

GOverlay Plugin Documentation

This documentation aims to explain the flow a plugin

When GOverlay starts

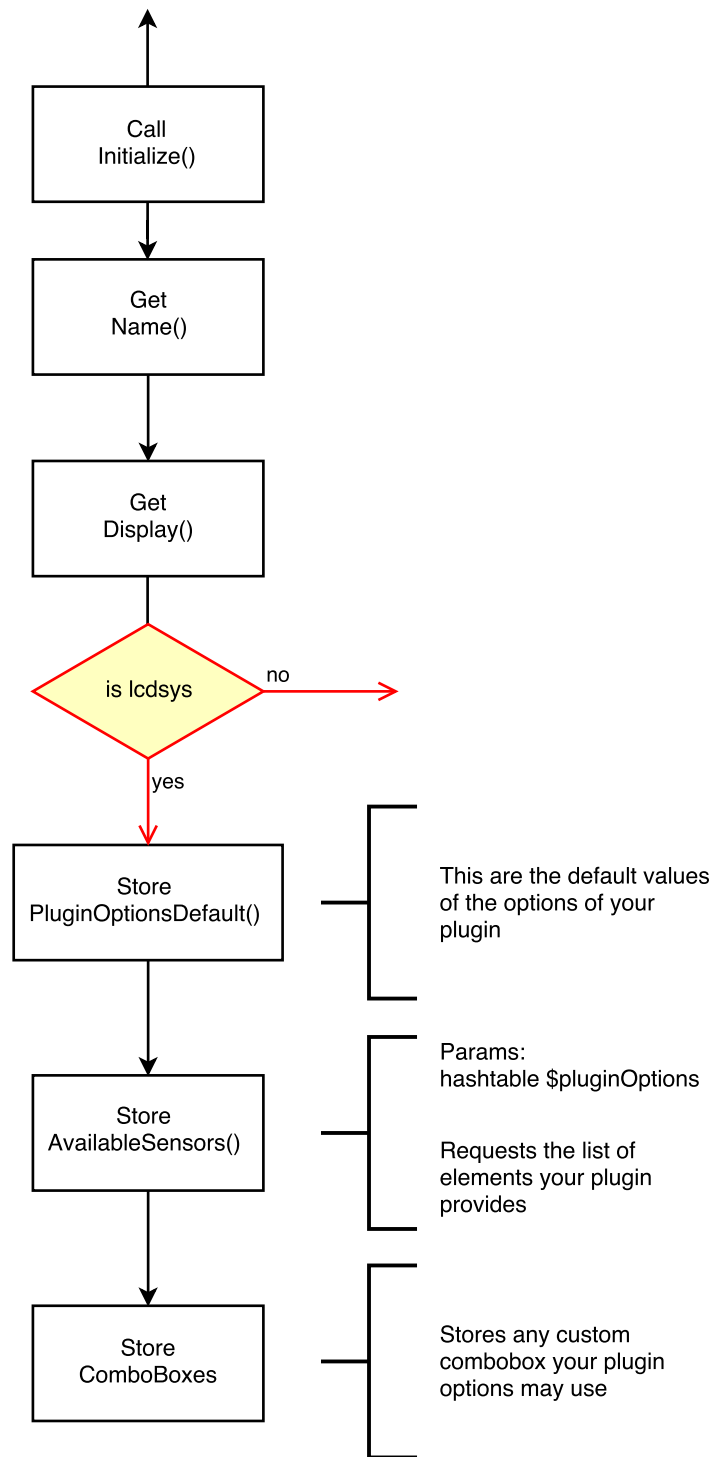
Every time GOverlay starts it calls the plugins to know which ones are available

Expected result

Goverlay needs to know if your plugin is valid, the name, display type and the default options so it can configure it. It will also request the elements it has and any custom ComboBoxes it uses so it can store them for later use

Flow

Goverlay Initializing



Example code

```
Public Sub Initialize(ByVal Host As GOverlayPlugin.Interfaces.IHost) Implements GOverlayPlugin.Interfaces.IPlugin.Initialize
    objHost = Host
    hwinfo = New Hashtable
End Sub
Public Function PluginOptionsDefault() As Hashtable Implements GOverlayPlugin.Interfaces.IPlugin.PluginOptionsDefault
    Dim options As New Hashtable
    options("unit") = "celsius"
    options("note") = "for demo purposes"
    options("lineColor") = "65535" 'white
    Return options
End Function
Public Function ComboBoxes() As Hashtable Implements GOverlayPlugin.Interfaces.IPlugin.ComboBoxes
    Dim boxes As New Hashtable
    Dim myboxOptions As New Hashtable
    myboxOptions.Add("celsius", "Celsius Unit")
    myboxOptions.Add("fahrenheit", "Fahrenheit Unit")
    boxes.Add(pluginId & "unit", myboxOptions)
    Return boxes
End Function
```

When the user enters the plugin options tab

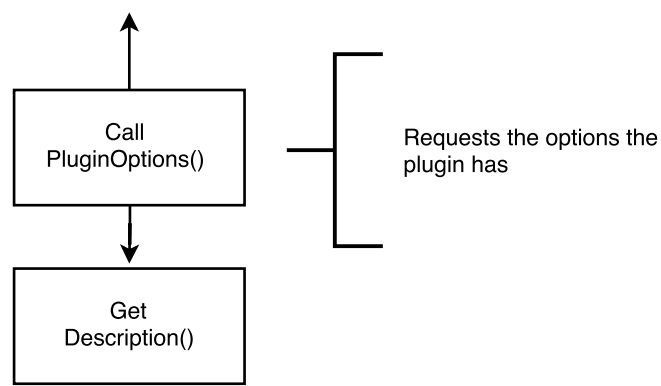
When a user enters the plugins options tab and selects your plugin, this is is the flow

Expected result

GOverlay needs to know the options you provide globally for this plugin. This options are not individual to the elements you have but for the plugin itself, for example, it could be the path of a .INI file the plugin uses or a metric for all the elements you provide to use.

Flow

GOverlay Plugin Options Tab



Example code

```
Public ReadOnly Property Description() As String Implements GOverlayPlugin.Interfaces.IPlugin.Description
    Get
        Return "An example plugin." & vbNewLine & vbNewLine & _
            "This plugin performs a complex draw (drawing ourselves) as well as keeping the simple draw of the previous example."
    End Get
End Property
```

Example code

```
Public Function PluginOptions(pluginCurrentOptions As Hashtable) As Hashtable Implements
GOverlayPlugin.Interfaces.IPlugin.PluginOptions
    Dim options As New Hashtable
    'Option: option_index as integer, option_data as ArrayList
    'Option_Data: option_type as string, option_label as string, option_name as string (no spaces, no _)
    options.Add(0, Helper.addPluginOption("Text", "A note", "note"))
    options.Add(1, Helper.addPluginOption(pluginId & "unit", "Type of unit to use", "unit"))
    options.Add(2, Helper.addPluginOption("ColorRGB", "Line Color", "lineColor"))
    Return options
End Function
```

When the user enters the profile/onWindows/whileGaming tab

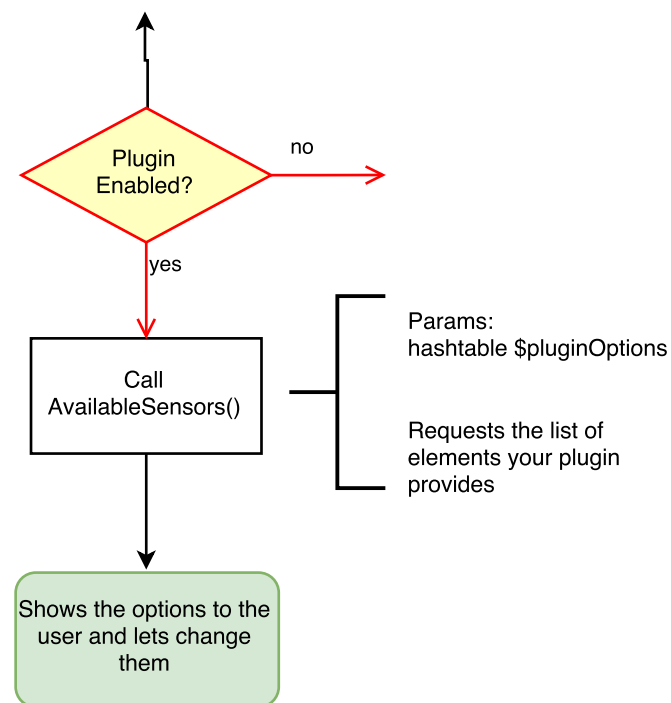
This flow happens when the user enters the "Profile" tab on lcd2 or "OnWindows/WhileGaming" on lcd1

Expected result

Every time the user enters this tabs GOverlay will fetch again the available plugin elements so it can display them on the list of elements the user has to use.

Flow

Goverlay Entering LCDSys Tab



Example code

```
Function AvailableSensors(pluginOptions As Hashtable) As System.Collections.Generic.Dictionary(Of String, String) Implements GOverlayPlugin.Interfaces.IPlugin.AvailableSensors
```

```
'Create the list of the sensors/elements this plugin has
```

```
'You can access your pluginOptions here as pluginOptions(your_option)
```

```
objHost.DebugMessage("DemoPlugin - Listing the plugin elements available")
```

```
Dim selectedUnit As String = pluginOptions("unit")
```

```
'Options: SensorTag, Sensor Display-Name
```

```
Dim sensors As New System.Collections.Generic.Dictionary(Of String, String)
```

```
sensors.Add(pluginId & "temp", "Temperature (in " & selectedUnit & ")")
```

```
sensors.Add(pluginId & "usage", "Usage")
```

```
sensors.Add(pluginId & "line", "Line")
```

```
sensors.Add(pluginId & "square", "Square")
```

```
sensors.Add(pluginId & "text", "A random text")
```

```
Return sensors
```

```
End Function
```

When the user clicks on your plugin element to add it

This flow happens when the user has picked your element in the list to add it to his screen, this is the initial setup of the element

Expected result

GOverlay will need to know if you are using a custom draw for this element, because if it does, GOverlay does not put all the regular default options for the element (since in this case, you define every single option the element has).

If its not a custom draw, GOverlay will set up its regular default options because it will provide the drawing, meaning:

If the user added this element by clicking on it:

This element will have font options, color, threshold, format, etc.

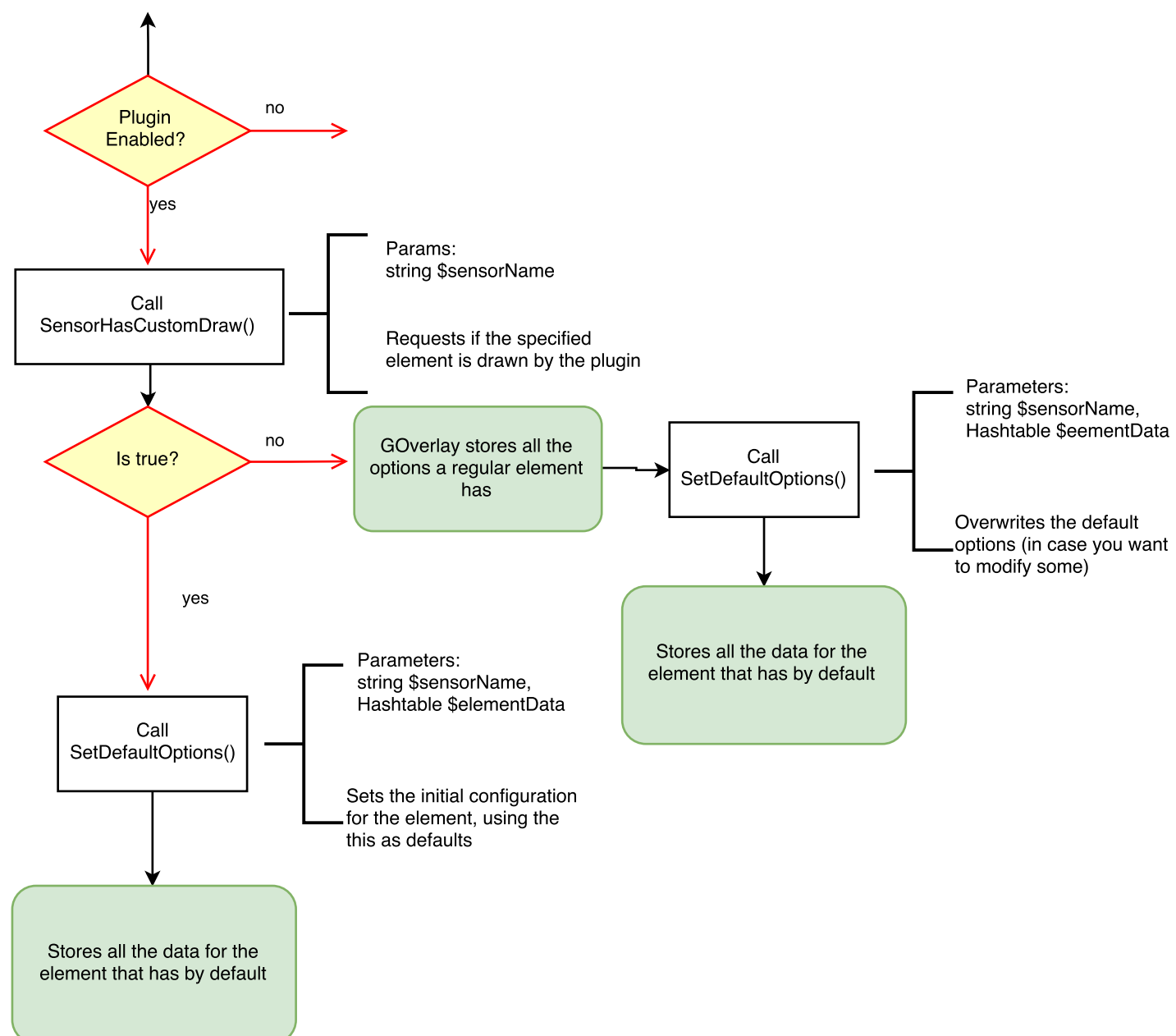
If the user added it as part of another element (plot graph, bar graph, slider, etc):

This element will have the specific options for those type of elements

IMPORTANT: A only custom draw element is not permitted to be used as part of other elements (plot graph, bar graph, etc) - Only those elements who are not custom drawn can be added to this elements.

Flow

Adding a plugin element to a profile



Example code

```
Public Function SetDefaultOptions(sensorId As String, elementData As Hashtable) As Hashtable Implements GOverlayPlugin.Interfaces.IPlugin.SetDefaultOptions
```

```
    If sensorId = pluginId & "temp" Then
        'For temp, lets modify the goverlay defaults
        elementData("width") = 100
        elementData("height") = 41
        elementData("maxnum") = 5
```

```
    End If
    Return elementData
```

```
End Function
```

When the user clicks on your element added to the display

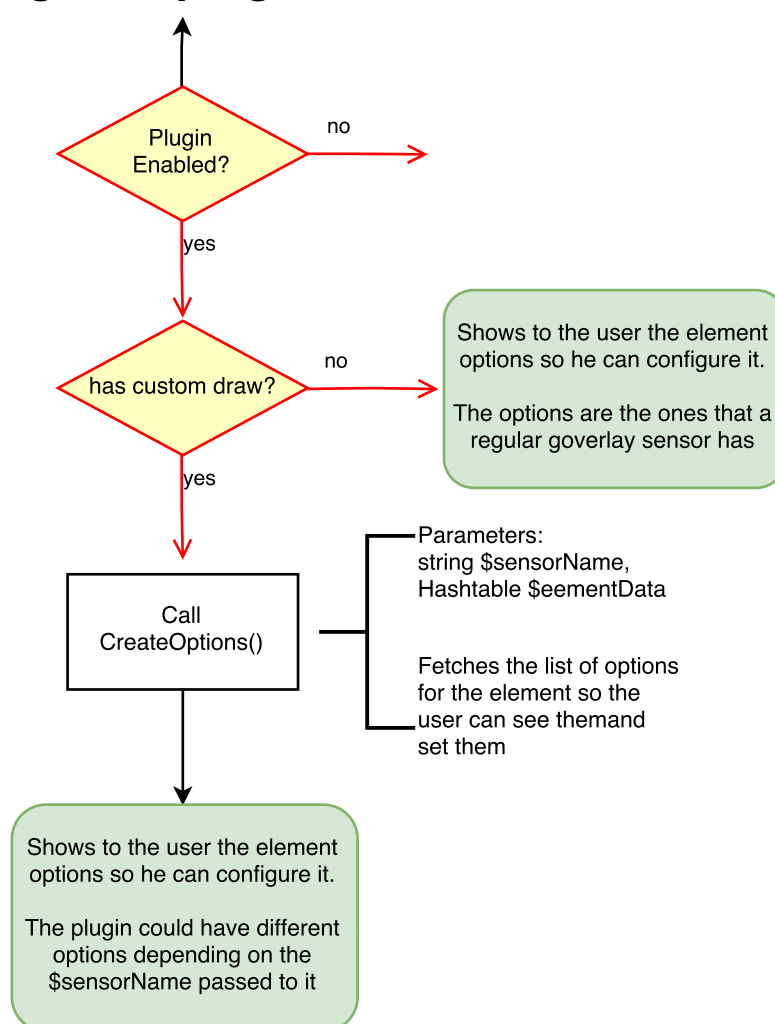
This flow happens when a user clicks on your element (which is now in the display) so he can configure it

Expected result

GOverlay will add the element options depending on if its a custom draw (in which case, will load ONLY the options your provided) or a regular element (in which case, it will add the regular options like font type, color, threshold, etc)

Flow

Clicking on a plugin element



Example code

```
Public Function CreateOptions(sensor_name As String, elementData As Hashtable) As Hashtable Implements GOverlayPlugin.Interfaces.IPlugin.CreateOptions
```

```
Dim options As New Hashtable
```

```
If (sensor_name = pluginId & "temp" Or sensor_name = pluginId & "usage") Then
```

```
ElseIf sensor_name = pluginId & "line" Then
```

```
options.Add(0, Helper.addPluginOption(pluginId & "lineDirection", "Line Direction", "lineDirection"))
```

```
options.Add(1, Helper.addPluginOption("Numerical11300", "Line Width", "lineWidth"))
```

```
ElseIf sensor_name = pluginId & "square" Then
```

```
options.Add(0, Helper.addPluginOption("ColorRGB", "The Color", "squareColor"))
```

```
options.Add(1, Helper.addPluginOption("Numerical11480", "Width", "width"))
```

```
options.Add(2, Helper.addPluginOption("Numerical11320", "height", "height"))
```

```
ElseIf sensor_name = pluginId & "text" Then
```

```
options.Add(0, Helper.addPluginOption("Text", "A text", "text"))
```

```
options.Add(1, Helper.addPluginOption("Numerical11480", "Width", "width"))
```

```
options.Add(2, Helper.addPluginOption("Numerical11320", "height", "height"))
```

```
End If
```

```
Return options
```

```
End Function
```

Performing the draw on the device

This flow happens every time a draw is sent to the device. This loop has the frequency that the user selected as "refresh rate" for the display, meaning that it could be called every 100ms to 10 seconds depending on his configuration.

Expected result

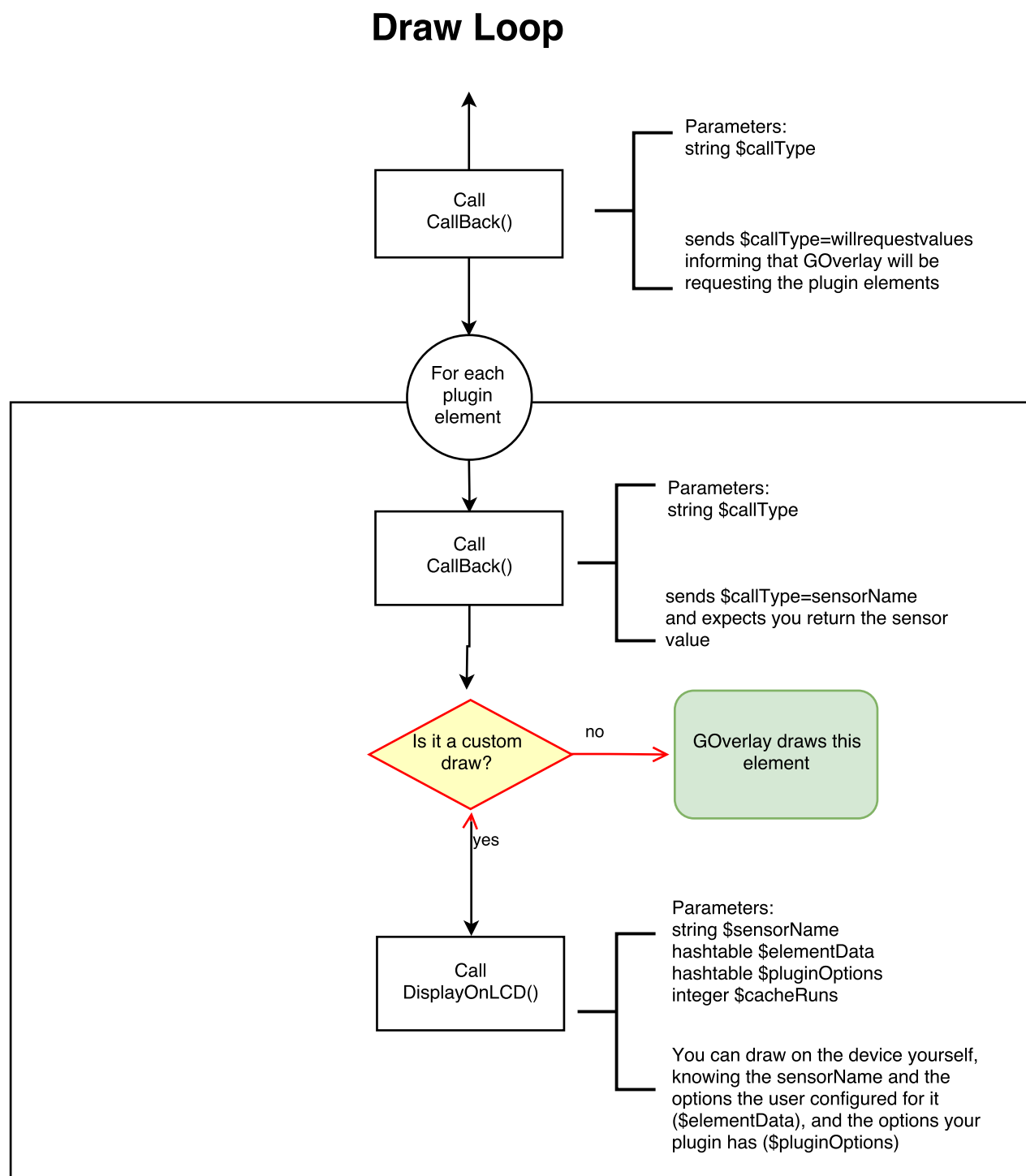
The draw loop, in every loop will first let your plugin know that its happening by sending a callback signal, after that, it will start requesting all the elements used by your plugin on the profile.

This is useful to update all your sensor information on the first signal and then just show those values from cache.

If its a custom draw, GOverlay will NOT output anything to the display, leaving you to do the drawing itself using the available commands.

If its NOT a custom draw, GOverlay will just want the VALUE of the sensor, so it can perform the draw accordingly for a plot graph, bar graph, or a simple value draw.

Flow



Example code

```
Function CallBack(method As String) As Hashtable Implements GOverlayPlugin.Interfaces.IPlugin.Callbacks
```

```
Dim returnHT As Hashtable = New Hashtable
```

```
Try
```

```
Dim value As Integer = 0
```

```
If method = "willrequestvalues" Then
```

```
    'Comes here once per run (only if sensors are used thru here)
```

```
Else
```

```
    value = InternalGatherSensorsValues(method)
```

```

End If

returnHT("value") = value

Catch ex As Exception
objHost.DebugMessage("Error grabbing action : " & method)
End Try

Return returnHT
End Function

```

Example code

```

Public Function LCDSys2_DisplayOnLCD(sensor_name As String, elementData As Hashtable, pluginOptions As Hashtable, cacheRuns As Integer) As ArrayList Implements GOverlayPlugin.Interfaces.IPlugin.LCDSys2_DisplayOnLCD

```

```

Dim MYpluginOptions As Hashtable = pluginOptions(pluginName)

```

```

If sensor_name = pluginId & "line" Then

```

```

Dim aLine As New ArrayList

```

```

aLine.Add(Helper.lcd2_addLine(elementData("x"), elementData("y"), MYpluginOptions("lineColor")))

```

```

Dim endX As Integer = elementData("x")

```

```

Dim endY As Integer = elementData("y")

```

```

If elementData("lineDirection") = "left" Then

```

```

endX = CInt(elementData("x")) - CInt(elementData("width"))

```

```

Elseif elementData("lineDirection") = "right" Then

```

```

endX = CInt(elementData("x")) + CInt(elementData("width"))

```

```

Elseif elementData("lineDirection") = "top" Then

```

```

endY = CInt(elementData("y")) - CInt(elementData("height"))

```

```

Elseif elementData("lineDirection") = "bottom" Then

```

```

endY = CInt(elementData("y")) + CInt(elementData("height"))

```

```

End If

```

```

If endX < 0 Then

```

```

endX = 0

```

```

End If

```

```

If endX > 480 Then

```

```

endX = 480

```

```

End If

```

```

If endY > 320 Then

```

```

endY = 320

```

```

End If

```

```

If endY < 0 Then

```

```

endY = 0

```

```

End If

```

```

aLine.Add(Helper.lcd2_addLine(endX, endY, MYpluginOptions("lineColor")))

```

```

objHost.DebugMessage("DemoPlugin - Drawing line from " & elementData("x") & ";" & elementData("y") & " to " & endX & ";" & endY & " -- elementW: " & elementData("width") & " --elementH: " & elementData("height") & " and color: " & MYpluginOptions("lineColor") & " and direction: " & elementData("lineDirection"))

```

```

objHost.LCDSys2_Draw_Lines(aLine, True, -1)

```

```

Elseif sensor_name = pluginId & "square" Then

```

```

objHost.LCDSys2_Draw_Rectangle(elementData("x"), elementData("y"), CInt(elementData("x")) + CInt(elementData("width")),

```

```

CInt(elementData("y")) + CInt(elementData("height")), elementData("squareColor"), 1, 0)

```

```

Elseif sensor_name = pluginId & "text" Then

```

```

objHost.LCDSys2_Draw_Text_Font(elementData("x"), elementData("y"), elementData("text"), elementData("width"), 65535, 0, "Arial_20px.bin")

```

```

End If

```

```

Return New ArrayList()

```

```

End Function

```