ALLAN THRAEN | 🕐 17 years ago | 📄 PDF | 💬

WHEN AND WHERE TO ATTACH DATAFACTORY EVENT HANDLERS

This is a classic problem with a lot of solutions: You're building a fancy plug-in for your favorite CMS (EPiServer of course), trying to fulfill your lifelong ambition to become an EPiServer MVP, when you suddenly realize that you your plugin to hook into some of the built-in events - for instance DataFactory events like "PagePublished". Of course, it's no sweat to make an event-handler - heck, you probably wrote it already - but where should that piece of code go? Naturally you'd like to attach to the event-handlers every time the httpApplication is started. Typically you learn that this is the kinda thing the "global.asax" is meant for. And it is. The problem is just that it doesn't really give your module that special "plug-and-play-module"-feeling, if you have to require the users of the module to edit their global.asax and perhaps even recompile in order for the module to work correctly.

Having faced this problem myself several times I've seen (and used) several solutions:

● Make an httpModule that in it's "void Init(HttpApplication context)" set's up the event-handlers. Either make a web.config merge file or ask the person installing the module to register the module in web.config.
● In EPiServer projects, Global.asax typically inherit from EPiServer.Global, so another approach is to make a new class that also inherits from EPiServer.Global and attaches the event handlers. During installation you'd then just have to change your global.asax to inherit from the global-class in the module instead of EPiServer.Global.
● Make a singleton class, that is stored in the HttpApplication context and upon a lazy load attaches the event-handlers. Then make a PagePlugin, that calls the singleton whenever a page is loaded, thereby forcing it to initialize. Not pretty, potentially slow, but seems to work without any necessary client-modifications.

Finally, my new favorite solution, that was suggested to me just 3 days ago by Magnus Stråle:

The automatic Plugin-loading mechanism in EPiServer will automatically try to call a static method with the signature "void Start()" on any attributes that inherit from EPiServer.Plugin.PluginAttribute compiled in an assembly in the bin folder.

This means that you can attach event-handlers like this:

```
public class CustomPluginAtt
 : EPiServer.PlugIn.PlugInAttribute
{
    public static void Start()
    {
        //Attach to the right events
        DataFactory.Instance.PublishedPage +=
       new PageEventHandler(Instance_PublishedPage);
    }

    static void Instance_PublishedPage(object sender, PageEventArgs e)
    {
        //Do stuff
    }
}
```