

Contents

- [Client application](#)
 - [Files and assemblies](#)
 - [The config file](#)
 - [Connection information](#)
 - [Logging](#)
 - [ObjectServer](#)
- [Packaging for deployment](#)
- [The data tier](#)
- [Security](#)

In a DevForce application, a **two-tier deployment** is one in which the [client tier](#) communicates directly with the [data tier](#), without the intervention of an [application server tier](#).

This is the simplest of the deployment options, since there are fewer "moving parts" to worry about. Generally your data tier is on one or more separate physical machines, but it need not be.

Client application

Without an application server, the responsibilities of the service tier are reduced, and packaged with the client application. The [EntityManager](#) will still communicate with an [EntityServer](#) to authenticate users and query and save entities, but the [EntityServer](#) will be deployed locally, not as a service but as part of the [AppDomain](#) in which the application runs.

Your [client tier](#) might be a WinForms, WPF or console application, or a Windows Service. An ASP.NET application which does not require that the [EntityServer](#) be exposed as a service can also be considered a two-tier client application.

Files and assemblies

The following assemblies are always required:

```
IdeaBlade.Core.dll
IdeaBlade.EntityModel.dll
IdeaBlade.EntityModel.Edm.dll
IdeaBlade.EntityModel.Server.dll
IdeaBlade.Linq.dll
IdeaBlade.Validation.dll
```

You can use ASP.NET security in a two-tier application. If so, you'll also need:
IdeaBlade.EntityModel.Web.dll

You'll usually need your .config file too:
app.exe.config (or web.config)

If you're deploying a [Code First](#) model, you'll also need the following:

```
EntityFramework.dll
IdeaBlade.Aop.dll
PostSharp.dll
```

You'll of course also need all of your own assemblies and other files needed by your application.

You may place the DevForce assemblies in the [GAC](#), although there's generally no compelling reason to do so, unless you have a large number of DevForce applications installed and they are all using the same DevForce version.

You do not need to, and in most cases should not, install DevForce to the target machine. Runtime license information is found in your model assembly and not the registry.

The config file

A typical configuration file will contain `<connectionStrings>` and `<ideablade.configuration>` sections, although neither is strictly required.

No part of the `<ideablade.configuration>` section is required, and can be removed altogether if you are not using it and can rely on DevForce defaults.

Connection information

You'll usually need the `<connectionStrings>` for any databases your application uses. (You can also use [<edmKeys>](#) instead of or in addition to `connectionStrings`, but they may be deprecated in the future.)

Remember the credential information in the connection string: if you must include a userid and password you may want to think about encrypting the connection strings section. Using Windows integrated security is often a better approach.

If you are using a custom [DataSourceKeyResolver](#) to dynamically provide connection information, you will not need to define connection information in the config file.

Logging

You can use the `<logging>` element to set the file name and location of the debug log, or to turn logging off altogether. You also might want to archive log files. Generally other logging attributes are not needed, or used only for debugging purposes.

A typical config file might contain the following:

```
<logging logFile="log\DebugLog.xml" archiveLogs="true" />
```

... to place the file in a sub-folder named log and archive files automatically.

Also see the [Log](#) topic for more information.

ObjectServer

You will not of course be communicating with a remote *EntityServer*, and frequently the `<objectServer>` element is not needed at all. You may find it useful if you wish to modify the login-related settings for *allowAnonymousLogin* and *loginManagerRequired*. Remember that the config file is not secure, so you may instead want to set these in your application code via the [IdeaBladeConfig.Instance](#) when the application starts.

All other settings for the `<objectServer>` element and sub-elements are not pertinent to a two-tier deployment.

Packaging for deployment

The computers on which you deploy the application likely have the .NET Framework 4.5 already installed or you will install it as part of your deployment.

Generally client applications are installed via an [installer](#) or a [ClickOnce](#) deployment.

The data tier

We don't really have much to say about your database(s) and leave its configuration to you and your DBA. You should consider security, however, both in terms of connection information on the client, access accounts, and transmission security. DevForce imposes no special requirements on the data tier.

Security

Remember that a client application deployed to a user's workstation can be compromised. Its code can be disassembled, its configuration, log and other files peered at, and its communications with the database intercepted.

See the [Security](#) topic for more information on steps you can take to secure your client application.