

SharePoint Learning Kit Course Manager Libraries Build Guide

Published July 2008

Contents

<i>Introduction.....</i>	<i>1</i>
Who should use this guide.....	1
System requirements.....	1
Install Ajax Control Toolkit.....	1
<i>Build the libraries.....</i>	<i>2</i>
Directory structure.....	2
DLL reference updates	3
Build sequence.....	4

Introduction

Use this guide to build the latest version or branch of the SharePoint Learning Kit (SLK) Course Manager libraries. They provide a multi-tier framework for business application development.

SLK is a SCORM 2004 certified e-learning delivery and tracking application built as a Windows® SharePoint® Services 3.0 solution. It works with either Windows® SharePoint® Services 3.0 or Microsoft® Office SharePoint® Server 2007.

SLK Course Manager is an add-on that provides additional features to help instructors with course planning, assignments, and grading. For information about how to install SLK Course Manager, see the SharePoint Learning Kit Course Manager Installation Guide.

Who should use this guide

Anyone building the libraries should use this guide. Often, those responsible for building libraries are developers.

System requirements

To build the libraries, you need the following:

- Windows Server® 2003, Windows Vista®, or Windows® XP operating system
- SLK Course Manager libraries source code
- [Microsoft® Visual Studio® 2008 Express Edition](#)
- [Microsoft® .NET Framework 3.5](#)
- [Ajax Control Toolkit](#) for .NET Framework 3.5
- Microsoft.Interop.Security.AzRoles in the [Windows Server 2003 Administration Tools Pack](#), which may already be installed. If you download the pack, include the latest updates.

Install Ajax Control Toolkit

To download and install the [Ajax Control Toolkit](#) from the CodePlex site, do the following:

1. On the **ASP.NET AJAX** page, in the **Files** section, double-click **AjaxControlToolkit-Framework3.5-NoSource.zip**.
2. Extract the contents and open the **AjaxControlExtender** folder.
3. Launch **AjaxControlExtender.vsi**, select all options, and then click the **Next** button.
Note If you receive a warning that the package has not been signed, you can ignore it.
4. After the application runs, click **Finish**.
5. Run **AjaxControlExtender.vsi** again.
6. In the **Visual Studio Content Installer** window, click **View files in Windows Explorer**, and then click **Cancel**.
7. In the **Windows Explorer** directory, open **AjaxControlExtenderProjectCS.zip**, and then extract **AjaxControlToolkit.dll** to the appropriate .NET assemblies folder. If you are not sure which assemblies folder to use, follow the steps below to add it to the Global Assemblies Cache (GAC).

To add the assembly using the gacutil.exe utility, do the following:

- a. In a Command Prompt window, type the directory path to the utility, which is typically located at:
`<drive>:\<Microsoft SDKs folder>\Windows\<version number>\bin`
For example, the utility might be located at:
`C:\Program Files\Microsoft SDKs\Windows\v6.0A\bin`
- b. Run the gacutil.exe utility with the /i parameter, and then type the directory where you extracted AjaxControlToolkit.dll.
For example, to run the utility, you might type the following:
`gacutil.exe /i C:\AjaxExtractedDir\AjaxControlToolkit.dll`

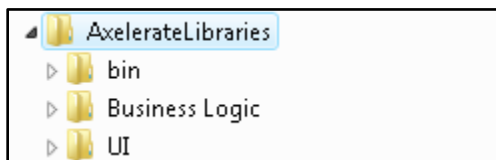
Build the libraries

Directory structure

The source code contains multiple directories, including:

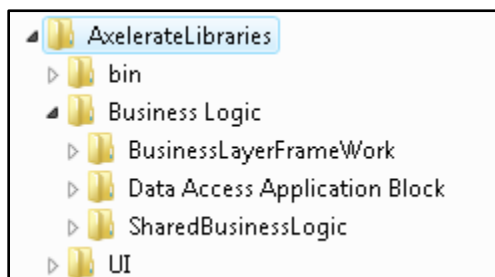
- AxelerateLibraries
- InstallerHelper
- SlkCourseManagerLogicalLayer

The libraries are in the AxelerateLibraries directory, which contains the Business Logic and UI folders.



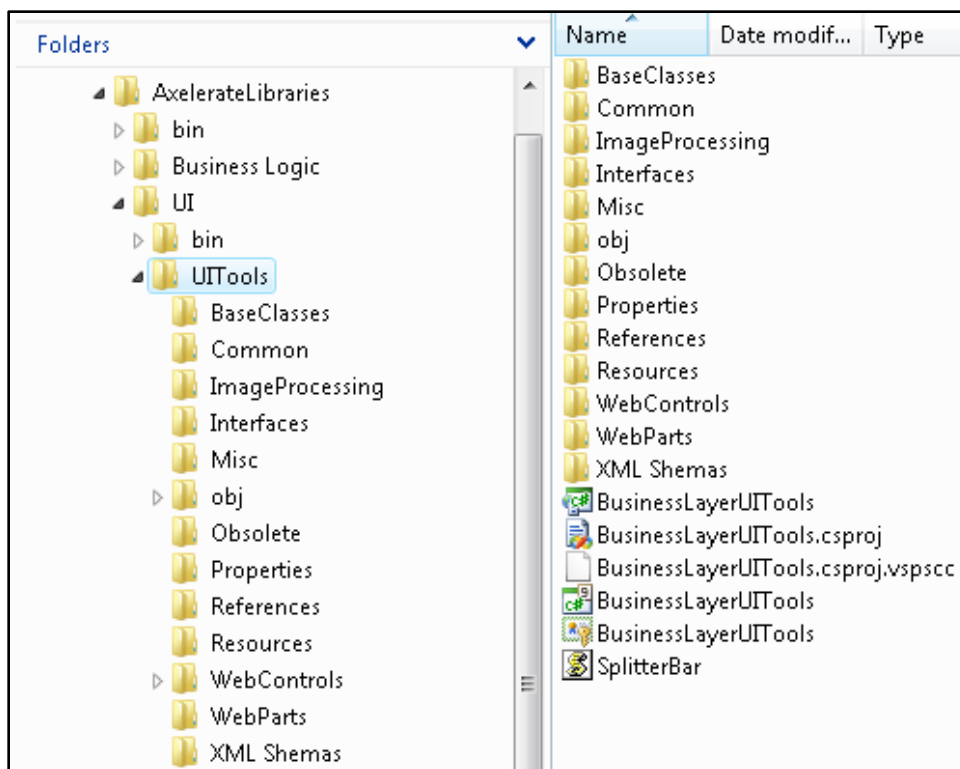
The BusinessLogic folder contains the following three folders and their corresponding projects:

- **BusinessLayerFrameWork** (VB) The business layer of the library, central to the library architecture.
- **Data Access Application Block** (VB) An abstraction and encapsulation for high performance and scalable best practices for common uses of the SQL Client.
- **SharedBusinessLogic** (C#) Interfaces and functionality for business logic.



Inside the UI folder is the UITools folder, which contains the following items:

- External tools
- BaseWebControls Web application
- BusinessLayerUITools (C#) project, the layer for UI tools based on business objects.

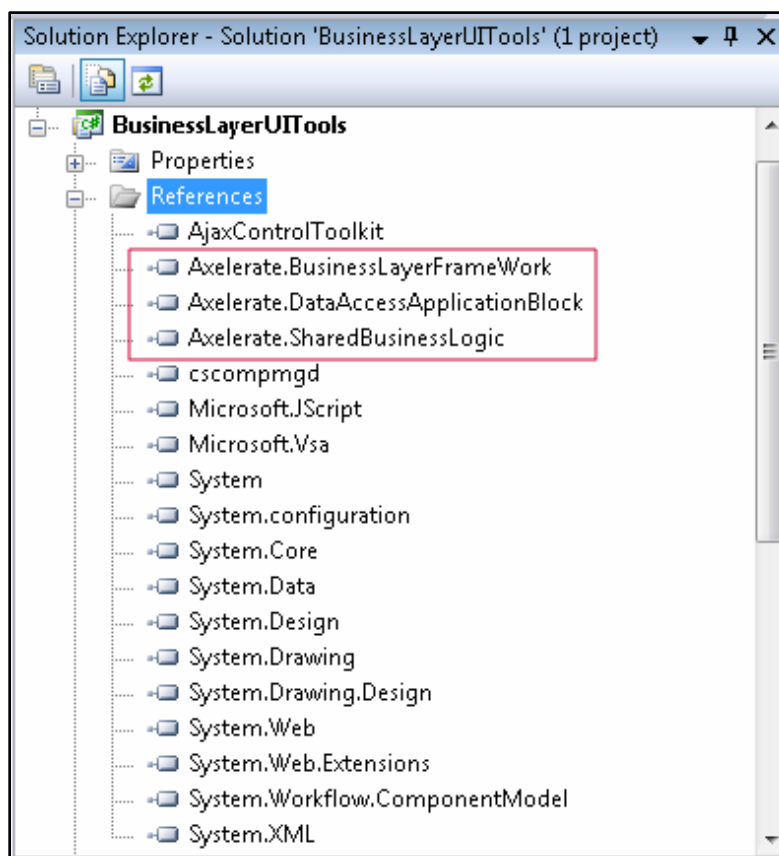


DLL reference updates

When you build the projects, they create Dynamic Link Libraries (DLLs). However, before you build the projects, you need to replace the references to the DLLs with the ones the projects require. The correct references are in the table below.

DLL	Location
Axelerate.DataAccessApplicationBlock.dll	\AxelerateLibraries\Business Logic\Data Access Application Block\bin\Debug\Axelerate.DataAccessApplicationBlock.dll
Axelerate.BusinessLayerFramework.dll	\AxelerateLibraries\Business Logic\BusinessLayerFrameWork\bin\Debug\Axelerate.BusinessLayerFramework.dll
Axelerate.SharedBusinessLogic.dll	\AxelerateLibraries\Business Logic\SharedBusinessLogic\bin\Release\Axelerate.SharedBusinessLogic.dll
Note Assemblies are not dropped to Debug here.	
Axelerate.BusinessLayerUITools.dll	\AxelerateLibraries\UI\UITools\obj\Debug\Axelerate.BusinessLayerUITools.dll

For example, in the BusinessLayerUITools project, replace the references to the DLLs with the paths specified in the table.



Note You can also add all three DLLs from `\AxelerateLibraries\Business Logic\SharedBusinessLogic\bin\Release`.

Build sequence

The projects are listed below in the sequence in which you should build them.

You can avoid build errors by including all dependencies and using correct path names. To assist in the build process, the list of projects includes solutions to common errors.

Project	Common Build Errors
Data Access Application Block	Minimal dependencies; no common errors.
BusinessLayerFrameWork	This project requires the Microsoft.Interop.Security.AzRules security library. This file is often missing in Windows XP installations. For more information, see the "System requirements" section of this guide.
SharedBusinessLogic	Minimal dependencies; no common errors.
BusinessLayerUITools	This project requires the AjaxControlToolkit library. For more information, see the "System Requirements" section of this guide.

To build each project, select **Build**, and then **Build Solution**.

Note If you attempt to build the project multiple times, select **Rebuild Solution** instead of Build Solution, if that option is available.