



IN THE WORKSHOP – Tip #13

Increase your productivity in Altium Designer with scripts

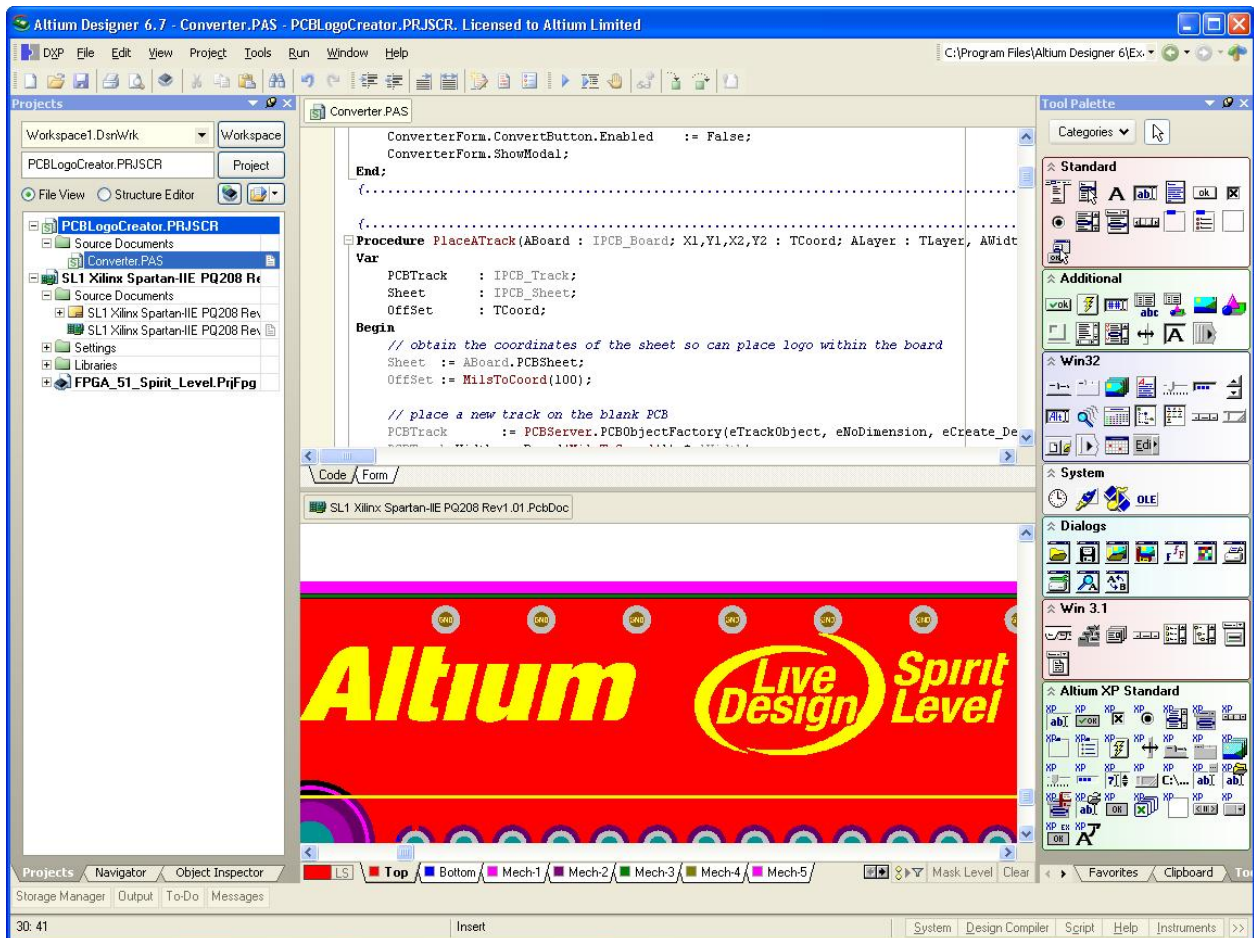
Summary

May 2007

Author: David Parker

Scripting provides a powerful method of increasing your capabilities and productivity with Altium Designer. Scripts can be written in a number of languages, complete with sophisticated dialogs, and debugged within Altium Designer.

Altium Designer features a scripting system that supports DelphiScript, Visual Basic and the Jscript languages. You do not need to use external development tools to write and debug your scripts, everything is included, from the professional-level syntax-aware text editor, the drag-and-drop form designer, through to the script debugger. Scripts can be used to automate simple repetitive tasks, generate custom reports, or directly manipulate design objects.



Altium Designer's scripting system allows you to develop and run scripts that perform actions on your design documents.

Why would I want to use a script?

You use scripts to achieve your design objectives with minimal user input; fewer keyboard presses and mouse clicks! With scripts you can automate repetitive tasks and perform edits that are not available with the existing commands in Altium Designer. For example, you could use a script to search and update fiducial marks on a PCB document, or to export a customized netlist.

Importing a bitmap of my company logo to a PCB using scripting

An example of a design challenge that could be efficiently solved by a script is that of logos on PCB files. The scenario is as follows: Every time I use Altium Designer's PCB Editor to create a PCB I need to include my company logo, which is a raster style bitmap. How do I achieve this? As far as I can tell, I can only place tracks and other vector type objects in the PCB Editor.

Altium Designer's PCB Editor is a vector based editor with a specific set of objects to work with. Since the logo is a raster (pixel-based) image, we need to a tool that converts a raster-based image to a set of vector objects in Altium Designer's PCB Editor.

How would a PCB Logo Creator script work?

We need a script that scans through the bitmap file, and converts each pixel into a track. As it does this it will test to see if adjacent pixels are **on** as well, and replace all of the adjacent-and-on pixels with a single track segment. The result will be a series of small horizontal tracks that build up the image.

To achieve this, the script will need to perform the following:

- Provide a dialog that allows the user to select the image, as well as present other options, such as scaling.
- Check if the bitmap is monochrome, and warn and abort if not. The bitmap has to go onto a single PCB layer, so it can only be constructed as a monochrome image.
- Scan the image file, checking to see if each pixel is on, or off.
- Collate adjacent on pixels so that they can be replaced by a single track object.
- Support scaling.
- Support mirroring, to simplify placement on the required layer.

The heart of the script is an algorithm that scans for black and white pixels and works out the adjacent colored pixels on the same line. For example, if there are 40 adjacent white pixels on the same line, this is translated as a one track. The length and width of a track is set by a user-defined scaling factor. Therefore if there are 40 pixels and the scaling factor is 1, then a PCB track of 1 mil width and 40 mils long is placed on the PCB document.

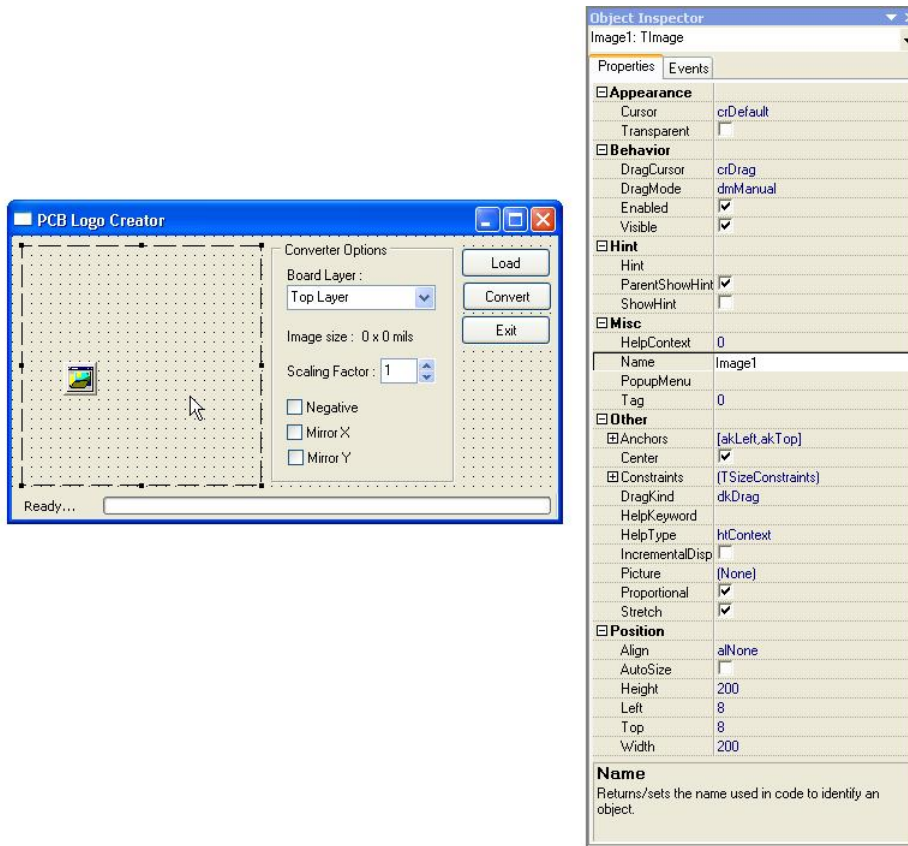
Rather than completely detailing the script here, if you are interested you can find a DelphiScript version of a PCB logo creator in the Altium Designer examples, details on its location are at the end of the article.



A logo converted into a series of PCB tracks by the example PCB Logo Creator script.

What sort of things can I do with a script?

Altium Designer's scripting engine supports DelphiScript, Visual Basic, and the Jscript languages. It also supports Enable Basic and TCL, but since these languages do not include the form design interface you will probably want to work with one of the three that do. With the form design interface you can create sophisticated dialogs or wizards that capture the script input requirements.



Design your own script dialogs (forms) using the DelphiScript form designer.

From your script you can interface to Altium Designer through two mechanisms. The first is via a process+parameter call, which works in exactly the same way as if you were using the standard menus, toolbars and shortcuts. For example, if you wanted to step through all the components in a PCB library and set the reference to the Center, you would run the process+parameter combo of `PCB:SetComponentReference + Location=Center`, followed by the process `PCB:NextComponent`. Written in DelphiScript it would look like:

```
ResetParameters;  
AddStringParameter('Location', 'Center');  
RunProcess('PCB:SetComponentReference');  
RunProcess('PCB:NextComponent');
```

A tip if you are exploring this approach, hold Ctrl as you click on a toolbar button or menu entry to see the process and parameters behind that command.

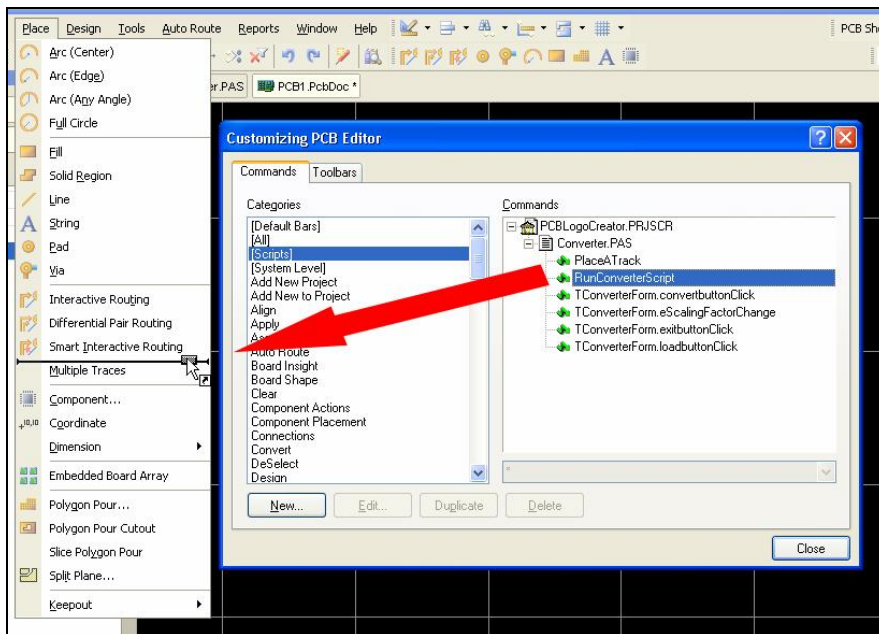
The second way of interfacing to Altium Designer from your script is via the Application Programming Interface. This programmatic approach is faster and far more powerful, giving you access to the same application-to-application functionality that Altium Designer uses itself when working between its different editors. This approach also gives you access to the Microsoft Windows API, and it is this approach that has been used to write the example PCB logo creator, as the code fragment below shows:

```
PCBTrack.X1 := Sheet.SheetX + MilsToCoord(X1) + Offset;  
PCBTrack.Y1 := Sheet.SheetY + MilsToCoord(Y1) + Offset;  
PCBTrack.X2 := Sheet.SheetX + MilsToCoord(X2) + Offset;  
PCBTrack.Y2 := Sheet.SheetY + MilsToCoord(Y2) + Offset;
```

How do I run a script in Altium Designer?

A script can include one or many procedures. For example, some might deal with the events of buttons, others deal with algorithms, and others handle file I/O. Each script has a main procedure as the starting point, so you need to invoke that procedure from the Run Script dialog to run it in Altium Designer (DXP menu).

That approach is fine for the occasional use of a script, for regular use it is better to hook the script up to a menu entry or toolbar button. That's quite straightforward: double click at the end of the menu bar to open the Customize dialog, locate [Scripts] in the categories, locate the procedure name in the Commands list, then click, hold, drag and drop it onto the menu. The Edit Command dialog will open, so you can enter a suitable Caption for the menu entry. You'll need the script open, or have it listed as an installed project in the Preferences dialog, to hook it up to a menu or button. But once it is, you no longer need it opened or installed.



Dragging a script procedure to the PCB's Place Menu.

Where to next?

The PCB Logo Creator script lives in the folder

```
C:\Program Files\Altium Designer 6\Examples\Scripts\Delphiscript Scripts\Pcb\PCB Logo Creator.
```

As well as numerous examples, there is also extensive documentation on scripting and the APIs. To explore the documentation, open the **Knowledge Center** panel and head for the **Configuring the System > Scripting in Altium Designer**. Try the **Tour of the Scripting System** for an overview, and **Getting Started with Scripting and Building Script Projects** to get yourself started.