**How DXCore plug-ins are loaded**

August 19th, 2010

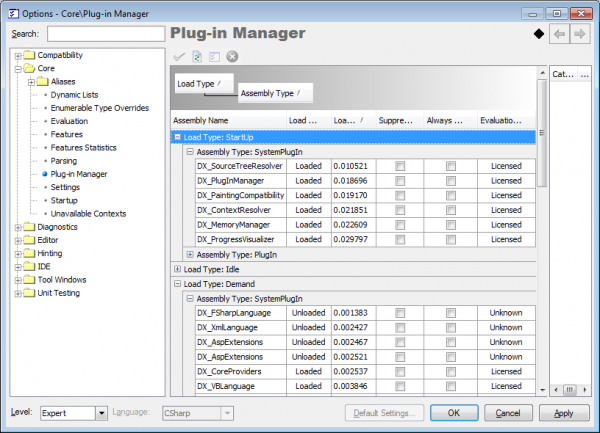
[DXCore](http://skorkin.com/2010/06/devexpress-dxcore-framework/) has a built-in **Loader Engine**, which is intended to improve the speed of its start-up process. However, the first **DXCore** start-up process is pretty long (it may take a minute or two). There’s a reason that the first start-up takes much more time then the subsequent launches. When **DXCore** loads for the first time, it has to load all of the plug-ins found in both, “**System**” and “**PlugIns**” subfolders. At this time, **DXCore** profiles every loaded assembly: checks its load type, loading time, so in brief, all important and necessary information that **DXCore** should know about a particular assembly. All this information is saved to Loader profiles in an **XML** format. This information will be used on every subsequent **DXCore** launch, so that Visual Studio should start instantly, because, in the first place, plug-ins are loaded from profiles only after the splash screen goes away and, secondly, because Loader Engine has optimized **DXCore** loading process specially for your system.

Plug-in profiles are stored in the “**Loader”** subfolder of the [Settings folder](http://skorkin.com/2010/08/where-ide-tools-coderush-andor-refactor-settings-are-stored/):

***%AppData%\CodeRush for VS .NET\1.1\Settings.xml\Loader***

If any of the plug-ins have changed on disk for some reason (recompiled and/or replaced) or a new plug-in is installed, the **Loader Engine** will notice that and correct its profiles for these specific plug-ins only.

You can see the profiles information on the “**Plug-in Manager**” options page in the [Options Dialog](http://skorkin.com/2010/08/ide-tools-options-dialog/):

[](http://skorkin.com/files/2010/08/IDEToolsPlugInManagerOptionsPage.png)

Follow these steps to get to the “**Plug-in Manager**” options page:

1. From the [DevExpress menu](http://skorkin.com/2010/08/how-to-show-the-devexpress-menu-if-it-is-not-visible/), select “Options…”.

2. In the tree view on the left, navigate to this folder:

**Core**

3. Select the “**Plug-in Manager**” options page.

**NOTE**: *This page level is Expert, and will only be visible if the Level combo on the lower-left of the Options dialog is set to Expert.*

If you have any trouble with loading of some plug-ins or **DXCore** itself, you may try to rebuild the **Loader** profiles. To do this, you need to remove the “**Loader”** folder with profiles from your system.

**NOTE**: if you are actively developing your own plug-in, it is recommended to change its “loading” setting to load it manually in the **Plug-In Manager**, to prevent it from being loaded automatically. This will allow you to compile and build your plug-in without errors due to file locks on the plug-in assembly. This option can be changed when you [create a new DXCore plug-in](http://skorkin.com/2010/08/how-to-create-a-new-dxcore-plug-in/). If you already have created a plug-in or modify the existing one, you may tweak the “***DXCoreAssembly***” attribute in the “***AssemblyInfo***” file of your plug-in solution to load it manually like this:

[assembly: DXCoreAssembly(DXCoreAssemblyType.PlugIn, “Assembly Title”, PlugInLoadType.Idle, LoadAbilityType.LoadDisabled)]

Change the **LoadAbilityType** parameter to **LoadDisabled** to make a plug-in loaded manually.