Custom script

Version 1.0.1, by Giorgio Bianchini

Description: Executes custom code.

Module type: Plotting

Module ID: `cdb74bfb-8a90-48b3-815a-8f908d2a1ff5`

This module makes it possible to execute custom C# code to plot the tree. This can be useful either to draw one-off complicated graphics, or as a first step in developing a new module for TreeViewer.

Parameters

Description

Control type: Text box

Default value: Describe the script

This parameter can be used to provide a short description to quickly identify what the module does without having to look at the source code. It is ignored by the module.

Source code

Control type: Source code

Default value:

```
using PhyloTree;
using System.Collections.Generic;
using VectSharp;
using TreeViewer;

namespace ae30070d8629a4f6fab2cfbe38c6b9c3c
{
    //Do not change class name
    public static class CustomCode
    {
        //Do not change method signature
        public static Point[] PerformPlotAction(TreeNode tree,
 Dictionary<string, Point> coordinates, Graphics graphics,
 InstanceStateData stateData)
```

```
        {
            Point topLeft = new Point();
            Point bottomRight = new Point();
            return new Point[] { topLeft, bottomRight };
        }
    }
}
```

This parameter contains the source code of the script. The arguments to the `PerformAction` method are as follows:

- `tree`: the final transformed tree that has gone through all the further transformation modules.
- `coordinates`: a dictionary associating the Id of each node in the tree to its coordinates. You can use the script to change the coordinates of some nodes; however, other Plot action modules will not be automatically notified of this and you will have to invalidate either the Coordinates module or all of the other Plot action modules for the changes to be applied.
- `graphics`: the graphics surface on which the plot should be drawn.
- `stateData`: an `InstanceStateData` object that can be used to access features in way that does not depend on the program running in command-line or GUI mode.

Further information

The difference between this module and the other module with the same name is that this module is a Plot action, while the other *Custom script* module (id `a76d00d2-95e0-4274-a77d-1439a013e3d9`) is instead a Further transformation.

The code in the module can do anything, including loading additional data from a file on disk. However, this is discouraged, becaus it ties the tree file on the computer it was created on. A better approach to load additional data would be to import the data file as an attachment and read the data from the attachment. Attachments can be accessed using the `Attachments` property of the `stateData` object that is passed as a method parameter.

Furthermore, since the code in the module can do anything, it may also be a security risk to open files originating from unknown sources; thus, you should either make sure that any file you open comes from a reputable source, or avoid loading source code from tree files at all.