How to Use Native Code (C++) to Create a Simple FTP Authentication Provider

By **Robert McMurray**March 18, 2009

Microsoft has created a new FTP service that has been completely rewritten forWindows Server® 2008. This new FTP service incorporates many new features that enable Web authors to publish content more easily than before, and offers Web administrators more security and deployment options.

The new FTP 7.5 service supports extensibility that lets you extend the built-in functionality that is included with the FTP service. More specifically, FTP 7.5 supports the creation of your own authentication providers. You can also create providers for custom FTP logging and for determining the home directory information for your FTP users.

This walkthrough will lead you through the steps to use native code to create a simple FTP authentication provider.

PREREQUISITES

The following items are required to complete the procedures in this article:

1. 1. IIS 7.0 or above must be installed on your Windows Server 2008 server, and the Internet Information Services (IIS) Manager must also be installed.
2. The new FTP 7.5 service must be installed. You can download and install the FTP 7.5 service from the http://www.iis.net/ web site using one of the following links:
   - FTP 7.5 for IIS (x86)
   - FTP 7.5 for IIS (x64)
3. You must create a root folder for FTP publishing.
4. You must use Visual Studio 2008.
   - Note: If you use an earlier version of Visual Studio, some of the steps in this walkthrough may not be correct.
   - Note: If you plan to develop custom FTP providers for computers that use a 64-bit version of Windows, you will have to install the 64-bit tools and compilers for Visual Studio. You can find additional information about 64-bit development in the Installing Visual Studio 64-bit

Components topic on the Microsoft MSDN Web site.

Step 1: Set up the Project Environment

In this step, you will create a project in Visual Studio 2008 for the demo provider.

1. Open Microsoft Visual Studio 2008.
2. Click the File menu, then **New**, then **Project**.
3. In the New Project dialog box:
    - Choose Visual C++ as the project type.
    - Choose ATL Project as the template.
    - Type FtpAuthenticationDemo as the name of the project.
    - Click OK.
4. When the ATL Project Wizard dialog appears:
    - Click Next.
    - Ensure that only Dynamic-link library (DLL) is checked.
    - Click Finish.
5. When the project opens, add an ATL Class to the project:
    - Click Project, and then click Add Class.
    - Choose ATL Simple Object as the template.
    - Click Add.
6. When the ATL Simple Object Wizard appears:
    - Enter "FtpAuthDemo" for the short name and accept the defaults for the other values.
    - Click Next.
    - Choose the following options:

        - Choose Apartment for the Threading model. (Note: This can be customized depending on your application's needs.)
        - Choose No for Aggregation. (Note: This can be customized depending on your application's needs.)
        - Choose Custom for the Interface.
    - Click Finish.

7. Add the extensibility interfaces:
    - Click View, and then click Class View.
    - In the Class View window, expand FtpAuthenticationDemo.
    - Right-click CFtpAuthDemo, then click Add, then click Implement Interface.
    - Choose File for the interface implementation.
    - For the Location, enter the full path of the FTP extensibility type library. For example:

    ```
    C:\Windows\System32\inetsrv\ftpext.tlb
    ```

    Note: If you are developing on a 64-bit computer, you should copy the FTP extensibility type library to the following 32-bit path and use that location. For example:

    ```
    C:\Windows\SysWOW64\inetsrv\ftpext.tlb
    ```

    - Choose the following interfaces to implement:
        - IFtpAuthenticationProvider
        - IFtpRoleProvider
    - Click Finish.

8. Configure the project so that the DLL will not be registered automatically:
    - Click Project, and then click FtpAuthenticationDemo Properties.
    - Expand Configuration Properties, and then click Linker.
    - Select Register Output, and select No from the drop-down menu.

- Click OK.

9. Optional: If you are developing your custom provider on a 32-bit version of Windows, you can optionally add a custom build event to automatically deploy and register the DLL on your development computer. (Note: These steps will not work on a 64-bit version of Windows.) To add the custom build event, follow these steps:

- Click Project, and then click FtpAuthenticationDemo Properties.
- Expand Configuration Properties, then expand Build Events, then click Post-build Event.
- Click the ellipsis (...) on the right side of the Command line text box.
- Enter the following in the Command line dialog box:

```
net stop ftpsvc
copy /y "$(TargetPath)" "%WINDIR%\System32\inetsrv"
regsvr32.exe /s "%WINDIR%\System32\inetsrv\$(TargetFileName)"
net start ftpsvc
```

- Click OK to close the Command line dialog box.
- Click OK to close the FtpAuthenticationDemo Property Pages dialog box.

10. If you are developing a provider for a 64-bit version of Windows, you will need to add a build configuration for 64-bit complilation:

- Click Build, and then click Configuration Manger...
- Select <New...> in the Active Solution Platform drop-down menu.
- Select x64 in the Type or select the new platform drop-down menu.
- Click OK.
- Click Close.

11. Save the project.

Step 2: Implement the Extensibility Interfaces

In this step, you will implement the extensibility interfaces for the demo provider.

1. Implement IFtpAuthenticationProvider:

   - In the Class View window, double-click the AuthenticateUser method.
   - Replace the existing implementation with the following code:

```
// IFtpAuthenticationProvider Methods
public:
    STDMETHOD(AuthenticateUser)(LPWSTR pszSessionId,
        LPWSTR pszSiteName,
        LPWSTR pszUserName,
        LPWSTR pszPassword,
        LPWSTR * ppszCanonicalUserName,
        long * pfAuthenticated)
    {
        // Note: You would add your own custom logic here.
        *ppszCanonicalUserName = pszUserName;

        CString strUserName = L"MyUser";
        CString strPassword = L"MyPassword";

        // Verify that the user name and password are valid.
        // In this example, the user name is case-insensitive
        // and the password is case-sensitive.
        if ((strUserName.CompareNoCase(pszUserName)==0) &&
            (strPassword.Compare(pszPassword)==0))
        {
            *pfAuthenticated = TRUE;
        }
        else
        {
            *pfAuthenticated = FALSE;
        }
        return S_OK;
    }
```

2. Implement IFtpRoleProvider:

- In the Class View window, double-click the IsUserInRole method.
- Replace the existing implementation with the following code:

```
// IFtpRoleProvider Methods
public:
    STDMETHOD(IsUserInRole)(LPWSTR pszSessionId,
        LPWSTR pszSiteName,
        LPWSTR pszUserName,
        LPWSTR pszRole,
        long * pfIsInRole)
    {
        // Note: You would add your own custom logic here.
        CString strUserName = L"MyUser";
        CString strRoleName = L"MyRole";

        // Verify that the user name and role name are valid.
        // In this example, both the user name and
        // the role name are case-insensitive.
        if ((strUserName.CompareNoCase(pszUserName)==0) &&
            (strRoleName.CompareNoCase(pszRole)==0))
        {
            *pfIsInRole = TRUE;
        }
        else
        {
            *pfIsInRole = FALSE;
        }

        return S_OK;
    }
```

3. Add an include file reference for atlstr.h to the start of the FtpAuthDemo.h file:

```
#include <atlstr.h>
```

4. Save and compile the project.

Step 3: Add the Authentication Provider to FTP

In this step, you will add the demo provider to your FTP service and the default Web site.

1. Add the extensibility provider to the global list of FTP authentication providers:
    - Open the Internet Information Services (IIS) Manager.
    - Click your computer name in the Connections pane.
    - Double-click FTP Authentication in the main window.
    - Click Custom Providers... in the Actions pane.
    - Click Register.
    - Enter FtpAuthenticationDemo for the provider Name.
    - Click Native Provider (COM).
    - Enter the class name for the extensibility provider asFtpAuthenticationDemo.FtpAuthDemo.
    - Click OK.
    - Clear the FtpAuthenticationDemo check box in the providers list.
    - Click OK.
2. Add the custom authentication provider for an FTP site:
    - Open an FTP site in the Internet Information Services (IIS) Manager.
    - Double-click FTP Authentication in the main window.
    - Click Custom Providers... in the Actions pane.
    - Check FtpAuthenticationDemo in the providers list.
    - Click OK.
3. Add an authorization rule for the authentication provider:

- Double-click FTP Authorization Rules in the main window.
- Click Add Allow Rule... in the Actions pane.
- You can add either of the following authorization rules:
    - For a specific user:bye
        - Select Specified users for the access option.
        - Type "MyUser" for the user name.
    - For a role or group:
        - Select Specified roles or user groups for the access option.
        - Type "MyRole" for the role name.
- Select **Read** and **Write** for the **Permissions** option.
- Click OK.

Summary

In this walkthrough you learned how to:

- Create a project in Visual Studio 2008 for a custom FTP authentication provider.
- Implement the extensibility interface for custom FTP authentication.
- Add a custom authentication provider to your FTP service.

When users connect to your FTP site, the FTP service will attempt to authenticate users with your custom authentication provider. If this fails, the FTP service will use other built-in or authentication providers to authenticate users.