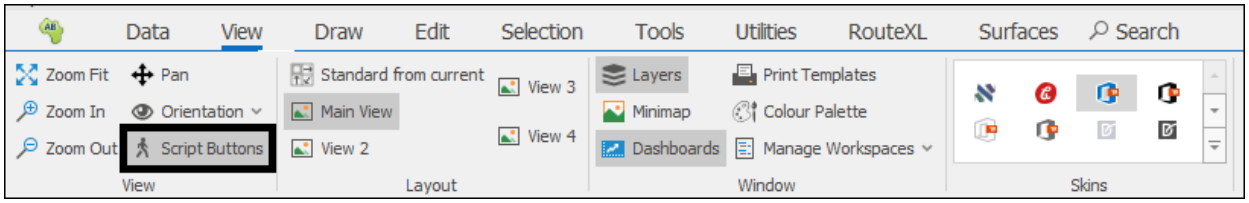
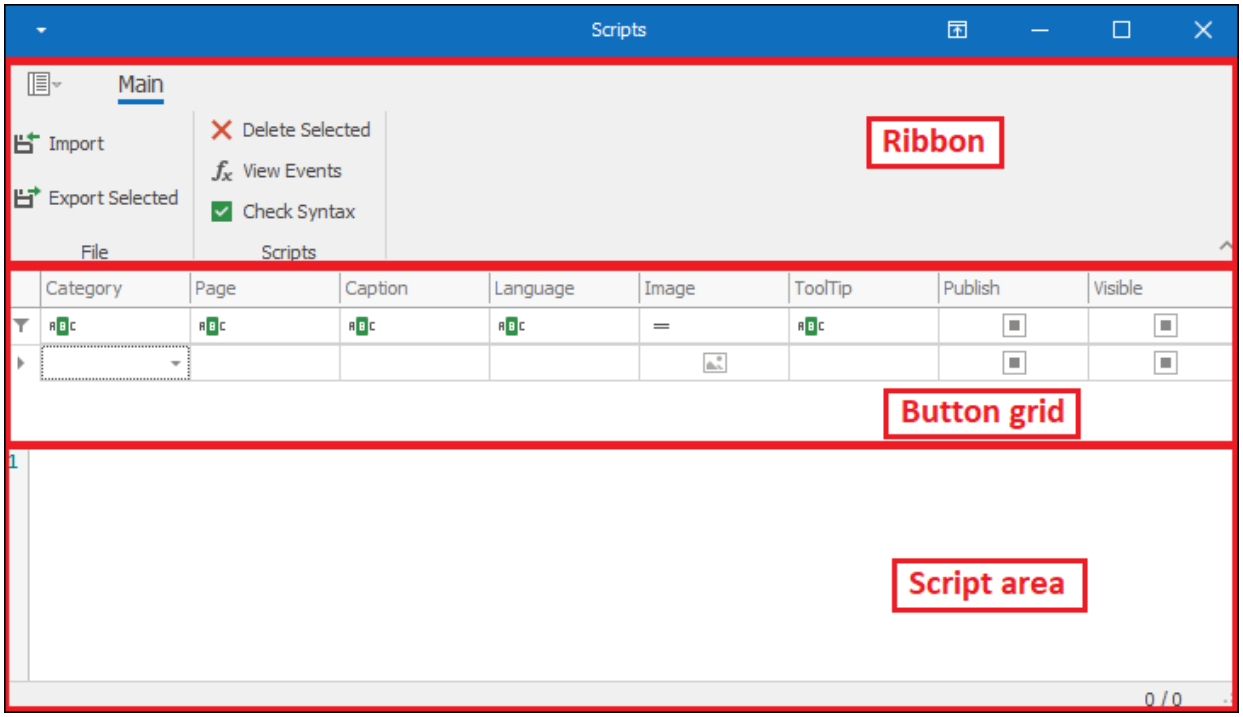
PrimeThought Software Solutions

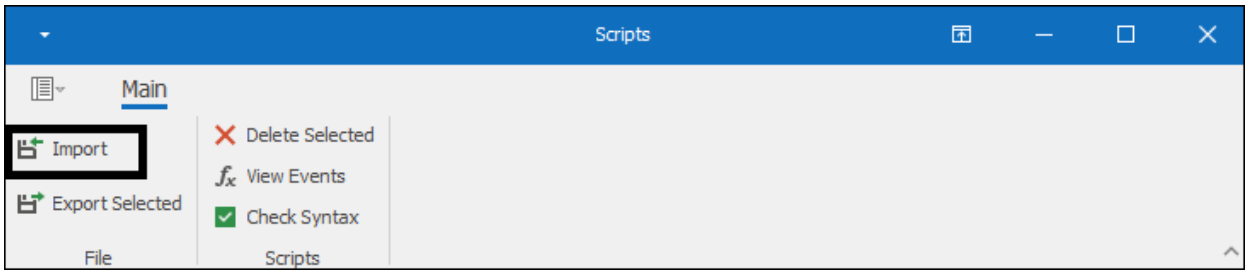
Script Buttons User Guide

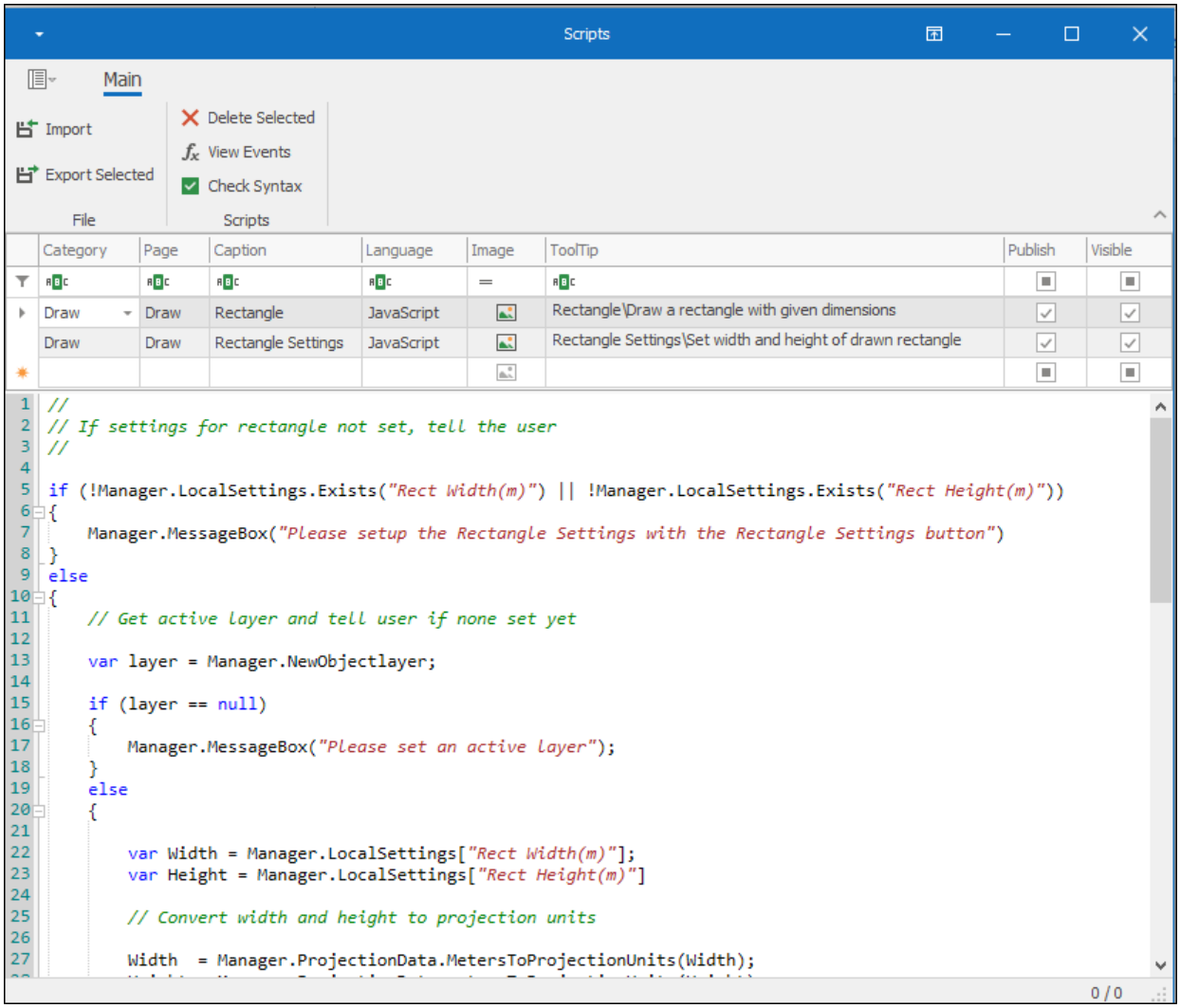


Script Buttons is a feature available in all our spatial products that allows you to write your own scripts for various buttons, with various functions, you would like to create. Within the spatial pane itself you are able to add your own tools. The following is the layout of the dialogue it brings up:



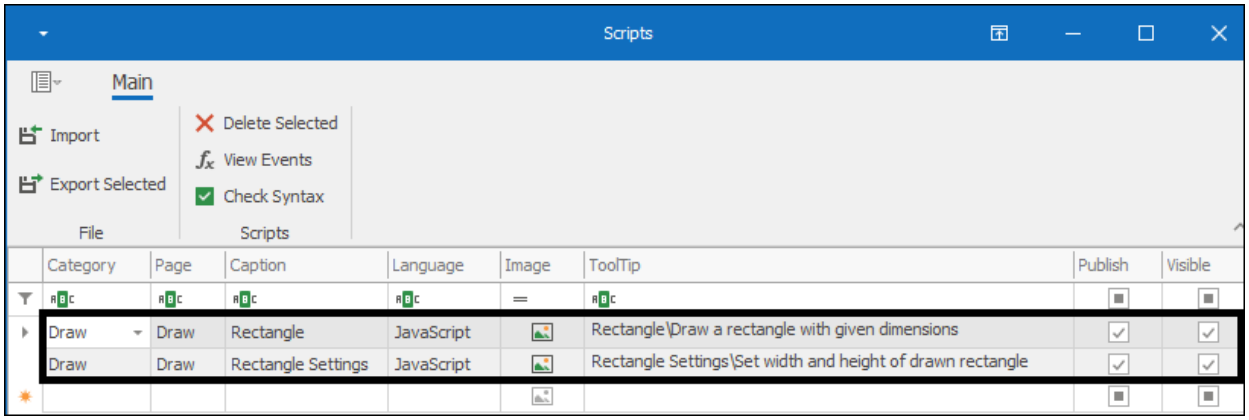
To import already created scripts you would use the **Import** button, that is what I will be doing in this example:



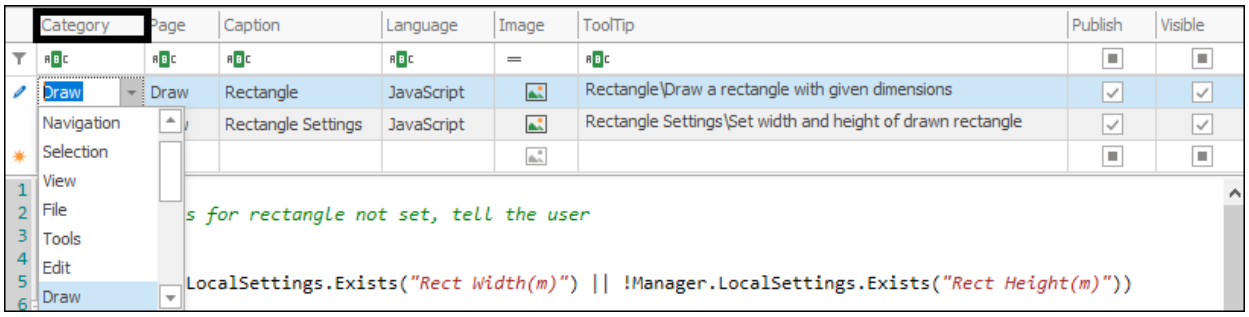


As you can see my scripts have now been imported.

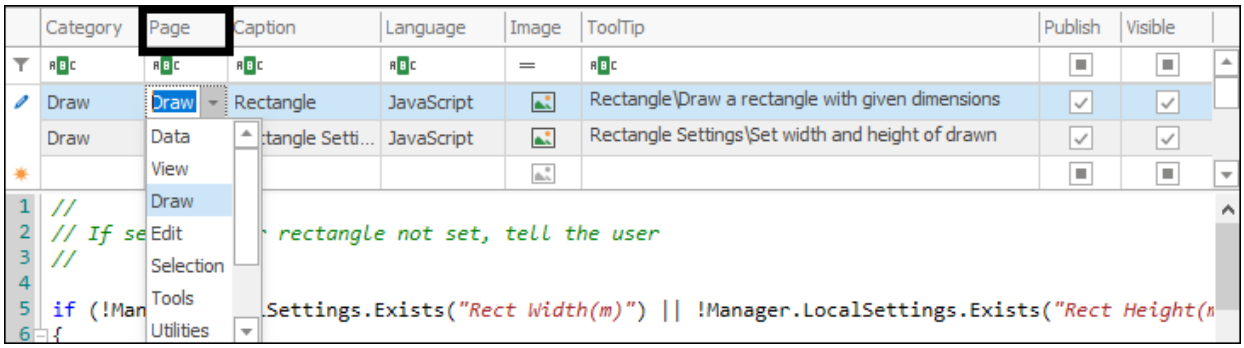
Now, I will start by going over the Button grid, which is where you set up the buttons. In the first row I have set up a button called **Rectangle** which will draw rectangles of a predefined height and width when you click at a point in the scene. In the second row is the **Rectangle Settings** button that I have set up which will bring up a dialogue where you can set the parameters of the rectangle:



The **Category** is what kind of tool it is; you can select from the dropdown list or type in your own:



The **Page** is what tab you would like the tool to appear in. You can select from existing tabs or type in your own to create a new tab:



The **Caption** is the name of the button which will be displayed. **Language** is what scripting language you would like to use, right now we only support JavaS-cript:

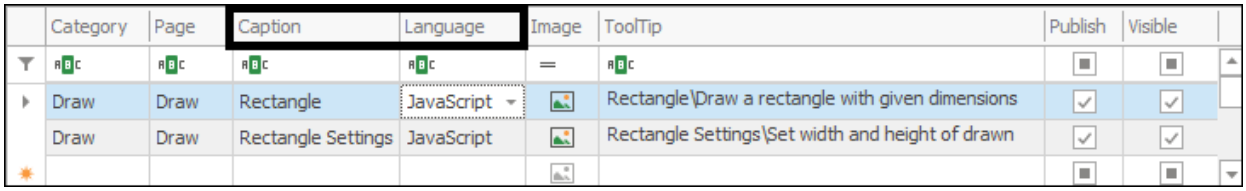
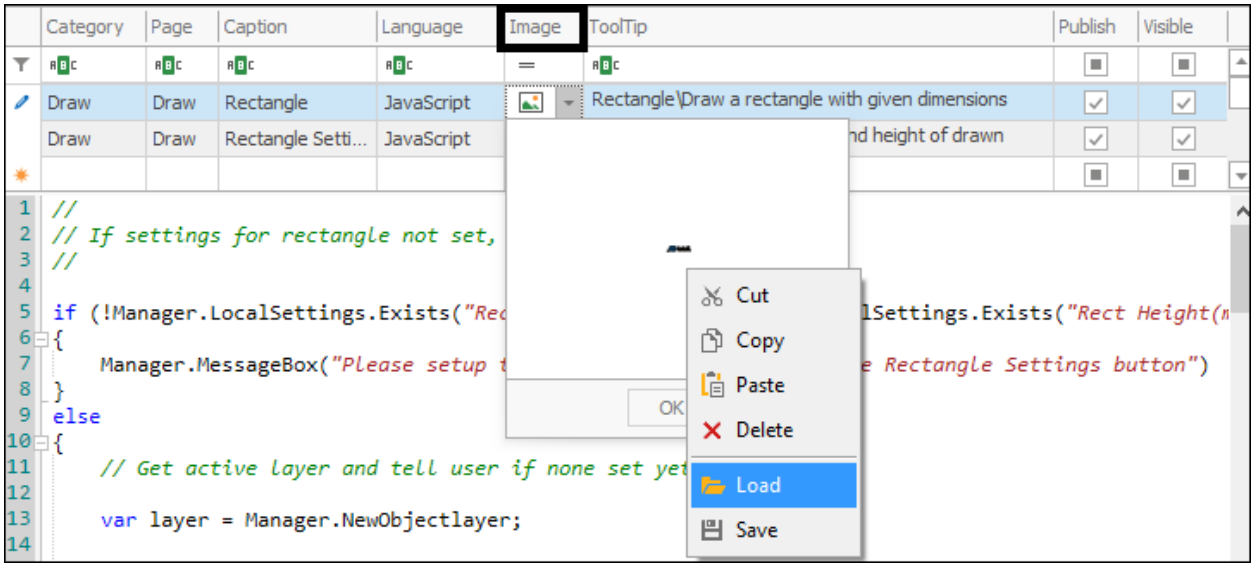
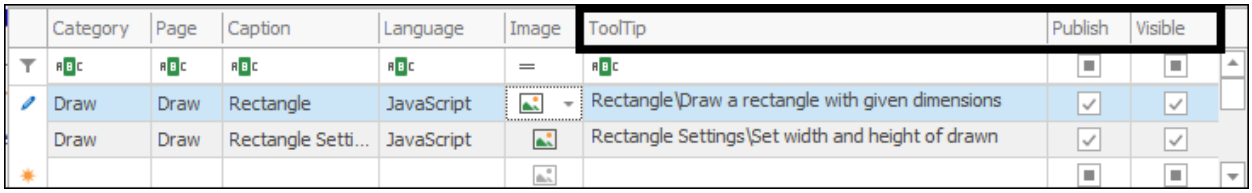


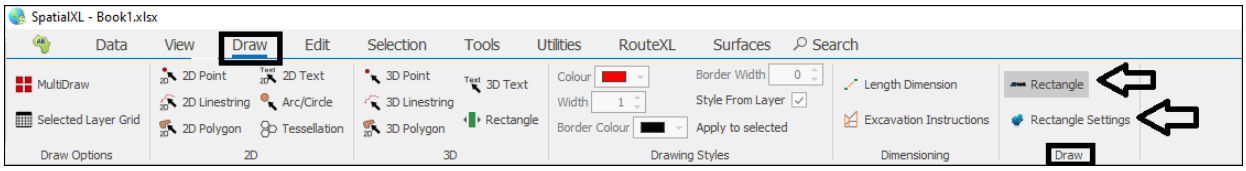
Image is where you can choose an image to be displayed next to the button, dropdown and then right click to load an image:



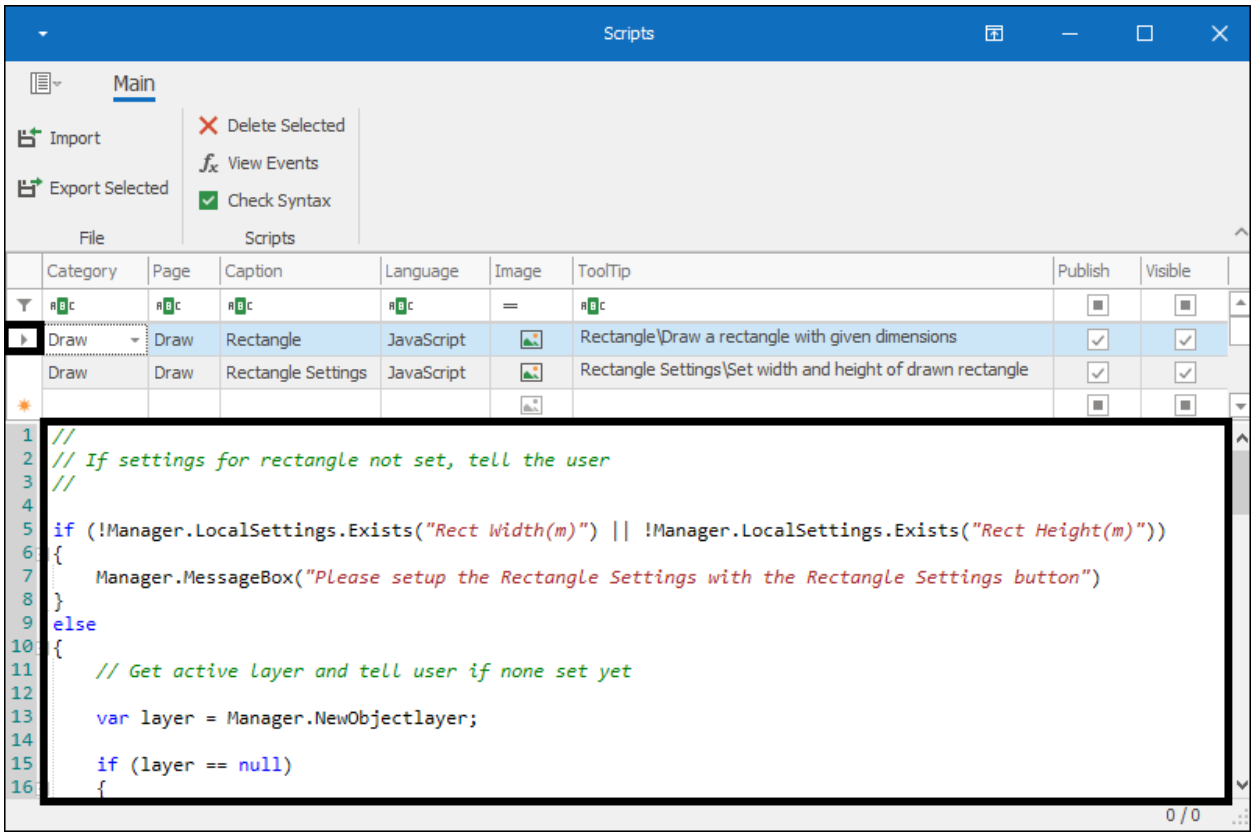
ToolTip is the tooltip you would like to be displayed for the button. **Publish** is whether you would like the button to be published when you are publishing Geoscope files. And **Visible** is whether you would like the button to be visible or not in the tab:



Now that I have set up these buttons here, if you go to your spatial pane you will see the buttons there:



Now, next we come to the script. When you select one of the buttons in the grid the corresponding script for the button comes up in the script box, or if there was no script for it you would write it in the box below making sure you first select the button in the grid. I have selected the **Rectangle** button and its associated script has come up:

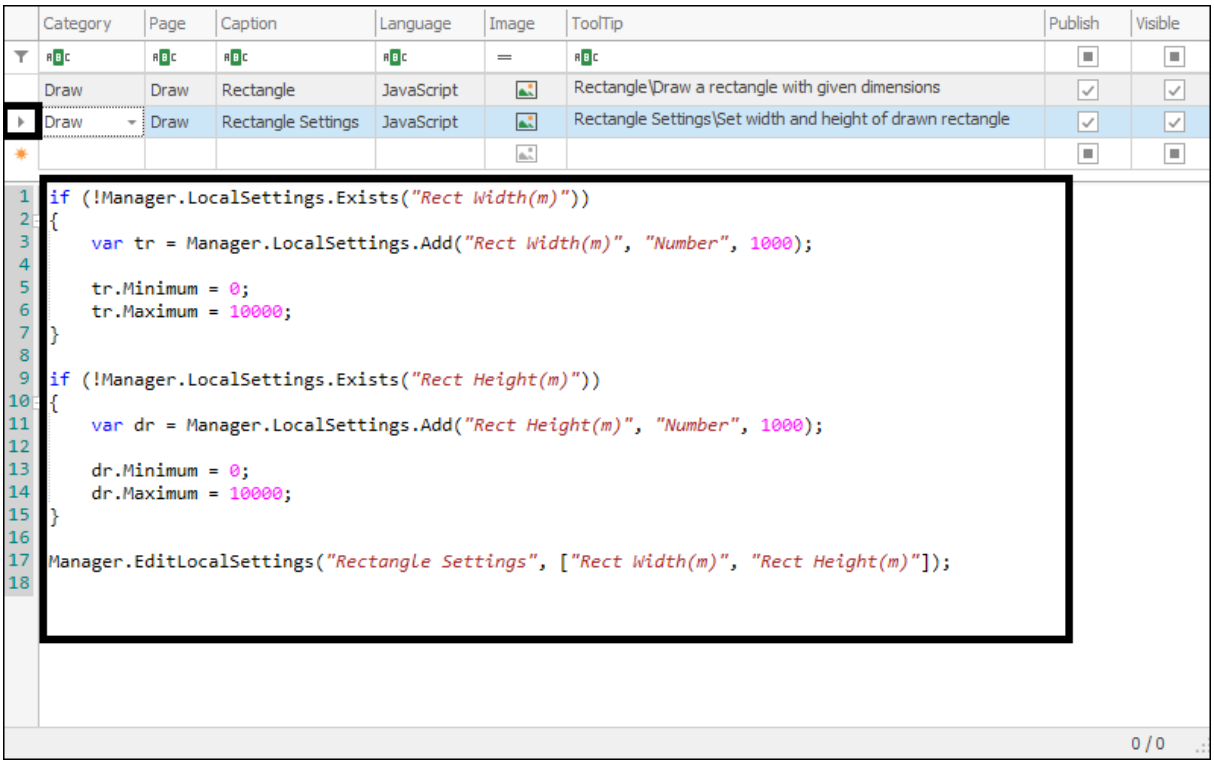


The script created for this draw rectangle button goes like this:

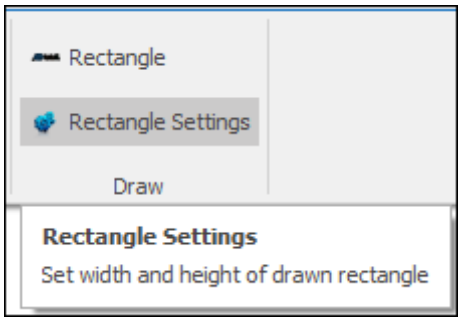
```
//
// If settings for rectangle not set, tell the user
//

if (!Manager.LocalSettings.Exists("Rect Width(m)") || !Manager.LocalSettings.Exists("Rect Height(m)"))
{
    Manager.MessageBox("Please setup the Rectangle Settings with the Rectangle Settings button")
}
else
{
    // Get active layer and tell user if none set yet

    var layer = Manager.NewObjectlayer;

    if (layer == null)
    {
        Manager.MessageBox("Please set an active layer");
    }
    else
    {

        var Width = Manager.LocalSettings["Rect Width(m)"];
        var Height = Manager.LocalSettings["Rect Height(m)"]

        // Convert width and height to projection units

        Width  = Manager.ProjectionData.MetersToProjectionUnits(Width);
        Height = Manager.ProjectionData.metersToProjectionUnits(Height);

```

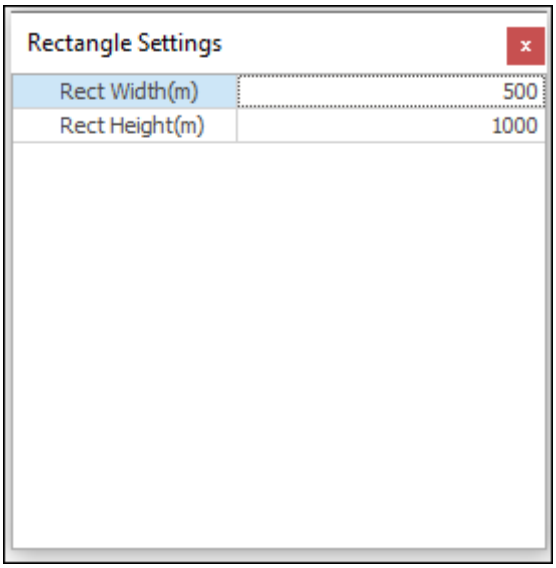
```
        // Function to be called on Mouse click to draw rectangle

        function DrawRect()
        {
            // Get world coordinates of poise location

            var pos = Manager.MouseWorldLocation();

            // Pointer to the PrimeThought Spatial namespace

            var PTS = importNamespace('PrimeThought.Spatial');

            // Calculate the corner points starting from bottom left and going counter clockwise

            var p1 = new PTS.SPoint2D(pos.X - Width / 2, pos.Y - Height / 2);
            var p2 = new PTS.SPoint2D(pos.X + Width / 2, pos.Y - Height / 2);
            var p3 = new PTS.SPoint2D(pos.X + Width / 2, pos.Y + Height / 2);
            var p4 = new PTS.SPoint2D(pos.X - Width / 2, pos.Y + Height / 2);

            // Add a polygon from these points to layer

            var poly = new PTS.Polygon();

            poly.OuterRing.Add(p1);
            poly.OuterRing.Add(p2);
            poly.OuterRing.Add(p3);
            poly.OuterRing.Add(p4);
            poly.OuterRing.Add(p1);

            layer.AddGeometry(poly, null);

            // Refresh the screen

            Manager.RefreshScenes();

            // Remove the click handler

            Events.RemoveHandler("Click");

            // Set the Prompt to empty string

            Manager.Prompt("");
        }

        Manager.Prompt("Please click to draw the Rectangle");

        Events.AddHandler("Click", "DrawRect");
    }
}
```

Then when I select my **Rectangle Settings** button its associated script will come up. This script creates the dialogue box where I can set the parameters for my rectangle. And its script goes like this:

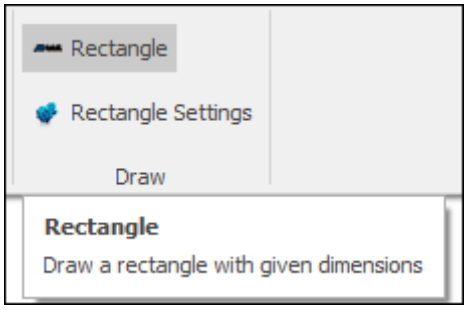


Let's now look at these buttons in action. First, I click on the **Rectangle Settings** button and set the parameters for my rectangle, which will be 500 metres width and 1000 metres height:

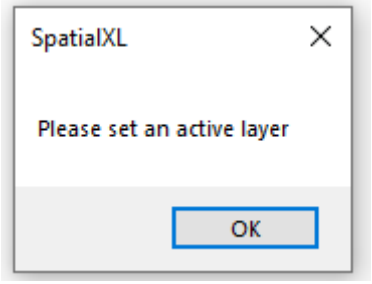




I can then go ahead and click my **Rectangle** button and then click on the point on the scene where I would like to have a rectangle, of the dimensions set, drawn:

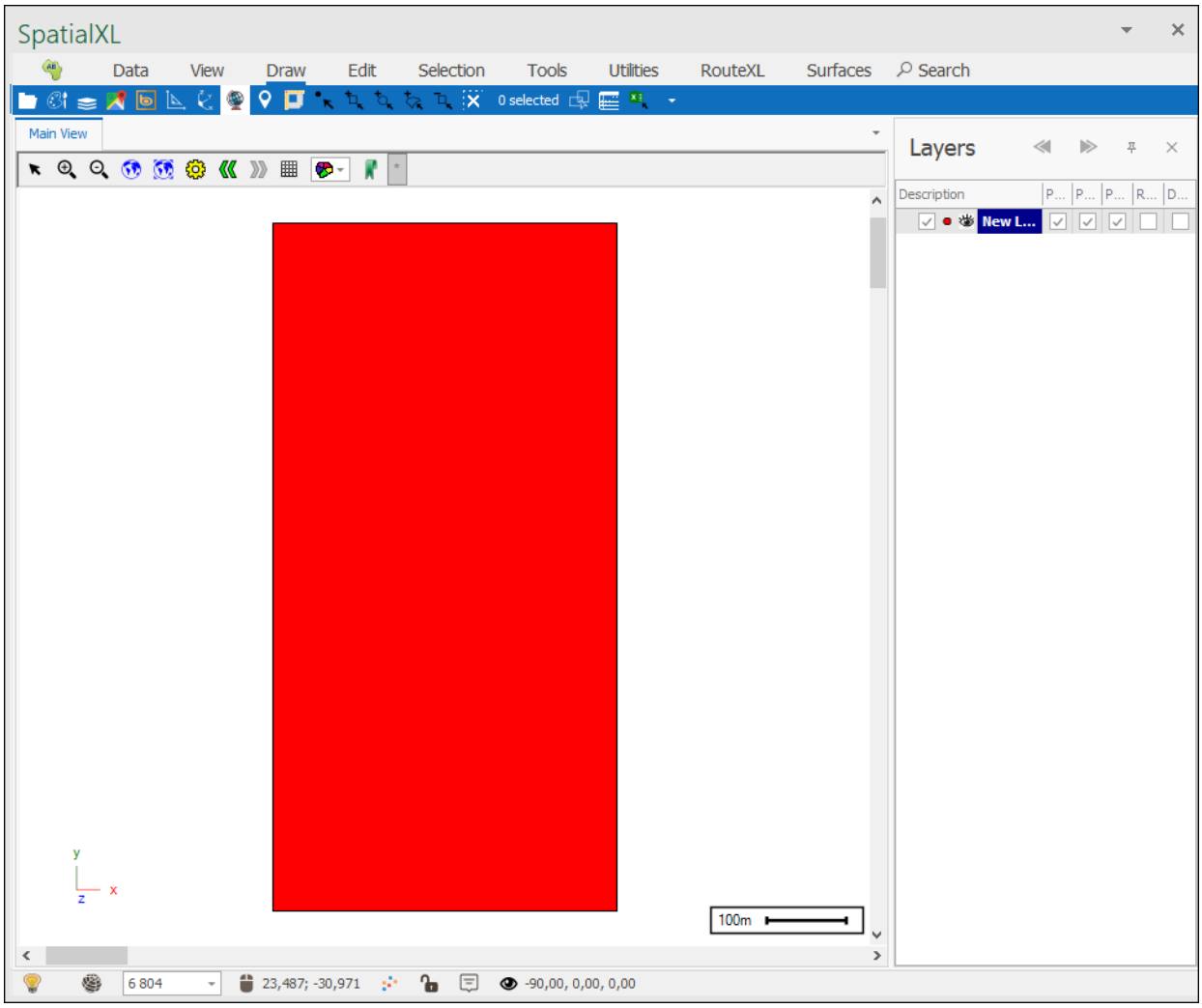


I clicked the tool but then got this error message because I don't have any active layer set for the rectangle to be drawn to. This is what I set up in my script to happen if no active layer was set:

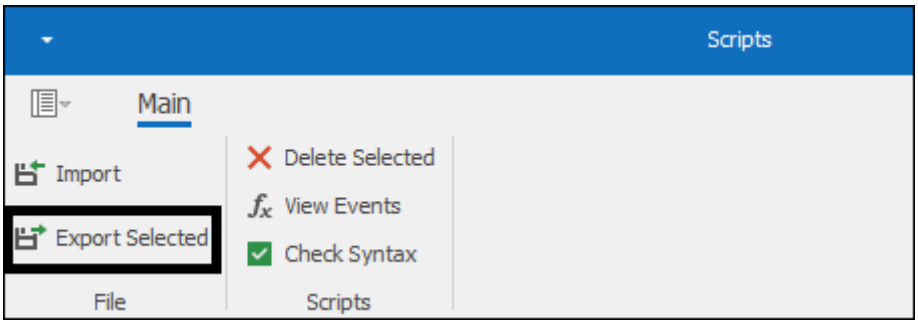


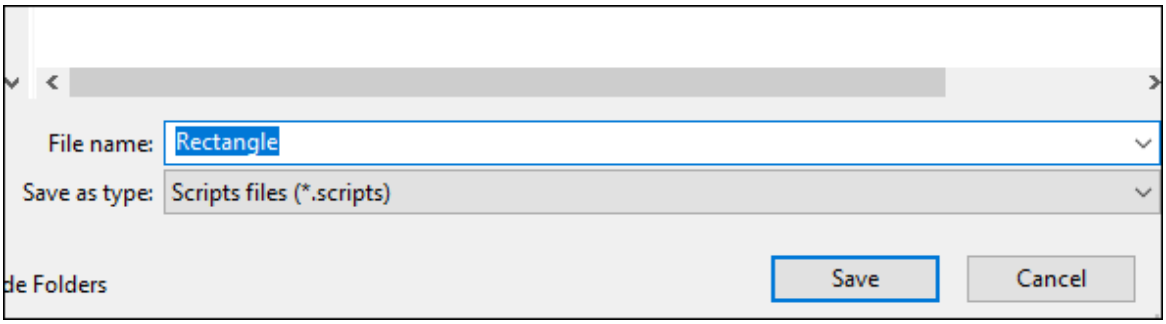
I will therefore set an active layer. I then click on the tool again and now it is ready to work as shown in the status bar in the spatial pane (a prompt which was set up to happen in the script as well):

I click in my scene at the desired point and my rectangle has been drawn:

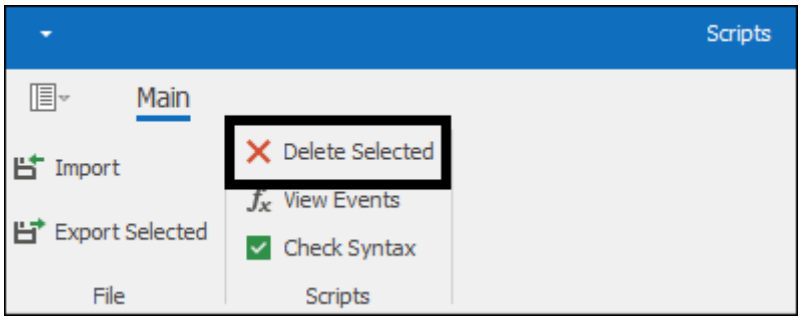


I am able to export these scripts using the **Export Selected** button which will export the selected button's script, I browse to where I'd like to save it and it will be saved as a **.scripts** file:



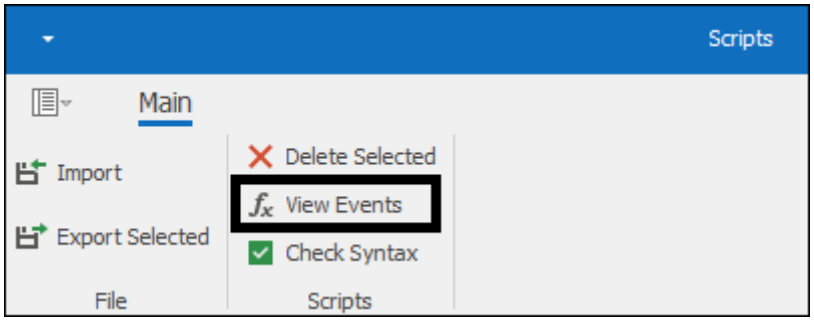


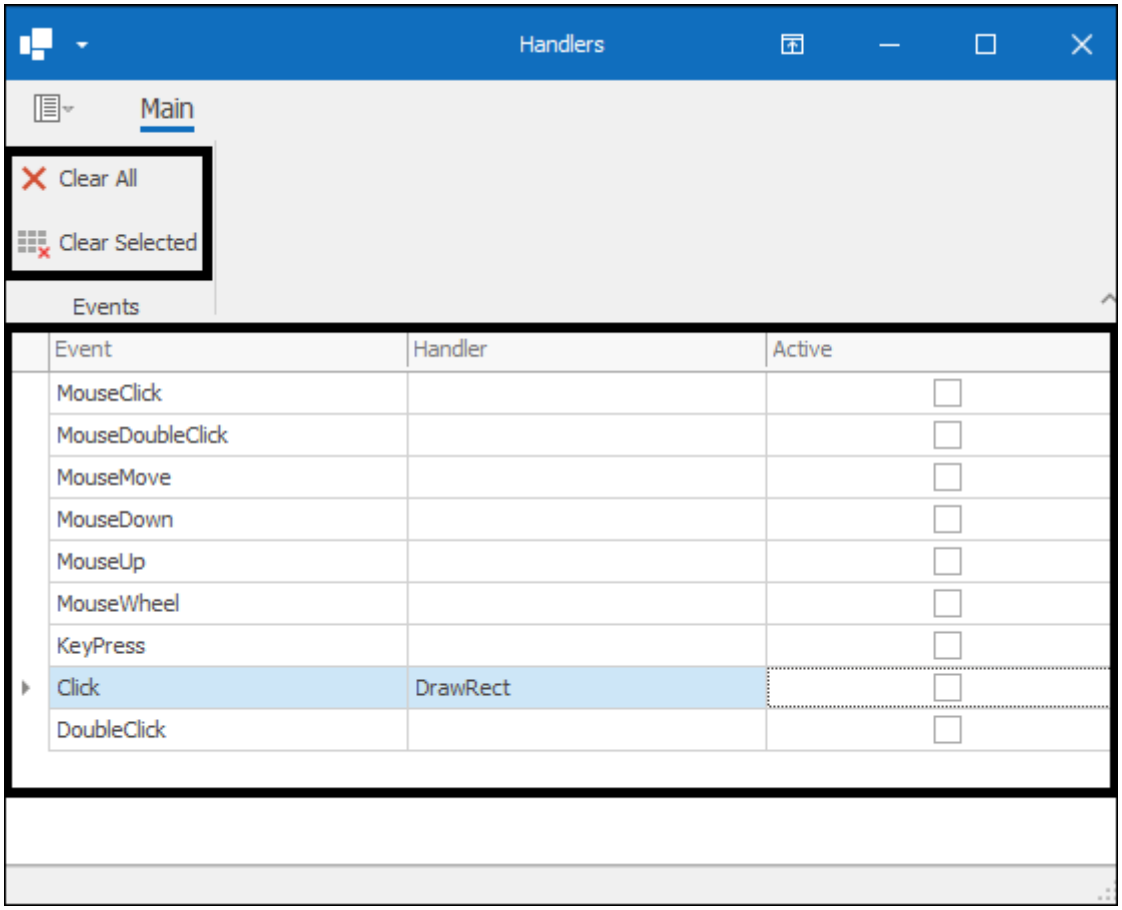
To delete any of the buttons you created select the button in the grid and click **Delete Selected**:



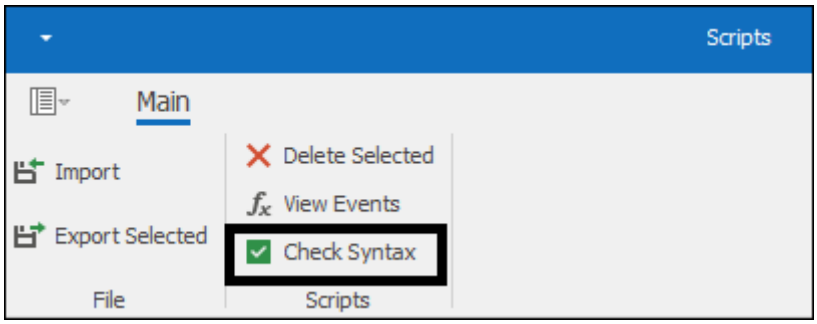
You can view and manage the events that have happened in relation to a specific script by clicking on the **View Events** button which will bring up a **Handlers** dialogue with the different events; here you can clear any handlers currently active (which will be shown by a tick next to it). In here you can see I have a handler

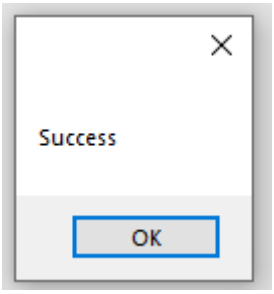
for the **Click** event which is when I click in my scene for the rectangle to be drawn, and the handler is the **DrawRect** function that I set up in my script:





The **Check Syntax** button will do a check of the syntax of your script and if good will give the following message:





Support

T: +27871354351

support@primethought.biz - primethought.biz
Kyalami Estate, Midrand, Johannesburg,
1684, South Africa