Membership Providers

.NET Framework 4 5 out of 8 rated this helpful

By using a provider model, you can easily adapt the membership system to use different data stores or data stores with different schemas. In addition, you can extend the membership system by creating a custom provider, which you might do to create an interface between the membership system and an existing user database.

An important feature of the tiered membership model is that applications that use the membership system are not bound to a specific data store. If necessary, you can switch the underlying membership data store in your application without changing any application code by simply configuring the application to use a different membership provider. (And of course by migrating any existing user information, if applicable.) For example, if you already have a database containing user information, a provider can be written to take advantage of that data and the Membership APIs can then invoke that provider to perform membership tasks.

ASP.NET ships with two membership providers: one that uses Microsoft SQL Server as a data source and another that uses Windows Active Directory. Third parties might make membership providers available for other databases, such as Oracle, or for SQL Server databases that have a different schema than that used by ASP.NET providers.

You can also configure multiple membership providers, which allows you to select a membership provider at run time based on application requirements. For example, for business reasons your membership information might be in separate regional databases. By configuring multiple membership providers that each interact with a different regional database, you can direct membership calls to the appropriate provider for different users.

Custom Membership Providers

It is also possible to create a custom membership provider, which you might do to link the membership system to a user database that already exists, connect to a database not supported by the providers that are included with ASP.NET, or extend membership to expose more features.

Once you have a custom membership provider, you can configure your application to use that provider in the same way that you configure the application to use an ASP.NET provider. The Membership class will automatically invoke your custom provider to communicate with your authentication data source.

For details on creating a custom membership provider, see Implementing a Membership Provider. Also see the Sample Membership Provider Implementation.

See Also

Other Resources

Managing Users by Using Membership

Community Additions