ShopTools Development Notes

All content Copyright (c) 2025 - Daniel Patterson, MCSD (danielanywhere)

# General Shop Procedures

The procedures in this section describe tool-agnostic activities that should be supported as directly and intuitively as possible.

## Operations From a Starting Point

Most cuts that can be made from a starting point consider a distance from an edge or a corner of the material. In those cases, the distance or axial tool offsets are expressed, as appropriate, and an offset from an origin specified by the user.

If a cut is going to begin on the edge of the material, only one distance is needed, along with the relative origin to which that distance is relative. However, when the cut needs to begin anywhere within the material, away from the edge, both X and Y offsets need to be provided along with associated offset origins for each.

The following procedures detail the specific information needed for each type of cut when a starting point is used.

### Drill a Hole

Typically, when you are prepared to drill a hole in a piece of material, you will have the following pieces of information in mind.

* **OffsetX**. The distance along the X axis into the material body.
* **OffsetY**. The distance along the Y axis into the material body.
* **OffsetXOrigin**. The reference edge or corner from which to measure the X starting point.
* **OffsetYOrigin**. The reference edge or corner from which to measure the Y starting point.
* **Depth**. The depth to drill from the surface. If blank, the depth of the material is used.
* **Tool**. The tool determines the diameter of the hole.

### Cut a Line - Scenario 1 - Angle and Length

To cut a line of a specified length and angle from a starting point, use the following information.

* **OffsetX**. The distance along the X axis into the material body.
* **OffsetXOrigin**. The origin of the offset for the starting X position. If OffsetX is 0 and OffsetXOrigin is Left or Right, then the cut begins at the left or right sides, respectively.
* **OffsetY**. The distance along the Y axis into the material body.
* **OffsetYOrigin**. The origin of the offset for the starting Y position. If OffsetY is 0 and OffsetYOrigin is Top or Bottom, then the cut begins at the top or bottom sides, respectively.
* **Length**. The length of the cut to make. If omitted, the full length of the material is used.
* **Angle**. The angle at which to travel, relative to the selected edge.
* **Kerf**. The side of travel upon which to orient the kerf.
* **Depth**. The depth to remove from the surface. If blank, the depth of the material is used.
* **Tool**. When used as a cutting implement, the router head will be positioned one radius into the Kerf side from the edge being cut. Information about cutting speed and number of passes required for depth of cut are also determined from the selected tool.

### Cut a Line - Scenario 2 - Point to Point

In some cases, a line from Point A to a specific Point B is needed. Rather than calculating the angle and length of that cut, it can be more convenient to express the ending point. The following values are used to accomplish that task.

* **StartOffsetX**. The distance along the X axis into the material body at which to start.
* **StartOffsetXOrigin**. The origin of the offset for the starting X position. If StartOffsetX is 0 and StartOffsetXOrigin is Left or Right, then the cut begins at the left or right sides, respectively.
* **StartOffsetY**. The distance along the Y axis into the material body.
* **StartOffsetYOrigin**. The origin of the offset for the starting Y position. If StartOffsetY is 0 and StartOffsetYOrigin is Top or Bottom, then the cut begins at the top or bottom sides, respectively.
* **EndOffsetX**. The distance along the X axis into the material body at which to stop.
* **EndOffsetXOrigin**. The origin of the offset for the ending X position. If EndOffsetX is 0 and EndOffsetXOrigin is Left or Right, then the cut ends at the left or right sides, respectively.
* **EndOffsetY**. The distance along the Y axis into the material body at which to stop.
* **EndOffsetYOrigin**. The origin of the offset for the ending Y position. If EndOffsetY is 0 and EndOffsetYOrigin is Top or Bottom, then the cut ends at the top or bottom sides, respectively.
* **Relative**. A boolean value indicating whether the ending offsets are relative to the starting offsets. If true, EndOffsetXOrigin and EndOffsetYOrigin are ignored if specified.
* **Kerf**. The side of travel upon which to orient the kerf.
* **Depth**. The depth to remove from the surface. If blank, the depth of the material is used.
* **Tool**. When used as a cutting implement, the router head will be positioned one radius into the Kerf side from the edge being cut. Information about cutting speed and number of passes required for depth of cut are also determined from the selected tool.

## Operations From the Current Position

In cutting lines and shapes from the current position, it is important to keep track of the kerf offset of the bit from the target location on the material, especially while making corners and other polylines, which consist of making multiple subsequent cuts. To accommodate this challenge, an internal virtual position is maintained at the exact location of the physical line being cut. When a new shape is started from the current position, which is also the last-known ending position, the router can make a slight jog around the corner whenever applicable to continue plotting the path with the kerf on the appropriate side, relative to travel direction.

When making cuts from the last-known position of the router, less information is needed than when using a starting point, but some of the basics still apply, as described in the following paragraphs.

### Drill a Hole

Since the router is already positioned at the hole, the only information needed is the depth of the plunge. The tool is specified, but only to help determine the plunging speed.

* **Depth**. The depth to drill from the surface. If blank, the depth of the material is used.
* **Tool**. The selected tool contains information that helps determine the speed of plunge.

### Cut a Line - Scenario 1 - Angle and Length

To cut a line when the router has already been positioned, either by the user or by a previous cut, use the following information.

* **Length**. The length of the cut to make. If omitted, the full remaining length of the material is used.
* **Angle**. The angle at which to travel, relative to the selected edge.
* **Kerf**. The side of travel upon which to orient the kerf. If the router has been positioned by the user, select None to plot a straight line from the exact position.
* **Depth**. The depth to remove from the surface. If blank, the depth of the material is used.
* **Tool**. When used as a cutting implement, the router head will be positioned one radius into the Kerf side from the edge being cut. Information about cutting speed and number of passes required for depth of cut are also determined from the selected tool.

### Cut a Line - Scenario 2 - To Next Point

To cut a line from the current router location, use the following information.

* **EndOffsetX**. The distance along the X axis into the material body at which to stop.
* **EndOffsetXOrigin**. The origin of the offset for the ending X position. If EndOffsetX is 0 and EndOffsetXOrigin is Left or Right, then the cut ends at the left or right sides, respectively.
* **EndOffsetY**. The distance along the Y axis into the material body at which to stop.
* **EndOffsetYOrigin**. The origin of the offset for the ending Y position. If EndOffsetY is 0 and EndOffsetYOrigin is Top or Bottom, then the cut ends at the top or bottom sides, respectively.
* **Relative**. A boolean value indicating whether the ending offsets are relative to the starting offsets. If true, EndOffsetXOrigin and EndOffsetYOrigin are ignored if specified.
* **Kerf**. The side of travel upon which to orient the kerf.
* **Depth**. The depth to remove from the surface. If blank, the depth of the material is used.
* **Tool**. When used as a cutting implement, the router head will be positioned one radius into the Kerf side from the edge being cut. Information about cutting speed and number of passes required for depth of cut are also determined from the selected tool.

### Move the Bit

Moving the bit without cutting doesn't have a parallel operation in the Operations From a Starting Point section because any number of non-plunging moves can be supplied if necessary although only one is typically used at a time.

In this version, moving the bit has the same considerations as cutting a line from the current position, but without a depth. Notice that since the kerf is still an important consideration to keeping the finished dimensions of the workpiece in order, the tool is still supplied while moving the bit, to support starting the next line or curve correctly when expressing non-plunging portions of a path. If you want the center of the bit to follow the physical line while moving the bit, set Kerf to None and the Tool property will be ignored.

#### Scenario 1 - Angle and Length

To move by an angle and length, use the following information.

* **Length**. The length of the move to make. If omitted, the full remaining length of the material is used.
* **Angle**. The angle at which to travel, relative to the selected edge.
* **Kerf**. The side of travel upon which to orient the kerf. If the router has been positioned by the user, select None to move a straight line from the exact position. This setting has no effect if a tool is not selected.
* **Tool**. When a kerf is specified, the router head will be positioned one radius into the Kerf side from the line being spanned.

#### Scenario 2 - To Next Point

To move to another point, the following information is needed.

* **EndOffsetX**. The distance along the X axis into the material body at which to stop.
* **EndOffsetXOrigin**. The origin of the offset for the ending X position. If EndOffsetX is 0 and EndOffsetXOrigin is Left or Right, then the movement ends at the left or right sides, respectively.
* **EndOffsetY**. The distance along the Y axis into the material body at which to stop.
* **EndOffsetYOrigin**. The origin of the offset for the ending Y position. If EndOffsetY is 0 and EndOffsetYOrigin is Top or Bottom, then the movement ends at the top or bottom sides, respectively.
* **Relative**. A boolean value indicating whether the ending offsets are relative to the starting offsets. If true, EndOffsetXOrigin and EndOffsetYOrigin are ignored if specified.
* **Kerf**. The side of travel upon which to orient the kerf. Specify None to travel a straight line directly to the ending offset. This setting has no effect if a tool is not selected.
* **Tool**. When a kerf is specified, the router head will be positioned one radius into the Kerf side from the line being spanned.

## Procedural Considerations

The following are some of the general characteristics that can be recognized from the specifications of the above tasks.

* When OffsetX and OffsetY naming styles are used, they apply to scenarios where only a single point is needed to complete the entire operation, such as in drilling a hole or cutting a line by Angle and Length.
* When a line is activated by Angle and Length and no angle is given, the cut is assumed to be a straight cut inward perpendicular to the nearest edge, especially if one of the offset origins is aligned directly to one of the edges.
* When a line is activated by Angle and Length and no length is given, the cut is assumed to span the remaining amount of material in the established direction of travel.
* Wherever StartOffsetX, StartOffsetY and EndOffsetX, EndOffsetY naming styles are used, point-to-point paths are indicated, where there is at least an explicit ending point.
* Whenever StartOffsetX and StartOffsetY are omitted in a point-to-point path, the last-known router position is used.
* OffsetXOrigin, OffsetYOrigin, StartOffsetXOrigin, StartOffsetYOrigin, EndOffsetXOrigin, and EndOffsetYOrigin are each ignored if the directly associated measurement value was either not needed or not provided in that context.
* In point-to-point paths, the Relative variable is available to indicate that the ending offset is relative to the starting point.
* The position of the physical line opposite the kerf is tracked internally to allow for corner-rounding adjustments between strokes on a polyline.
* During a line cutting operation, the Depth is specified to limit the depth of cut or not specified to imply that the entire depth of the material will be cut.
* The major difference between a move operation and a line cut operation is that a movement doesn't implement a Depth property.

# Definition and Use of Custom Defined Properties

In the first draft of this application, an extensive system of custom defined user properties was created with the intention of allowing different kinds of operations and different kinds of tools than those already supported to be defined and loaded and runtime. Although there is a solid case for allowing the definition of custom patterns to be loaded in a configuration file, all of the basic properties for both tools and patterns are already known. As a result, the idea of custom property definitions will be completely dropped going forward.

# Configuration Data

The current schema of the configuration data is structured according to the following outline.

* PatternTemplates. PatternTemplateCollection. Collection of patterns defined for a session.
* Properties. PropertyCollection. Names and values of application-specific properties that configure the current session. The following are the currently defined properties in this list.
  + AxisXOpenEnded. Boolean. A value indicating whether access along the X axis is open-ended.
  + AxisYOpenEnded. Boolean. A value indicating whether access along the Y axis is open-ended.
  + DisplayUnits. DisplayUnitEnum. The preferred unit of display for the session. System units are always in millimeters.
  + Depth. Measurement. The depth (Z) dimension of the table, in resolved units.
  + GeneralCuttingTool. String. The name of the cutting tool to be used during general cuts where no specific tool is specified.
  + TravelX. DirectionLeftRightEnum. The positive X travel direction. Choices are Left and Right. Default is Right.
  + TravelY. DirectionUpDownEnum. The positive Y travel direction. Choices are Up and Down. Default is Up.
  + TravelZ. DirectionUpDownEnum. The positive Z travel direction. Choices are Up and Down. Default is Up.
  + UserDepth. Measurement. The user-supplied depth dimension of the table.
  + UserXDimension. Measurement. The user-supplied X dimension of the table.
  + UserYDimension. Measurement. The user-supplied Y dimension of the table.
  + XDimension. Measurement. The X dimension of the table, in resolved units.
  + XYOrigin. OriginLocationEnum. The XY origin location of the table.
  + YDimension. Measurement. The Y dimension of the table, in resolved units.
  + ZOrigin. OriginLocationEnum. The Z origin location of the table.
* UserTools. UserToolCollection. Collection of user tools defined for use in a pattern during a session.

# Handling Multiple Operations In a Single Template

When cutting a pattern that contains multiple operations, it becomes necessary to multiplex the common per-operation variables that would be repeated, like "OffsetX" and others.

To address this possibility, the variable **OperationName** is introduced for optional use on each operation of a pattern, if necessary. Here are some of the characteristics of the presence of operation name in the Operations list of any pattern.

* When operation names have been specified in operation entries, the name of the corresponding user variable to be filled in for that operation is prefixed with that operation name. For example, for an operation with the name "Cut 2", the corresponding "Offset X" user variable becomes "Cut 2 Offset X".
* The sample operation name can be used on multiple operation entries in the list. The corresponding user variables for each operation name will be filled on all matching operations.
* When no operation name has been specified for one or more operations, a set of non-prefixed user variables is added to the list of prompts, and those unnamed operations all receive the user value supplied for the non-prefixed variable.

# Coordinate Spaces

For the sake of maintaining the user's ability to read and write coordinates with direct regard to the specific physical origin and coordinates of the target table, ShopTools maintains multiple coordinate spaces; one that allows for direct expression of physical activities, and the other that allows for the visual rendering of objects and activities on the display.

To make this inter-dimensional coordinate handling as consistent as possible, both with the user and with real-world activities, all coordinates and dimensions are saved in real system units, as they pertain to the size of the physical table, but both the workspace workpiece areas are offset from the top left coordinate of 0, 0 while in the user display. Whenever an absolute coordinate is expressed, it is converted to its associated display space value.

For operations having typical relative offsets, a travel direction translation is commonly applied to guide the X and Y cursor in the expected directions for the table. For alignment and justification relative directional references like left, right, up, down, etc., all calculations are performed within the established display space.

The workpiece is maintained in the same units and alignment as the global workspace. All cuts placed on the workpiece are stored in a manner that correspond to the top-left oriented rectangle representing the workpiece area.

As noted, any truly absolute coordinates are saved in their notion and can be verified to have their stated specific location within the physical domain but are translated and rendered within the top, left-oriented workpiece space.

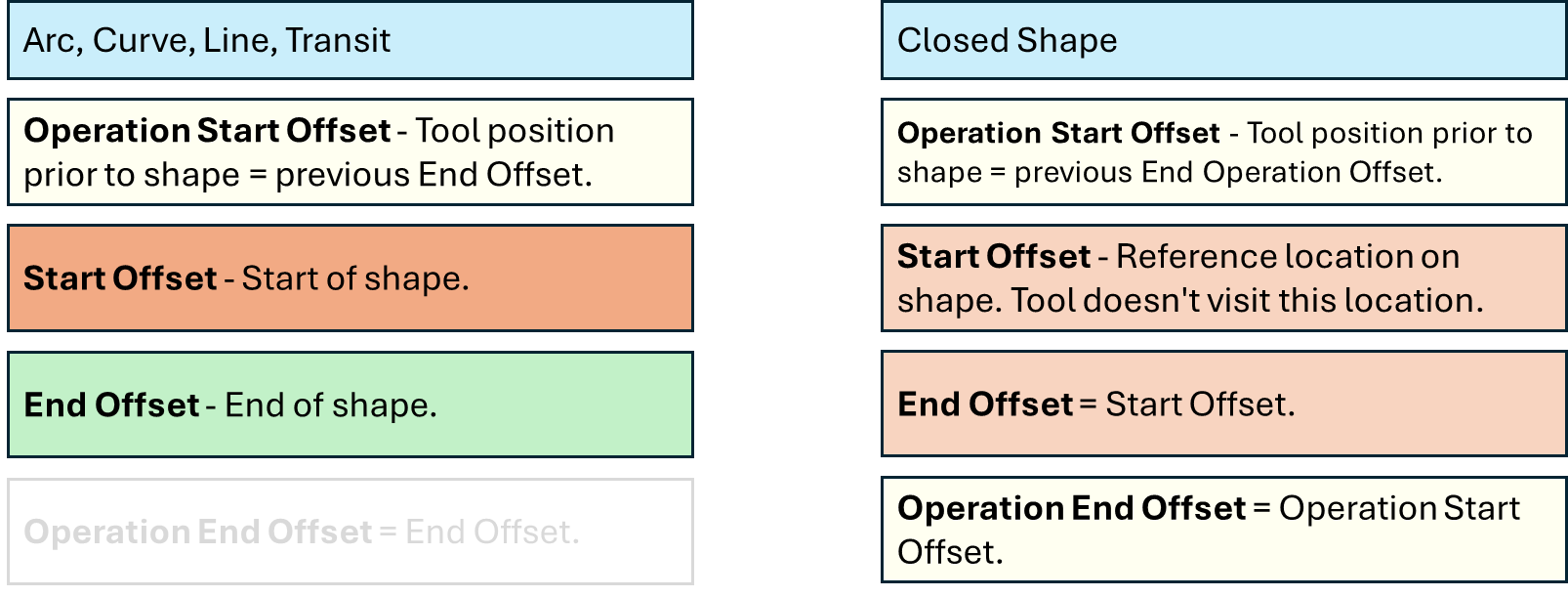
When the cuts are rendered for export to G-code, all of the coordinates of the work are re-rendered for placement in the physical space.

# Tracking Tool Location

With basic straight movements and lines, as well as open-ended paths like arcs and curves, the task of keeping track of a transiting movement to the next cut is trivial. For every operation, we know the starting and ending offsets, and those values are incrementally updated for an entire series of operations to render the cutting and non-cutting paths, both in the drawing window and the g-code rendering of the same series.

However, when it comes to closed shapes, particularly center-oriented shapes whose reference points don't touch the active path at all, additional information is needed to track the starting and ending locations of the tool separately from the starting and ending locations of the shape, the latter two of which are always equal.

The following table illustrates the general differences in how various values are handled depending upon whether they are path-based open shapes or closed shapes. Note that the term *Operation Offset* refers to the position of the tool, while *Start Offset* and *End Offset* both refer to the position of the shape.



Ellipses and circles are represented in the drawing start and end offsets by their top left and bottom right corner coordinates, in drawing space. On these shapes, the starting and ending tool locations are both equal to the closest point from the last-known tool location to the edge of the current shape.

When representing arcs, the drawing beginning and ending offsets represent the corners of an ellipse and the StartAngle and EndAngle properties are also used.

## Enclosed Shape Handling - Initial Position

When positioning for enclosed shapes, the tool is moved to the starting point on the shape, starts cutting there and ends cutting on the same coordinate.

## Enclosed Shape Handling - Visual Representation

When drawing an enclosed shape on the display, the drawing coordinates don't necessarily have any relation to the tool coordinates, unless the tool happens to be aligned with the upper left corner of the shape when drawing starts.

Otherwise, most closed shapes are drawn with reference to the upper left corner in drawing space.

## Enclosed Shape Handling - Physical Plotting

When physically plotting an enclosed shape, individual line segments are created that provide closest-point support for starting and ending that shape.

From the drawing space perspective, all plotting of enclosed shapes is done in a counterclockwise direction and the design line is left intact, which means that the outer edge of the shape follows a left kerf while the inner hollowing lines need no kerf.

Incidentally, this also means that the Kerf property is unneeded in filled shapes.

# Filling an Ellipse

When filling enclosed shapes in this version, the outer perimeter is drawn first using the equivalent of a Draw*{ShapeName}* command prior to completing the shape with the specified Fill*{ShapeName}* command. This approach allows a working area to be cleared, the interior of which can be cleared evenly.

The outlines of all of the enclosed shapes are somewhat trivial to produce. However, when it comes to the clearing process inside the outline, non-orthogonal shapes like the ellipse can be challenging because the head starts mid-way across the band on the first cut, stops at a different locations on each clearing pass, an finally also ends mid-way across the band on the last cut.

The following steps are used to reduce the perceived complexity of this cut.

* A bounding box of the full ellipse is taken.
* Uniformly reduce the reference ellipse by one tool radius to account for the outline already drawn.
* Create a horizontal line one tool radius down from the top of the ellipse and intersect the line with the ellipse. The intersections will be the ends of the first cut row in the band.
* At the end of this row, move the line down by one tool radius.
* Move the tip diagonally to the start of the next row.
* Move the tip across the current row.

# Working With Patterns

This application allows predefined cuts and operations to be defined within the PatternTemplates property of the loaded configuration.

## Operations

A pattern can contain multiple operations, stored in the Operations collection.

### Hidden Variables

If an operation doesn't need to interact with various of its native properties, the names of those properties can be included in the HiddenVariables string collection.

## Shared Variables

If a variable name is to be shared for all operations on the pattern, its name, as found in the PropertyName property of the OperationActionProperties.json list, can be added to the SharedProperties list. This has the effect of not prefixing that property with its operation's OperationName value and listing it only once in the user properties for the cut.

# Usage Notes

When entering values, the user is allowed to enter whole numbers, with or without measurement units defined in the distance category, decimal values with a leading non-decimal digit, spaces and fractions.