

Where Should You Store Your SharePoint Solution's Configuration Data?



Dr. Z 27 Mar 2012 11:37 AM

6

This is a common scenario: you are designing a great SharePoint solution and you have some configuration information, such as URLs, connection strings, list names, and so on, that you have to deal with. What should you do?

The easiest and dirtiest way is hard-coding them in code. It maybe fine for a small and quick proof-of-concept but of cause you should not do it.

Next place is application config file. In the case of SharePoint solutions, it is the web.config. Configuration data stored in web.config is not hard-coded, easy to change (who doesn't like NotePad?), and very easy to access from code.

There are other places to store configuration data, SharePoint lists are also popular. The problem is that you have to store the lists' sites and names somewhere else, otherwise you code would not know where to get the data. An external data store, such as SQL server, is also being used at some customers. Similarly, you have to deal with the data connection information separately as SharePoint lists.

However, in this blog I am going to recommend *NOT* to use any of the approaches mentioned above. To make my point, please check the comparison sheet below:

Place	Location	Data Management Tools	Backup	Pros	Cons	Should you do this
Code	GAC	Visual Studio	Server	Easy	Dirty	HELL NO!
web.config	File system	Notepad and so on...	Server	Easy	Multiple copies, hard to sync. Requires to restart app pool when changed. Not being backed up by SQL	No
SharePoint List	Content database	SharePoint web UI	Content db	Built-in UI, easy to manage	Have to store location data somewhere else. Too easy to manipulate	Maybe
External Data Source	External data source	Custom	Custom	You have full control	Have to store connection data somewhere else. Outside dependency.	Maybe not
Property Bag	Content database	Custom application page, utility	Content db	Backed up with content, on site and site collection levels	No built-in UI, have to create your own management tool	Yes
Hierarchical Object Store	Config database	Custom application page, utility	Config db	Backed up with config, on farm and web app levels	No built-in UI, have to create your own management tool	Yes

For storing configuration data for SharePoint solutions, I would like to recommend use either property bags or hierarchical object store, maybe together with SharePoint lists depends on the amount of data you are dealing with. Some people may argue that they do not have built-in UI, actually it could be a good thing in this case, because that means they are not likely to be altered by accident, and it is easier to control access.

Below are some useful links on property bags, hierarchical object store, and application pages:

- Property bag:

- <http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.utilities.sppropertybag.aspx>
- <http://pbs2010.codeplex.com/>
- Hierarchical object store:
 - <http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.administration.sppersistedobject.aspx>
 - <http://www.chaholl.com/archive/2011/01/30/the-skinny-on-sppersistedobject-and-the-hierarchical-object-store-in.aspx>
- Application page:
 - <http://msdn.microsoft.com/en-us/library/ee231581.aspx>

What about farm level SharePoint solutions? For example one of my customers developed a solution to support their enterprise search, some other customers developed timer jobs. Those solutions are not deployed for a specific site, site collection or a web application. Where should they store the configuration data?

The answer is to store the data in property bag in Central Admin or hierarchical object store, and deploy your application pages to CA. Here is an article about how to do it:

- Get CA's URL: <http://www.sytone.com/2010/07/getting-the-central-admin-url-for-a-farm-in-sharepoint/>
- Deploy an app page to CA: <http://msdn.microsoft.com/en-us/library/ff798467.aspx>

Zewei Song, Ph.D.

Microsoft Certified Master: SharePoint 2010

Microsoft Consulting Services

Comments



Bil Simser 2 Apr 2012 7:28 PM #

I don't agree that the Property Bag or Object Store is the best place for this. I would rate those as Maybe and always use a custom list. Why? The list has it's own UI you don't have to code for, it's simple to access the data and it's just as flexible as both of the other options (data is stored in the content databases so no special backup needed). Every developer knows how to read from a list. Not many know how to access the property bag or object store plus you have to build a custom UI for each of those.



Dr. Z 3 Apr 2012 10:32 AM #

That's a good argument Bill. But where do you store your list's url, name, and column names? A built-in UI sometime could be a problem...anybody with access can change the data, and people can accidentally remove it and make you application crash.



David Lozzi 3 Jan 2013 5:50 AM #

Awesome, thanks for sharing this. Always been a question of mine, and something I always leaned towards property bag or lists, depending on the need. Seeing persisted objects is interesting, looks easier than using the property bag;) Thanks again!



Scoutman 3 Jan 2013 8:51 AM #

Hi guys, last year i developed a feature to store dataconnections within SharePoint CA:

SharePoint 2010 Data Connection Manager

spugptwakeymanager.codeplex.com

is at Codeplex, hope it helps



Harold 27 Apr 2013 8:15 AM #

Well first, you are completely wrong about web.config. With SPWebConfigModification, everything is backed by SQL Server and changes are made instantly across the farm. Advocating editing by notepad is preposterous.



Dr. Z 8 May 2013 11:42 AM #

Harold, I don't think SPWebConfigModification class can store configuration changes to content or config databases (please correct me if I am wrong). It is just a convenient way to change web.config on

all WFEs (implemented by a timer job behind the scene). However, you must still write some code to call the class and implement the changes to web.config, usually by a console app or an application page. And since you are doing that anyway, why not use property bags or Hierarchical Object Store, because they are stored in databases. Also, remember whenever you change web.config you recycle the application pool, and that could be a BIG problem for production.

Zewei