

BCiTool_PM – PatternMaker Macro for TurboCAD

Creating your own brush patterns for TurboCAD is a two-step process.

- Firstly you have to create the pattern and export it to a pattern file – you can use the BCiTool_PM macro for this purpose.
- Then you have to import the pattern file into TC's settings, or into a drawing where you can retrieve it easily.
 - With TC-X you can use TC Explorer or XMLINI to import the **xml** file generated by BCiTool_PM
 - For V9 and before you have to manually edit TC's ini file, and paste in the **txt** file produced by BCiTool_PM

This manual is mainly concerned with the creation part, though makes brief mention of the latter as appropriate. Consult other help files for more details on the importing process (eg the BCiTool_XI help file).

Compatibility

The macro works with TC7 Pro, TC8 Pro, TC9 Pro, TC-X Pro

Installing and running

The install procedure is roughly the same as for other macros in TC

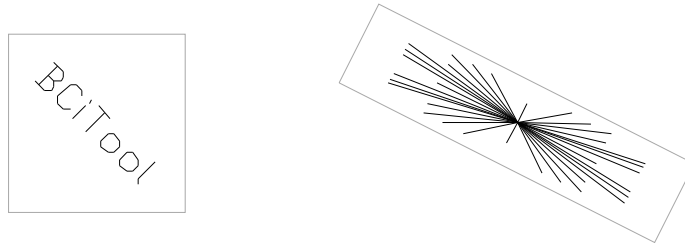
- Unzip the macro into the ...\\TC??\\Macros folder
- In TC open or start a new drawing
- Tools > VBA Macro > Macros ...
 - In the dialog
 - Type a new name in the 'Project Scope' edit box
 - Add... browse to PatternMaker.tcm, select it, Open
 - Close
- The macro tool bar should then pop up with the macros available for use.

Creating the Pattern

I'm pretty sure that brushes are drawn in 'drawing units' so it's probably best to draw your patterns in the unit you usually use. If you want to distribute your patterns, the majority seem to be in inches. The process is as follows

PatternMaker uses the concept of a 'repeat cell', the contents of which are indefinitely repeated grid-fashion to create patterns.

- First draw a rectangle to represent the 'repeat cell' of the pattern. It can be any dimensions, and can also be rotated to any angle
- Draw your pattern inside the 'repeat cell' (actually you can draw outside it as well if that's convenient)
- The pattern must be drawn completely in single lines ... if necessary draw normally then explode down to two-point lines. The two following examples are both valid patterns.



The text one was simply made by using the txt.shx font, exploding down to lines. The star pattern was made initially in a square repeat cell, then all graphics including the repeat cell were scaled and rotated.

So in summary the macro will expect to find the rectangle as the first graphic in the pattern, and all the other graphics must be simple two-point lines. It will fail if it finds something else. If your rectangle/pattern has got out of order you can set the rectangle back to the first graphic by selecting the rest of the pattern (not the rectangle), then Edit Cut, Edit Paste.

- Save the tcw, in case you want to edit the pattern later.
- Once you have completed your pattern, Select it including the rectangle and run the CreatePattern macro.
- The macro will create two files that you can edit into TC's settings using your normal methods...
 - an xml file which can be merged into TC-X settings – eg using the BCiTool_XI/XMLINI utility.
 - a txt file which can be manually added into TC9 (and before) settings – eg using Notepad
 - Note you have to shut TC down while making the changes.
- You're Done. Too easy? **Well, yes, there is a catch ...**

Only Certain 'Repeat Cell' Patterns are possible! ... read on ...

Limitations of the PatternMaker method of creating patterns

The PatternMaker method of using a 'repeat cell' is a fairly intuitive way to draw them IMO ... however ... this is not how TurboCAD draws patterns – even if that's what it looks like! (details in Tech Corner)

Only certain 'repeat cell' style patterns suit TurboCAD's method of drawing patterns. Specifically, the patterns must be drawn using a limited range of 'compatible' line angles ... in fact only 32 are allowed and half of these are cancelled out because they are the reverse of the other half!

Effectively sixteen line angles may sound a bit limiting, but keep in mind this is just a pattern we are making ... most patterns use only eight angles (effectively four).

Exactly which angles are deemed 'compatible with the repeat cell' depends on dimensions and rotation of the repeat cell rectangle.

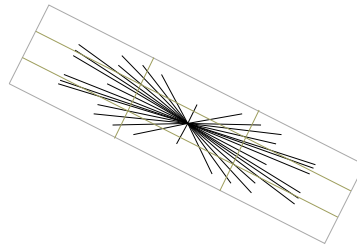
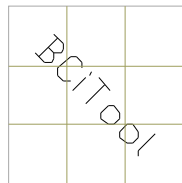
For example in a 1" x 1" repeat cell ...

- 0° , 45° and 90° are all OK,
- but 22.5° is out,
- and so are 30° and 60°
- On the other hand you can use 26.57° and 63.73° .
- All of the above changes depending on the repeat cell rectangle!

So which line angles are valid for a given Repeat Cell?

You can use the following procedure to construct angles compatible with the current Repeat Cell...

- Construct a 3 x 3 grid that fits inside and is correctly aligns with the repeat cell rectangle.



- Any angle you can draw between the 16 vertices of this 3 x 3 grid is OK.
- Position and Size of lines doesn't matter, so just use the grid to get the angles right, then shift/stretch it into position/size.
- For stretched/rotated repeat cells such as the example on the right, it is often easier to create the pattern in a plain square cell, then stretch and rotate the whole group later.

Managing Patterns

There are a couple of related issues to consider before you start making patterns by the thousand.

Rogue patterns

In Turbocad, every time you make a new drawing, all of TC's current patterns are copied into it. So if you create a drawing with a half-finished pattern in TC ... it gets copied to the drawing ... and once a pattern is in a drawing, it can be quite difficult to remove. And what happens when later on when you finish the pattern and save it in TC's settings, and re-open the drawing? Which is pattern is displayed – the one in TC, or the one in the Drawing? I don't know.

The moral of the story is that when developing patterns make this your only task for the moment. After inserting them into TC's settings, reopen TC and check them in a New From Scratch drawing. If they are not right, shut down TC immediately without saving anything, and remove the pattern from the ini file ASAP!

Keeping the number of patterns down.

The second issue is that while there is nothing to stop you adding as many patterns to the settings file as you want, this can lead to unnecessarily bloated drawings, because they all get loaded into each new drawing.

The following procedure can be used to create a separate patterns collection drawing, so you can import specific patterns into drawings only when you need them.

- (Shut down TC)
- In TC's ini file replace all the brushes with the ones you want to import (If you are using TC-X/XMLINI use **Replace** Settings with the appropriate merge file). TC should now only have the new patterns (plus Solid & None) in its setting file.
- Restart TC
 - New from Scratch (or open a file that already contains a pattern collection).
 - Optionally you can use the PatternSampler macro to make a sample page from the new patterns.
 - File Save
- Shut down TC
- Restore TC's ini file back to its normal state (If you are using TC-X/XMLINI use **Undo Replace** or **Reload**)
- Restart TC

Ok, so now we have TC with a normal set of brushes, and a saved drawing that contains an alternative collection of patterns. To use the patterns in a new drawing, do as follows

- With TC-X ...
 - Open the TC-Explorer palette
 - Browse to the Collection drawing, then right-click on it's thumbnail and **Force Extract** – this will expose it's objects in the palette without actually opening the file.
 - Still in TC Explorer, browse to and open the file's Brush Styles collection
 - Drag the brushes you want into the working drawing area.
- With TC-9 and before ...
 - I think the easiest way is to copy an object containing the pattern from the Collections drawing to the new drawing

Tech corner - as far as I can make out ...

As noted above TC does not use a 'repeat cell' method for drawing patterns. Instead it draws a series of infinite sets of infinitely long dashed lines. Well, infinite in the sense that it just keeps drawing until the relevant space is filled. The line sets can be drawn with any angle, dash length or x-y offsets from one-another. It is the interaction between these different line sets that creates the pattern.

TC's pattern drawing function can handle any line parameters. But unless the parameters of items within a series are carefully chosen they will not create a regular pattern. In fact, normally most combinations would result in non-repeating patterns such as the CONCRETE example that ships with TC. The only reason most patterns you find do repeat is that they have been specifically designed that way.

By contrast PatternMaker only allows regular patterns. It will not create non-repeating elements.

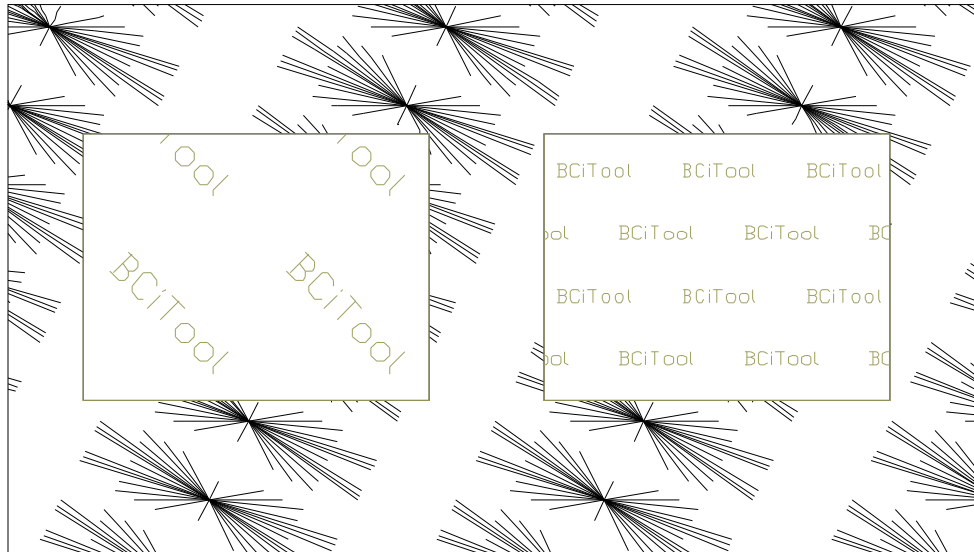
In the end, the parameter that really separates TC's method and PatternMaker's method is the line angle – all the others can be accounted for. The actual rule is, that after normalising to allow for the dimensions and angle of the repeat cell, you can use any line angle where $\tan(\text{angle}) = \text{rational number}$...

- $\tan(0^\circ) = 0.0 = \text{rational}$
- $\tan(26.57^\circ) = 0.5 = \text{rational}$
- $\tan(30^\circ) = 1/\sqrt{3} = \text{irrational}$
- $\tan(45^\circ) = 1.0 = \text{rational}$
- $\tan(90^\circ) = \text{undefined but OK because is a special case}$

So as a general rule, 30° cannot be used

If you really need 30° , one workaround is to modify the repeat cell parameters so that after normalising it becomes a valid angle. However, this would in turn knock out other common angles like 0° , 45° and 90° (which ones depends on how you go about it). The next workaround is to use the approximation of 29.9987° , which has the rational tan value of $56/97$. But the consequence of this is that the distance between dashes within a single line have to be stretched to cover roughly 112 cells and to compensate the lines are brought closer together and multiplied by the same factor (I think).

PatternMaker restricts the number of allowable angles to $\tan(\text{angle}) = \pm 3/\pm 3$. This comes down to just 32 angles (or 16 depending on how you look at it), but I believe it balances reasonable flexibility without potentially affecting TC's redraw performance... and to be honest I don't know if the PatternMaker algorithms will reliably work on a bigger range.



But after all it's just a brush pattern isn't it? :-)

Have fun. DB