

Re: Business layer wrapping user information and transactions?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2007-01/msg02117.html>

- *From:* "sloan" <sloan@xxxxxxxxx>
 - *Date:* Thu, 18 Jan 2007 09:03:53 -0500
-

<<What he said>>

I'd strongly recommend (with emphasis on the strongly) not trying to keep "sync" data.

Using the Provider Model, you can create your ~own
MyCustomMembershipProvider : MembershipProvider

This will allow you to use your custom tables, and such.

There's no reason why you can't use your MyCustomMembershipProvider from a Winforms.

You'll just lose the "drag and drop" ability that you have in asp.net 2.0.

See

<http://www.15seconds.com/issue/050216.htm>

"Peter Bradley" <pbradley@xxxxxxxxx> wrote in message
<news:OO0UonvOHHA.4172@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

One of the golden rules of software development is never, ever to hold the same data in more than one place. It's a maintenance nightmare.

Is there no way you can get both front ends to use the same data?

Petere

"Mark Baldwin" <sWozzi3@xxxxxxxxxxxxxxxxxxxx> wrote in message
[news:uHC\\$givOHHA.324@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](news:uHC$givOHHA.324@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

I have an application with both Windows Forms and ASP.NET front ends and uses both integrated web security and a security database. The problem I have is that I need to synchronize these security databases.

Re: Business layer wrapping user information and transactions?

The web front end uses ASP membership with the SQL provider, the
database

is moved to the SQL Server instead of the SQL Express datafile.

The Windows Forms app uses tables such as "users" within the database.

If I add a user on the website, I need also to add a user to the
database

"users" table.

To achieve this I write a business layer component that wraps both the security data tables and the ASP.NET membership functions. So for instance, to delete a user...

```
[DataObjectMethod(DataObjectMethodType.Delete, true)]
static public void Delete(string UserName)
{
    // delete user from internal database table "users"
    usersTableAdapter UserTableAdapter = new usersTableAdapter();
    int rowsAffected = UserTableAdapter.DeleteByName(UserName);

    // delete user from ASP.NET membership database
    Membership.DeleteUser(UserName, true);
}
```

The question is how do I get these two functions into a transaction so that if the deletion for the ASP.NET member fails, the delete on the internal database table is rolled back?

--
Best regards
Mark