



SPANNING JAVA & .NET

**Accessing EJBs from .NET  
using Oracle WebLogic® and JNBridgePro™**

**Version 11.0**

JNBridge, LLC  
jnbridge.com

COPYRIGHT © 2002–2019 JNBridge, LLC. All rights reserved.

JNBridge is a registered trademark and JNBridgePro and the JNBridge logo are trademarks of JNBridge, LLC.

Java is a registered trademark of Oracle and/or its affiliates. Microsoft, Windows, and Visual Studio are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries. BEA and WebLogic are registered trademarks of Oracle Corporation. All other marks are the property of their respective owners.

October 7, 2020



## Accessing EJBs from .NET using WebLogic and JNBridgePro

The following example shows how to access Enterprise Java Beans from C# using Visual Studio and the JNBridgePro Visual Studio proxy generation plug-in. The EJBs will be running on Oracle WebLogic 10 or later. We assume that JNBridgePro 6.0 or later, and .NET framework 2.0 (3.0/3.5) or .NET 4.0 or later, have been installed on a Windows machine and that you are familiar with the use of Visual Studio 2005/2008/2010/2012/2013/2015/2017/2019 and JNBridgePro and have read the documentation. We also assume that WebLogic 10, including the examples server, has been installed on a compatible machine, which may be the same machine as the .NET machine, or a different machine reachable on the network. In addition, the .NET machine must have a suitable JRE installed (if the .NET machine is also hosting WebLogic 10, then the private JDK can be used).

If using VS 2010 or later, the directory, `..\WebLogic10\WebLogic10Example_Net40\`, contains a VS 2010 project already targeted for .NET 4.0. (The project can also be consumed by VS 2012, 2013, 2015, 2017, and 2019.) The directory, `..\WebLogic10\WebLogic10Example\`, contains a VS 2005 project targeting .NET 2.0. The VS 2005 project can, of course, be upgraded to later versions of Visual Studio.

## Deploying the Example EJB

This example uses a managed EJB already installed on the WebLogic 10 examples server. The source and build directories for this example bean can be found, on Windows, at `[WL10InstallDir]\samples\server\examples\src\examples\ejb\ejb20\basic\beanManaged` and `[WL10InstallDir]\samples\server\examples\build\ejb20BeanMgedEardemo`, respectively.

The Java archive file for the managed bean is `ejb20_basic_beanManaged_client.jar` and can be found in the root directory for this example. Please make sure that the WebLogic 10 examples server is installed and started. Make sure the example managed bean has been deployed by viewing the JNDI Tree for the examples server and looking for “`ejb20-beanManaged-AccountHome`”.

## Building the Proxy DLL

Before opening Visual Studio, it may be wise to copy the example directory, `WebLogic10`, to another location to avoid access privilege problems. Also be aware that some files may require setting read/write access.

Build the proxy assembly DLL as follows:

1. Start up Visual Studio and load the solution, `WebLogic10\WebLogic10Example\WebLogic10Example.sln`. Select the menu item `File→Add →New Project...` and select project type JNBridge. Type in any project name. After the JNBridgePro project has been added to the solution, click on the object `DotNetToJavaProxies.jnb` in the Solution Explorer to open the proxy tool interface.
2. Select the menu item `JNBridgePro→JNBridgePro Java Options...` and verify that the box `Start Java automatically` has been checked, and that the Java configuration data is correct. Close the dialog box. This dialog may automatically open if any configuration data is incomplete.
3. Select the menu item `JNBridgePro→Edit Classpath...` and add the following files to the classpath:



- **ejb20\_basic\_beanManaged\_client.jar** found in the root directory for this example, `..\J2EE-Examples\WebLogic10`. This archive contains the Account and AccountHome classes used in the C# code.
- **wlclient.jar** and **wl-j2ee-client.jar** found in the WebLogic `lib` directory. These archives constitute the WebLogic stand-alone thin-client (it contains the classes EJBHome, EJBObject, InitialContext and other EJB and JNDI related support classes). These jar files can be copied to a location on the .NET machine.

Close the Edit Class Path dialog box

4. Select the menu item JNBridgePro→Add Classes from Classpath... and add the following classes. For each class added, make sure that the box Include Supporting Classes is checked.
  - `examples.ejb.ejb20.basic.beanManaged.Account`
  - `examples.ejb.ejb20.basic.beanManaged.AccountHome`
  - `examples.ejb.ejb20.basic.beanManaged.ProcessingErrorException`
  - `javax.naming.Context`
  - `javax.naming.InitialContext`
  - `javax.naming.NamingException`
  - `javax.rmi.PortableRemoteObject`
  - `java.util.Properties`
  - `javax.ejb.CreateException`
5. Click OK to add the classes and all supporting classes to the Environment pane. Check all the items in the environment (use the menu item JNBridgePro→Check All in Environment) and click on the Add button to move them all to the Exposed Proxies pane.
6. Select the menu item JNBridgePro→ Build to build the proxy assembly.

## Building and Running the Client Application

You've now completed the task of building an assembly containing proxies of the managed Account EJB. The next step is to reference the proxy assembly and JNBridgePro components from the .NET console application, `WebLogicExample`, and configure the .NET-to-Java bridge for run-time execution.

1. Using the Add Reference... dialog in Visual Studio, add the project that created the proxy assembly as a reference. Also, add a reference to `JNShare.dll`, found in the JNBridgePro install directory, e.g. `C:\Program Files\JNBridge\JNBridgePro v11.0\2.0-targeted\JNShare.dll`. If using VS 2010 or later and targeting .NET 4.0, use `C:\Program Files\JNBridge\JNBridgePro v11.0\4.0-targeted\JNShare.dll`.
2. Open the source file, `Program.cs`, and provide the values to connect to the WebLogic examples server for the string variables `hostName`, `portNumber`, `login` and `passWord`. If the WebLogic implementation does not need authentication credentials, then leave the credentials as empty strings.
3. Open the `App.config` XML file and uncomment the element `dotNetToJavaConfig`. Edit these attributes :



- **jvm** points to the Java Virtual Machine, `jvm.dll`. The .NET machine (where you're building this example) must have a JRE installed. If this machine is also running WebLogic, then the JDK client JVM can be used. A Java 6 JVM must be used.
  - **jnbcore** points to `jnbcore.jar`, found in the `jnbcore` directory where JNBridgePro is installed.
  - **bcel** points to `bcel-5.1-jnbridge.jar`, found in the `jnbcore` directory where JNBridgePro is installed.
  - **classpath** points to the three jar files used to create the proxy assembly:  
`ejb20_basic_beanManaged_client.jar`, `wlclient.jar` and `wl-j2ee-client.jar`.
4. Build the project. There should be no errors.
  5. Run the .NET application. Output showing the results of account queries will be displayed.

### Notes on this Example

- The default .NET-to-Java bridge configuration for this example is to use the shared memory transport. The `App.config` file can be modified to use `tcp/binary`. See the User's Guide for details.
- Instead of using the JNBridgePro Visual Studio plug-in, it is possible to use the JNBridgePro stand-alone proxy generation tool to create the proxy assembly. The resulting assembly can then be referenced from the .NET project in VS.
- On the .NET machine, you should use the same version of Java recommended for the version of WebLogic that you are using.