing was saved. The dialog box structure is explained in general terms in the following paragraphs. Note that if you have a custom interface the dialog structure described may not exactly match what you will be using.

NCG Command – This command is stored in the AUTO-CODE partial menu that must be loaded. NCG can only be used after AUTO-CODE has been started and then cancelled in a drawing session. To start AUTO-CODE, select the NC Generator option from the pull down menu inserted during installation of the software. If the pull down menu is missing, use the AutoCAD MENULOAD command to restore it.

Select & Load Interface – The first time AUTO-CODE is loaded into a drawing, the interface to use is requested. The interface is the post processor or target machine tool. Some custom interfaces may ask for additional information at this time such as project numbers and other NC/CNC program header information.

Main Dialog – The main dialog box of AUTO-CODE is where you decide what you want to do next. The current NC/CNC program coordinates are displayed as is the state of the tool (in cutting mode or not). Specific information about the buttons and options in the main dialog box is provided later in this manual.

Settings Dialog – Tool settings, feed rates, spindle speeds and other control information is established in the settings dialog box. Custom machine tool interfaces may use this dialog box to provide additional (or reduced) input as needed. Most of the time, you should check the settings dialog box values when starting AUTO-CODE to establish that they are correct for the application at hand.

AUTO-CODE Route, and Drill Dialogs – These buttons lead to the route or drill dialog boxes. The route dialog box is where you set up the system to define a series of tool movements. These movements are along entities such as contour cutting a polyline or line/arc path sequence. The drill dialog box is where you set up the system to define a series of drilling operations. Objects

such as circles and block inserts can be located automatically and processed. The path and drill dialogs will be used when you have set up the tool and are ready to define cutting sequences.

Icon Functions – Various machine operations are accessed through the icon functions selected in the main dialog box. These functions will vary from one machine type to another. See the specific reference sections for the individual machine interfaces for more details.

Edit Dialog – Should you make a mistake in the tool path, you can correct it with the edit dialog options. New movements can be inserted into existing programs, points moved, and points erased with the edit dialog options. Of course, you can also use the AutoCAD edit commands to change geometry as well.

Generate NC Code – Once the program path is defined, the next step is to generate the NC program file. You will be given the chance to define the name of the NC output file and then the system will return you to AutoCAD after creating the file. Generally this option is only selected once in a programming session.

Edit CFG – Control over how the NC/CNC code will look at the sub-block level is found in this dialog box series. Data such as decimal point location, precision of output, coordinate addressing, and so forth can be edited. Details about editing the translation system can be found in the CUSTOM.HLP file found in the AUTO-CODE program directory.

Adding Graphics after AUTO-CODE has been started

You can always leave the AUTO-CODE interface and return to the AutoCAD command line. When doing this, the current state of AUTO-CODE is preserved for the next time you start the interface in the same drawing. To restart the interface, type NCG at the command line.

Any graphics can be added to the drawing after AUTO-CODE has started operations. These additions will not interfere with the operations of the software as long as you avoid using the AUTO-CODE layers and standard block names. Note that all AUTO-CODE layers, blocks, and AutoLISP symbols start with the letters “NCG” making it easy to avoid using them.

There are times when you will want to remove the graphics that AUTO-CODE has added. For example, you may want to run the tool over a contour twice. If this is the case, remove the lines and arcs from the NCG* layers and you will see the original drawing with only the control points shown. The control points are the actual “smarts” behind AUTO-CODE and beginners should not manipulate them using conventional AutoCAD commands. Once you have learned the AUTO-CODE system and understand how control points work inside the system, you can edit them directly.

The AutoCAD STRETCH command can be used to move control points in a region and thereby reprogram a shape with different sizes. After stretching the objects, restart AUTO-CODE by typing NCG and then select the Generate NC button to create a new program for this redesigned part.

AutoCAD Knowledge

The polyline object is used to define contiguous tool paths for contour cutting. Circles and block inserts are used to define the location of drill operations. Thus a properly prepared drawing can be processed by AUTO-CODE efficiently; however, a poorly prepared drawing may be worthless in terms of CAD/CAM integration.

The following concepts and methods should be understood in AutoCAD to achieve the best possible results. The specific commands are well documented in the AutoCAD reference materials and are summarized here with a focus on using them for the purpose of preparing a drawing for NC/CNC programming.

The polyline object is used extensively inside AUTO-CODE. It is used to define contour perimeters and roughing boundaries. In AutoCAD, the command "PLINE" will start the polyline drawing process. A polyline is a continuous series of lines and arcs with connected end points. When creating a polyline you define the sequence as a series of points with connecting segments made up of a line or an arc.

Because you must define the points in sequence, most CAD operators find the polyline difficult to create initially and prefer to work with simpler entities such as lines, circles, and arcs. When this is the case, the polyline edit command "PEDIT" is used to join the independent objects into a single object. This operation also verifies that the end points match up properly. If they do not match up, the PEDIT command will not join the objects into one.

PEDIT asks that you initially select a polyline. When just starting to convert the graphics information in a drawing into polylines, you can select a line or an arc and AutoCAD will convert it into a polyline object. Once in the PEDIT command, select the Join option and then select the objects that join up with the initial object selected. You do not have to select these objects in sequence and can even use the window selection options (or type in ALL to try and join anything in the drawing to the selected object!). When the Join operation is complete, it will inform you as to how many objects (entities) were joined to the form the polyline.

Hint - When joining objects that represent continuous cutting paths make sure they are closed through the PEDIT command. When the polyline object is closed, the PEDIT command will present an option to open the object in its command option set. When the object is open, the option to close it will be present instead.

PEDIT allows you to do many different things with polylines ranging from inserting points to cutting out segments to moving individual vertices. It is strongly advised that time be spent learning how to use this command. Think of the polylines as the tooling path and you will quickly see why this concept is quite powerful for designers and NC parts programmers alike.

Sometimes the drawing has errors in the end points. When end points do not match up, the PEDIT command will fail to connect the objects. You can repair the end points using the AutoCAD "FILLET" command. By setting the radius to zero, the line segments selected will be corrected so that they intersect at a point. Other options include using the GRIPS system to move the end points of both lines and arcs.

The AutoCAD Express Tool package contains a polyline join command macro that will find points that nearly miss and correct them. Ask your AutoCAD supplier about this useful tool.

Drawings are not always created in the same orientation in relation to how they will be machined. You can use the "ROTATE" and "MIRROR" AutoCAD commands to position the part drawing in a template drawing of the machine work area. The template drawing is optional but recommended when you are just learning how to program the system. You make your own template drawing and include features such as the tool change or rest position and clamps or holder locations.

Mechanical Desktop® Integration

AutoCAD Mechanical Desktop is an extension for AutoCAD that brings surfaces and solids into the tool set of the designer. Mechanical Desktop not only provides extensive 3D manipulation tools but also provides a parameter engine for defining part families and assemblies. This product is rapidly gaining popularity as a design tool of choice for engineers who need to think in more than just orthographic (2D drawing) projections.

For the machine tool programmer, Mechanical Desktop supplies some very powerful commands that can be used to manipulate the 3D objects. Basically, the problem is to reduce the 3D object information into something that represents tooling paths. This is accomplished using the section and flow line features of Mechanical Desktop.

Section cuts are when you create a series of objects from a 3D object by cutting a section through the surface object. When defining a section cut, the cutting plane is defined as the current UCS (user coordinate system) which in most cases will be defined as parallel to the X-Y plane. The UCS can be set to any plane and should be located at the first cut you want to make. A series of section cuts can then be created along parallel planes by selecting the multiple cut option and specifying an offset between each cut. The cut sections created by this command are lightweight polyline objects that can be used by AUTO-CODE to program a tooling path.

Flow lines are polylines that flow across the surface of a 3D object. While section cuts are best applied to solid objects to determine inside areas, flow lines work best with outside surfaces. You can control the direction and density of the flow lines and when satisfied with the results, use them to define the tooling path.

These tools can be used in whatever manner you desire to achieve the result of building a series of polylines. Sometimes it will make more sense to do section cuts, while other times the flow lines will seem more proper. One thing to keep in mind is that the surface can be used directly if you are working with cutter compensation (in all three axes of control). Otherwise, you will want to offset the surface by the radius of the cutting tool as you generate the polylines. The offset is defined at the same time you are defining the section cuts or flow lines.

In order to create section cuts and flow lines, you must have the Mechanical Desktop software installed and running. If you do not have the Mechanical Desktop enhancement, the polylines will need to be created at the workstation that does have the software. Mechanical Desktop can be added to an existing AutoCAD system by contacting your nearest Mechanical Desktop authorized dealer and purchasing the software.

AUTO-CODE has an extra utility command called "SLICER" that repeats the section cut command and creates polylines from the results. This utility is accessed through the machine controls menu documented in the CUSTOM.HLP file. The SLICER utility will cut up most surfaces defined in Mechanical Desktop.

DFM – Drawing For Manufacturing

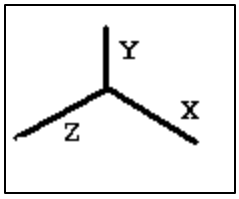
AutoCAD is a very powerful graphic editor and can be used to draw virtually anything you can imagine to machine. The more you learn about AutoCAD, the more productive you will be in using the system for CAD/CAM. This section discusses some common sense aspects of using AutoCAD for the purposes of CAD/CAM in general. These ideas apply to virtually any system that works with AutoCAD.

Drawings created in AutoCAD can be very accurate. In fact, the coordinate values found in AutoCAD drawings can have up to 15 digits and still be accurate. The problem is that AutoCAD drawings can be too accurate for the machine tools if you are not careful. It is important to create the graphics at not only a proper scale but also using base coordinates that are realistic. For example, you can draw a 2" x 1" rectangle that is very accurate. But if the corner is located at a point such as 1.23451923432, 3.234501248 the machine tool will not be able to match it exactly.

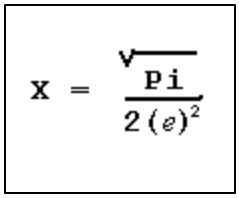
There are several additional points to keep in mind when creating drawings for manufacturing output. These rules apply to all CAM interfaces and not just AUTO-CODE.

Be aware of the zero point. The zero point of the drawing is most often the lower left corner of the display or drawing border. This is not always the case with the machine tool the part will be created on.

Draw at a one to one scale . Do not mix scales in the drawing of the part profile that you want to machine. For example, it is not all right to show a groove at one scale and the profile at another. If this is done, the CAM system will most likely follow the groove as drawn resulting in the wrong part being created. If a detail is within the machine tool tolerance, then it should be drawn at the same scale as the rest of the part. The scale selected should be a one to one relationship between the drawing units and output units needed by the machine tool for most CAM systems. AUTO-CODE does support scaled drawings; however, the scale of the part must remain consistent for the output to be proper.



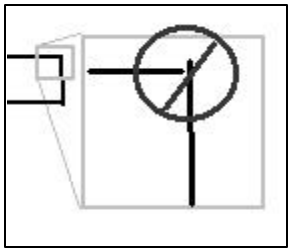
Orient the Part Drawing. Most CAM systems use the X-Y orientation of the AutoCAD drawing to control the X-Y orientation of the NC output. You should rotate and move the part drawing into place. A background picture is often helpful in this regard and many AUTO-CODE users create a primitive drawing of the work area and tool-changing zone (if any). The drawing of the part is then located inside of this area for machining.



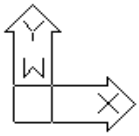
Using realistic point values cannot be stressed enough. The point values in AutoCAD are used directly in the CAM software and any precision problems may result in parts being made out of tolerance. If the scales and points are invalid then the resulting output from the CAM system will be invalid. A common term used in computers is GIGO (which stands for

Garbage In Garbage Out). This situation applies when drawings are created without any concern for the manufacturing output. Simply changing the dimension text in a drawing is not enough when changes are required. The graphics need to be kept accurate for CAD/CAM to work properly.

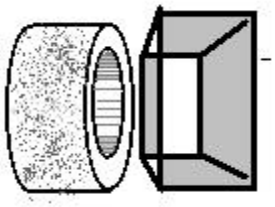
Use layers to aid in identifying contours and machine surfaces. This is not required for AUTO-CODE but is in other CAM systems. The idea is to use AutoCAD layers that help isolate the geometry of interest to the machining operations. In this theory, you would have a layer for the part, another for dimensional text, another for tables, and so forth. When programming, only the part layer needs to be enabled for display thereby removing excess graphical information from the display.



Intersection points should be accurate. Once again, accuracy is important. If two lines meet then they should actually meet by sharing common end points. The easiest way to check that situation is to convert connecting lines and arcs into polyline objects. Polyline objects are continuous objects that happen to be the best way to introduce information to the AUTO-CODE software, as well as to many other CAM systems that work in conjunction with AutoCAD.



Always start AUTO-CODE in the World Coordinate System You can draw in other coordinate systems, but return to the World Coordinate System in Model Space before running AUTO-CODE. AUTO-CODE will convert polylines drawn in alternate coordinate systems (as is often the case when integrating with the Mechanical Desktop software) to the world coordinate system for cutting purposes. Machine tools can cut arcs in one plane only (some can switch between three primary planes) and thus any arc movements should be drawn accordingly.



Draw parts that can be made. Using AutoCAD and Mechanical Desktop, it is not too difficult to design parts that cannot be made on the machine tools available. Just because you are using a CAD/CAM system does not mean you can now make things the machine tools could not make before. You may be able to better utilize the machine capabilities, but you cannot do the physically impossible. After all, you cannot put a square peg in a round hole.

The key to successfully using AutoCAD with any CAM system is to be consistent and accurate in the AutoCAD model. Drawings that have no realistic sizes or have dimensions that do not match the actual graphics are sometimes called “Electronic Cartoons”. This type of drawing is useless when it comes to integrated processes such as CAD/CAM and should be avoided. Integration of design and manufacturing requires a cooperative effort between the electronic drafting and manufacturing specialists.

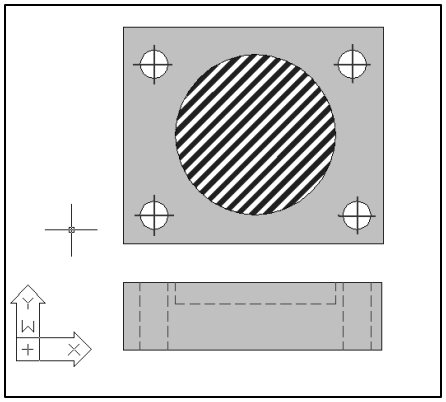
AutoCAD Drawing Methods for AUTO-CODE

There are several things you can do that improve the ability of AUTO-CODE to create NC/CNC code from your AutoCAD drawings. As mentioned above, drawing in polylines is the recommended procedure for defining continuous graphics. Below are some other methods that can be employed that will increase the throughput of the AUTO-CODE system. These are only suggestions and do not have to be followed exactly as described. Just as there are many ways to draw in AutoCAD, there are many ways to program in AUTO-CODE.

Contour Cutting Setup – Contour cutting includes mills, flame & plasma cutters, wire EDM machines, and lasers. When defining the graphics for contour cutting it is strongly recommended that polylines be used to represent both inside and outside cuts. AUTO-CODE will add leaders if needed to any closed polyline objects.

To create a polyline from AutoCAD graphics, use the PEDIT command inside AutoCAD. The easiest approach is to select a line or an arc segment and convert it to a polyline. Then select the Join option in the PEDIT command and when asked to select entities, enter ALL to select the entire drawing. PEDIT will then attempt to join objects to the initial line or arc selected. After this operation is completed, check to see if the Open or Close option appears in the PEDIT command options. If the term Open appears, then the polyline is closed and ready for contour cutting. If the Close option appears then the polyline is not closed. If the path described is what you want to cut, then it is ready for processing, otherwise select the Close option to join the polyline into a continuous object.

More information about PEDIT can be found in the AutoCAD reference manual. A closed polyline is not required. Open polylines can be processed by AUTO-CODE and are commonly used when you need absolute control over leader placement and geometry.



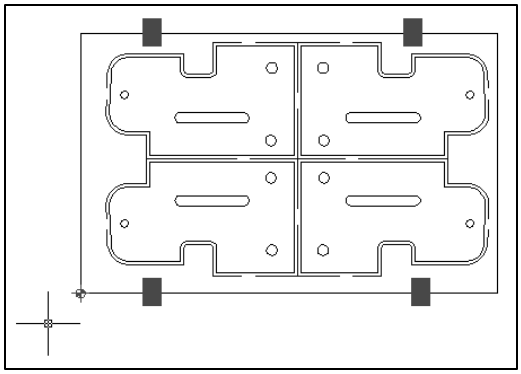
Drill Operations Setup – Drill operations apply specifically to the mill interface. The AutoCAD circle and block insert entity objects are the best way to present drill geometry to AUTO-CODE.

The AUTO-CODE system contains a utility for searching your drawing for circles and block inserts, and then routing the drill tool to each of them. The search utility for circles will look for either all circles in the drawing or for circles with a specific radius value.

When preparing a drawing in which you wish to spot drill some holes and tap just a few of them, you can draw the tapped holes at a slightly different radius. That way, when AUTO-CODE is run the search for all circles can be used to spot drill all of the holes and another search performed for a specific radius match for the tapped holes.

Changes to Drawings – In most cases a machine tool programmer cannot directly use the drawings as supplied by an engineering graphics group without some preparation work. The reason is that the graphics group has different objectives when putting a drawing together. They want to create a presentation drawing showing the part to be created. The drawing supplied is not an "in process" drawing.

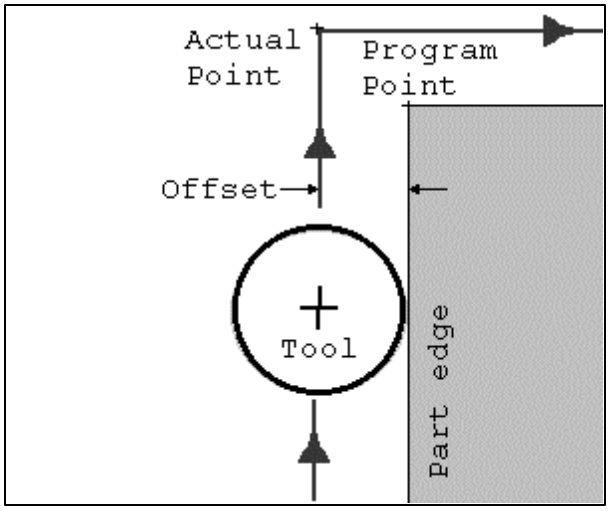
If the original drawing is accurate then it can be used in the CAM application after some edits have been applied. The edits generally involve the combination of lines and arcs into polylines, copying and rotating the parts into position for machining, and applying any offsets that may be desired.



Tabs and Break Off Points – When programming a contour cutter, it is not uncommon to leave "shake-break" points. These are small segments that are not cut but are used to hold another portion of the material rigid while cutting completes elsewhere. When all cutting is completed, these shake-break segments are then cut free or broken away from the stock.

To program tab breaks in the continuous geometry, use the BREAK command before running AUTO-CODE and defining the tool path. Apply BREAK to the polyline objects at the location where the segment is to remain uncut.

A tab break geometry utility is provided in the AUTO-CODE software system that enables the introduction of tab based leaders into a drawing while running AUTO-CODE. See the CUSTOM.HLP file for details about this utility and how to incorporate it into your machine controls list.



Cutter Compensation – AUTO-CODE supports cutter compensation as a programmatic item. That is, you can include it in your source codes at your discretion while generating code inside AutoCAD using AUTO-CODE.

An important thing to keep in mind is that if you decide not to use cutter compensation for edge programming, you will need to define the centerline graphics in the drawing. Centerline graphics are created quickly by using the OFFSET command with polyline objects. OFFSET is an AutoCAD drawing editor command.

Starting AUTO-CODE inside AutoCAD

See the chapter on installation for details about installing AUTO-CODE on your system. Installation of AUTO-CODE is done inside AutoCAD.

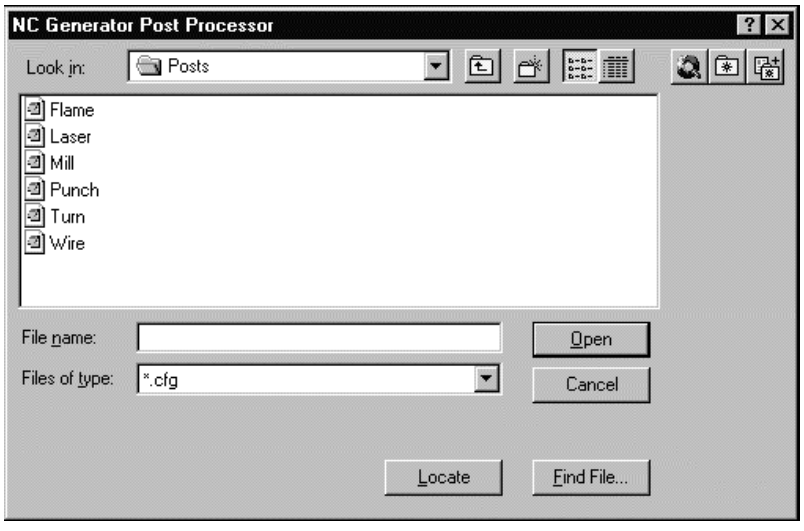
AUTO-CODE runs inside AutoCAD, meaning that you start it in the AutoCAD drawing editor. When first learning how to use AUTO-CODE it is highly recommended that you save any drawings in a file with a different name before loading the system so that you can try again if you go off in the wrong direction for whatever reason. Even experienced AUTO-CODE operators can benefit from this precautionary measure.

When AUTO-CODE was installed, a pull down menu labeled "AUTO-CODE 4.2" was placed near the right end of the pull down menus. *If you do not see the AUTO-CODE 4.2 pull down menu, see the SETUP chapter for manual installation of the custom menu.* Select the AUTO-CODE menu and then the "NC Generator" option to start the interface.

The "NC Generator" option can be selected when first initiating AUTO-CODE in a drawing or to restart the program after changing or adding some more graphics to the drawing. You can also restart the interface at the keyboard by typing the command NCG.

The "Clear program" option is provided to remove any AUTO-CODE data objects remaining in a drawing. This allows you to start over. It is strongly recommended that the drawing be reloaded when starting again to make sure that all variables have been reset to the initial values.

In the "Help" sub-menu you will find help about running AUTO-CODE and customizing the software. There is also an entry to un-install the software. When the un-install command is selected, the AUTO-CODE menu and files inside AutoCAD are removed. The AUTO-CODE program directory is not removed and should be removed using the Windows File Explorer program.



When first starting, AUTO-CODE will ask you to select a machine tool interface to run. Select the one that most closely resembles the type of machine you are using or one written specifically for your machine tool. Machine tool interfaces are all located in the same directory inside the AUTO-CODE program directory structure. Picking a CFG file with the name of the interface desired makes the selection. For example, you might select the MILL.CFG file for milling operations.

Common Questions

New users always have more questions than the documentation can hope to cover and often times it is terminology that stands in the way of getting the answers needed. The following are questions that many people ask when learning about CAD/CAM and the AUTO-CODE system. Most of these questions relate to the system and future maintenance of the interface files. Others are about specific machine operation issues.

Should you save your drawings when done in AUTO-CODE?
It is not really necessary to save your drawings when finished with the generation of an NC/CNC program. You can save the drawing to the disk system with the AUTO-CODE additions and then recall the same drawing at a later time, make edits, restart AUTO-CODE and then regenerate the program file. Most AUTO-CODE users do not save their drawings when finished with the coding. The primary purpose of the software is to convert AutoCAD drawings into working NC/CNC programs and once the program is created, there is virtually no need to save the modified AutoCAD graphics.

Can you program multiple, different parts inside the same drawing?
Yes you can. Before starting any of the programming, save the drawing to disk. For each part to be programmed load the drawing into AutoCAD and then load AUTO-CODE. This will result in different NC/CNC program files being created for each part in the drawing. You can also generate code for all the parts into a single program and then use a text editor (such as Notepad) to break the program modules up into smaller segments for each part.

Can you work with drawings created in Mechanical Desktop®?
Yes you can. Mechanical Desktop is an enhancement to AutoCAD that provides facilities for 3D solid and surface manipulations. Using features from Mechanical Desktop such as flow lines and cut sections, polylines can be created for use in AUTO-CODE. AUTO-CODE does not accept the 3D surface models directly but instead works with the wire framing information to generate tool paths. This gives you, the programmer, the ultimate in control over how a part is manufactured. 3D surface models are wonderful for creating visual checks and other computer graphic output; however, they are not the best input for machine tool programming as jigs and other fixtures are difficult to introduce into the model as “no cut” areas.

Can you work with drawings created in other CAD systems?
If the other CAD system can create an AutoCAD DWG or DXF file, then AutoCAD can be used to read this file. Once inside AutoCAD, it can be worked with AUTO-CODE to generate an NC/CNC program. The main issues are the conversion of line and arc objects into polylines and the accuracy of the imported data. Using the AutoCAD editor to manipulate the graphics into the proper condition can solve both of these issues.

Can you use AUTO-CODE on another workstation?
Yes you can. In fact, you do not even have to have it authorized. Instead, save the drawing after defining the tool path and then move the drawing to the system with the authorized AUTO-CODE software. Load the drawing and press the Generate NC button to create the program. The machine tool interface file set should be identical on both systems otherwise an improper translation may result. This technique is used in schools to allow the teacher to require a final check of the NC/CNC program before it is downloaded to the machine tool.

What happens when the AutoCAD platform is upgraded?
The AUTO-CODE authorization system is tied into the AutoCAD serial number and upgrading AutoCAD will most likely result in a new serial number. This will result in your needing a new authorization code for the AUTO-CODE software before you can run it with the updated AutoCAD. Generally, the incremental updates from Autodesk do not change the serial number but in some cases they do. In those cases, AUTO-CODE does not need upgraded and a new authorization code will be issued at no charge if you can present the old and new serial numbers of AutoCAD when ordering the authorization code. For new releases of AutoCAD it will be required that a new version of AUTO-CODE be created. In these cases, new version releases are treated as additional licenses of the software and sold at a discounted price off the cost of the first license. An update diskette (or CD) will be supplied with the newest software release along with any updates to the documentation (on CD).

Can I make my own post processors?
Yes you can. But first, let's get some terminology under control. The AUTO-CODE system is an interactive programming solution and as such it does not use post processors. A post processor reads a standard format output file (such as from the APT language) and converts that code into the specific NC/CNC command sequences needed to run a target machine tool. Post processors are somewhat rigid in that they only accept the standard input format. AUTO-CODE uses a translation system, which is roughly like a post processor, but the translator is dynamic and changes as it translates the codes. Another difference is that the primary programming tool is user driven using dialog boxes, custom AutoLISP programs, and patterns. Hence the term interface is better than post processor when describing the software files needed to have AUTO-CODE assist in the programming of the NC/CNC machine tool.

What happens to my custom interfaces in the future?
We are making every effort to keep custom interface systems standard into future releases of AUTO-CODE. If anything, we will be adding new features and, when file formats are revised, a utility will be provided to convert older formats to the newer ones.

Can AUTO-CODE be made fully automatic?
Yes it can, but you must supply AutoLISP program modules that drive the system. There are many AutoLISP latches provided in the system where your programs are called as events occur. By tying into these latches you can write programs that perform varying degrees of automation, ranging from tool change setups to taking over as soon as the custom interface is loaded. AUTO-CODE automation can be achieved in most instances where well controlled drawing standards are in place and the integrity of the geometry is trusted to a high degree.

How often is AUTO-CODE upgraded?
We place new updates for AUTO-CODE on our web site at www.autocode.com several times a year. New updates occur when new features have been added to the system. You can download the updated version for your copy of AutoCAD at any time and simply replace the program and associated data files. The download version will tell you which version of AutoCAD is required in case you have fallen behind in updating that piece of software. Be aware of the serial number shift if you are also updating your AutoCAD.

Just what is AutoLISP and Visual LISP?
AutoLISP is a programming language that runs inside AutoCAD. Using a peculiar syntax, AutoLISP is exceptionally strong at manipulating point data and collections of points and data. The language provides direct access into the AutoCAD graphic database and user interface options for obtaining input and presenting information. AutoLISP can be learned by reading

books on the subject, attending classes, or by simply reviewing as many examples as possible. Visual LISP is the new name for AutoLISP in AutoCAD 2000. Visual LISP is the same as AutoLISP but with many extensions. Programs written in AutoLISP will run fine in Visual LISP.

Can AUTO-CODE be customized using VBA?

VBA, Visual BASIC for Applications, is a powerful programming language that also runs inside AutoCAD. It uses a Windows tool known as ACTIVEX to communicate with AutoCAD. AUTO-CODE does not support ACTIVEX automation and thus you cannot use VBA to customize the interface. You can use VBA to initiate AUTO-CODE and run many of the Visual LISP functions. However, due to the nature of the interface and the desire to provide you with the best solution for customization we have selected to stay with Visual LISP as the primary programming system for total automation.

What is a pattern?

Patterns are used in AUTO-CODE as a way of detailing the block sequence of NC/CNC output for specific circumstances. There are patterns for tool changing, starting the program, ending the program, cutter compensation, and any other event you need to program. Patterns are detailed in the CUSTOM.HLP file found in the AUTO-CODE program directory.

What is a Control Point?

Control points are the basic data objects of the AUTO-CODE system. They are stored in drawings to define the tool path and tooling setup information. In AutoCAD operator terms, control points are block inserts with a pair of attributes. Normally these block inserts only appear as dots (point objects) in the drawing. The PDMODE system variable can be set to a different value if you want to see the control points more clearly. Control points are used in patterns to build complete blocks of NC/CNC program code. See the CUSTOM.HLP file for more details regarding the contents of control points and what the attribute values can be.