

Plug In Design for .NET App

Asked 16 years, 3 months ago Modified 12 years, 4 months ago

Viewed 683 times



5



I'm looking at rewriting a portion of our application in C# (currently legacy VB6 code). The module I am starting with is responsible for importing data from a variety of systems into our database. About 5-6 times a year, a new client asks us to write a new import for the system that they use. Presently, this requires us to release a new version of our software for each new import option we add to the application.

One of the goals of the rewrite is to make the application support plug-ins. Every new import can become a separate assembly which the host application will recognize and allow the end user to interact with. This will hopefully simplify life to some degree as we can simply drop a new assembly into the directory and have it be recognized and used by the main (host) application.

One of the items I am struggling with relates to the differences between the import options we currently support. In some cases we actually let the user point to a directory and read all of the files within the directory into our system. In other cases we allow them to point to a single file and import its contents. Additionally, some imports have a date range restriction that the user applies while others do not.

My question is, how can I design the application in a manner that allows for some flexibility among the imports we build and support while at the same time implementing a common interface that will allow the host application to easily recognize the plug-ins and the options that each one exposes to the user?

.net

plugins

interface-design

Share Follow

edited Jul 27, 2012 at 7:46



ThinkingStiff

65.3k ● 30 ● 147 ● 241

asked Sep 5, 2008 at 16:53



Tim Lentine

7,852 ● 5 ● 36 ● 40

3 Answers

Sorted by:

Highest score (default)



3

I would recommend you take a look at the Managed Add-In Framework that shipped with .NET 3.5. The [Add-In team](#) has posted some samples and tools at [CodePlex site](#) as well..



Share Follow

answered Sep 5, 2008 at 16:59



David Mohundro

12.4k ● 5 ● 42 ● 45





1

.Net 3.5 has the system.Addin namespace.

This thread also has some good information for older versions of the framework:



<http://forums.devshed.com/net-development-87/system-plugin-532149.html>



Share Follow

answered Sep 5, 2008 at 17:00



Joel Coehoorn

415k ● 114 ● 577 ● 813



1

for the theory take a look at the [plugin pattern](#) in martin fowlers Patterns of Enterprise Application Architecture

for an interesting example take a look at this tutorial:



[Plugin Architecture using C#](#)



Share Follow

answered Sep 5, 2008 at 17:02



Vitor Silva

17.6k ● 8 ● 34 ● 27