

## Getting reusable API to "discover" the DB connection it needs at runtime

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.adonet/2006-09/msg00519.html>

---

- *From:* "jdonnici" <[jdonnici@xxxxxxxxxxxxxx](mailto:jdonnici@xxxxxxxxxxxxxx)>
  - *Date:* Fri, 22 Sep 2006 11:11:00 -0600
- 

We're working on an application that has a 'common' project that hosts a variety of 'general desktop app' APIs – user preferences, the non–UI code the app's services uses, etc. As part of that logic, it makes use of another new project which hosts the 'licensing' system. This licensing project contains an ORM wrapper around the licensing database, as well as the licensing/permissions APIs and logic.

In this app, the database connections will come from the 'common' project (because they're specified by the user's preferences). However, the licensing project needs that connection also so that the its implementations of the licensing APIs can access the current licensing database. We don't want the licensing project to deal with WHERE it goes for a connection because other apps down the road will use this same licensing library. For example, this app uses a preference setting while a web app would specify these in a config file.

So the question is how to design it so that the licensing system gets the connection details it needs at runtime using logic that will work down the road for other types of apps.

We've considered having the licensing system provide an `IConnectionProvider` interface. Any app that's going to use the licensing system would provide an implementation of that.

```
class DesktopConnectionProvider : IConnectionProvider {  
    public SqlConnection GetConnection() { ... }  
}
```

Then the code in the licensing system would need some place to go to get a connection

```
public class ConnectionManager {  
    public static IConnectionProvider ConnectionProvider;  
}
```

At startup, the app would "initialize" or "register" its connection provider implementation with the licensing system.

## Getting reusable API to "discover" the DB connection it needs at runtime

```
Licensing.ConnectionManager.ConnectionProvider = new  
DesktopConnectionProvider();
```

.... I know this would get the job done and work just fine. However, I'm not sure I care for the need to initialize or register something explicitly at startup. I'm wondering if there's a more elegant approach that would give the licensing system more "runtime discoverability" of the connection details it will need.

I'm curious what thoughts or ideas others might have. Thanks.

.