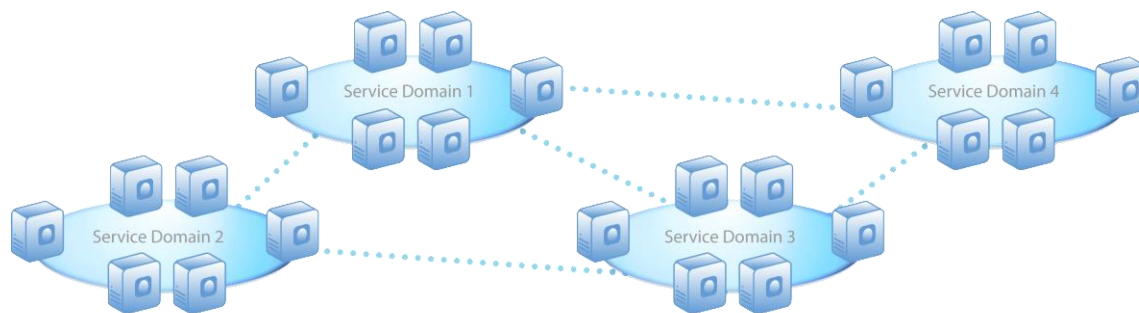


.NET StockTrader 5.0 Guide to Visual Studio Solutions

6/9/2011

© Microsoft Corporation 2011



THIS IS NOT A PRODUCT SPECIFICATION.

This document and related sample code supports Windows Server® 2008 R2, Windows Azure, SQL Azure, Azure AppFabric and the Microsoft .NET Framework 4.0 as a redistributable sample application kit.

The information contained in this document represents the current view of Microsoft Corp. on the issues disclosed as of the date of publication. Because Microsoft must respond to changing market conditions, this document should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Microsoft grants you the right to reproduce this guide, in whole or in part.

Microsoft may have patents, patent applications, trademarks, copyrights or other intellectual property rights covering subject matter in this document, except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights or other intellectual property.

© 2011 Microsoft Corp. All rights reserved.

Microsoft, Windows Server, Windows Azure, SQL Azure, SQL Server, the Windows logo, Windows, Active Directory, Windows Vista, Visual Studio, Internet Explorer, Windows Server System, Windows NT, Windows Mobile, Windows Media, Win32, WinFX, Windows PowerShell, Hyper-V, and MSDN are trademarks of the Microsoft group of companies.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

Overview	2
Visual Studio Development Environment Required	2
Recommended and Optional SDKs	2
Shared Libraries	2
Windows Azure and Selectable Host Environment	3
Windows Azure Projects	3
The Builds Folder.....	4
The Configuration Service Source and Solution.....	4
The Main .NET StockTrader Visual Studio Solutions.....	5
Supplementary .NET StockTrader Visual Studio Projects and Solutions	5
Configuration Service Template and Solution Wizard	6
Capacity Planner Solution	6

Overview

The .NET StockTrader 5.0 download has several Visual Studio projects and solutions for developers. This reference document provides a quick overview of these solutions, and their install locations. It also discusses the use of various shared libraries within the samples provided.

Visual Studio Development Environment Required

The sample was developed using Visual Studio 2010 Ultimate, however everything should work just fine in Pro editions of Visual Studio. You should be running Visual Studio 2010 SP1: **it is important to apply Visual Studio 2010 SP1 which will update both Visual Studio and the .NET Framework 4.0.** The VS 2010 SP1 update can be downloaded from MSDN.

Recommended and Optional SDKs

- Highly recommended: The latest Windows Azure SDK and Azure Tools for Visual Studio (free download on MSDN). This is required to open, build and publish the Azure projects.
- Recommended: The latest Windows 7.x SDK. Not required, but recommended for getting the latest WCF tools, such as svcutil.exe. Also necessary to generate your own self-signed certificates and create x.509 certificate .pfx files.
- Optional: Windows Server App Fabric SDK. Optional, but required on cache servers to setup a distribute cache cluster on Windows Server machines.
- Optional: Windows Azure App Fabric SDK. Optional, and in fact the caching client libraries are redistributed with the sample itself as shared libraries, so you do not need to install to use and develop against the Windows Azure cloud distributed cache. Do not install this 1.x SDK on the same machine with the Windows Server App Fabric 1.x SDK.

Shared Libraries

Many of the provided Visual Studio projects reference various shared libraries, which are contained in the various sub-folders in the \SharedLibraries folder. The provided Visual Studio projects are configured with post-build steps such that their compiled output will be properly copied into the appropriate subfolder in the base SharedLibraries folder, as appropriate. For example, the .NET StockTrader Business Service solution has projects that rely on shared schema information for the .NET StockTrader Order Processor Service (the service contract and data contract assemblies). Hence, these Order Processor projects (within the overall Order Processor Service solution) copy their output assemblies to the subfolder \SharedLibraries\OrderProcessorService. Here, they are referenced by the Business Service projects (contained in the overall Business Service solution) that require such references. The intent is to be as modular as possible, and setup an overall project structure that is cleanly partitioned into elements that might be developed separately by different groups or even different organizations.

Windows Azure and Selectable Host Environment

With StockTrader 5.0 and the new Visual Studio 2010 ConfigService 5 Template Wizard, all types of host environments are supported across on-premise and the Windows Azure cloud:

- Windows Azure
 - Web Role
 - Worker Role
 - VM Role (in this role, you would select an on-premise environment below but deploy the application to your VM Role in Windows Azure)
- On Premise
 - Web Application
 - Windows NT Service
 - Console Application
 - Windows Application (this environment includes a base-class Windows .NET Forms shell within which the services operate, and provides live viewing of debug and trace information)

It is important to note that all hosting environments utilize the same shared projects for all elements of the application: the only difference is the host project itself, which simply bootstraps the service or application to startup in the correct application host environment (in just a few lines of code). So, other than this simple host project, all the code across the on-premise and cloud deployments is the same, referencing the same shared projects and/or shared libraries. However, for convenience, the Azure Web projects are separated out into separate projects and solutions. The web pages in these solutions are exact copies of those in the on-premise projects for the StockTrader Web application and ConfigWeb. The key differences are in the host project including gobal.asax and web.config. These Azure projects reference all the same shared projects as the on-premise host solutions.

The list of directories and solutions is documented in the following section, and because of the shared libraries utilized, a complete proper build order (depending on what you change) is:

1. Configuration Service Solution
2. StockTrader Order Processor Solution
3. StockTrader Business Service Solution
4. StockTrader Web Application
5. Azure Stocker Order Processor Solution
6. Azure Business Service Solution
7. Azure StockTrader Web Application

You could optionally create an MSBuild build batch that could automate this sequence.

Windows Azure Projects

The complete list of Windows Azure Projects is:

- **AzureStockTrader:** The StockTrader Web user interface, an ASP.NET Web application running in a Web Role.
- **AzureBusinessService:** The StockTrader Business Services middle tier processing, running in a Web Role.
- **AzureOrderProcessorService:** The StockTrader Order Processor Service that processes buys and sells, running in a Worker Role.
- **AzureConfigWeb:** The Configuration Service 5.0 Web application running in a Web Role.
- **SQLLoaderWorker:** A database loader to load the AzureStockTrader Database with very large datasets (for benchmarking), running in a Web Role. This includes a Web user interface to monitor data load progress. This application works in conjunction with the SQL Loader .NET Windows application also included, which enables you to create SQL Azure StockTrader databases and choose their edition (Web, Business); and their size (1GB-50GB). You can expect about 1GB taken per 6 million records loaded.

The Builds Folder

For solutions that output executable code, such as the .NET StockTrader self hosts, the database loaders and the Capacity Planner, there is a special Builds folder that contains the executables for each, along with any dependent assemblies such that the programs can be directly executed from here. The solutions have post-build steps such that the latest compiled version will be copied to the appropriate \Builds subfolder for the specific application. The Windows Start menu items for StockTrader link to the compiled executables contained in this subfolder. The StockTrader self-hosts that host WCF services, if started, should always be **run 'As Administrator'** since they register http endpoints on the host machine. Hence, they will not start properly on Windows 7 if 'Run As Administrator' is not selected.

The Configuration Service Source and Solution

To view/modify the Configuration Service source code, open the solution in the \Configuration subdirectory. This solution contains all of the projects for the Configuration Service and ConfigWeb. Each project has a post-build command to copy the compiled output into the \SharedLibraries\Configuration directory, where all StockTrader and the various other sample projects in the kit reference them. The Azure ConfigWeb WebRole solution is located at \stocktrader\azuretrade\AzureConfigWeb. This is a copy of the same Web pages in the \Configuration\ConfigWeb directory, with some minor differences in global.asax and web.config, in addition to the Azure WebRole project itself.

The Main .NET StockTrader Visual Studio Solutions

In general, you want to open the Visual Studio solutions (not the individual projects); as they are intended to be grouped as units.

The base Visual Studio solutions for the .NET StockTrader sample are:

- **.NET StockTrader Order Processor Service Solution**
 - \stocktrader\StockTraderOrderProcessorService folder. Open the Visual Studio solution contained in this folder, which contains all projects related to the partitioned Order Processor Service.
- **.NET StockTrader Business Service Solution**
 - \stocktrader\StockTraderBusinessService folder. Open the Visual Studio solution contained in this folder, which contains all the projects related to the .NET StockTrader Business Service layer.
- **.NET StockTrader Web Application**
 - \stocktrader\StockTraderWebApplication folder. Open the Visual Studio solution contained in this folder, which contains all the projects related to the Web application.
- **.NET StockTrader WPF Smart Client**
 - \stocktrader\StockTrader_SmartClient_WindowsPresentationFoundation folder. Open the Visual Studio solution contained in this folder for developing/viewing code for the .NET StockTrader smart client application.
- **Azure StockTrader Web Application**
 - \stocktrader\AzureTrade\AzureStockTrader. Open the Visual Studio solution contained here to work with the Azure WebRole project and publish/deploy to Windows Azure.
- **Azure StockTrader Business Service Application**
 - \stocktrader\AzureTrade\ AzureBusinessService. Open the Visual Studio solution contained here to work with the Azure WebRole project and publish/deploy to Windows Azure.
- **Azure StockTrader Order Processor Service Application**
 - \stocktrader\AzureTrade\ AzureOrderProcessorService. Open the Visual Studio solution contained here to work with the Azure WorkerRole project and publish/deploy to Windows Azure.

Supplementary .NET StockTrader Visual Studio Projects and Solutions

There are several other solutions that contain source code to various extras in the download. Open the Visual Studio solutions contained in the folders called out below:

- Database Loader Program for SQL Server
 - \databaseloaders\tradesqlloader
- Database Loader for SQL Azure

- \databaseloaders\tradeSQLAzureLoader
- Used in conjunction with the Database Loader Program for SQL Server above. This is a WebRole Azure project that you can deploy to the same Azure region as the SQL Azure Trade database. As such, the data load is run from within the same data center as the SQL Azure database, greatly speeding large data loads to SQL Azure. Otherwise data load performance is gated by your corporate proxy server and/or ISP connection to the Internet.
- Database Loader Program for Oracle 11G (requires installation of Oracle's 11G client with the Oracle Data Provider for .NET; a free download from Oracle Corporation's Web site). For 5.0, the Oracle loader and StockTrader Oracle DAL have been updated for .NET Framework 4.0 and Oracle's latest client as of June 1, 2011.
 - \databaseloaders\tradeoracleloader
- Database Loader Program for IBM DB2 v9.7 (requires installation of IBM's DB2 v9.7 Client, from IBM's Web site)
 - \databaseloaders\tradedb2loader

Configuration Service Template and Solution Wizard

The test harness solutions that shipped with previous versions of .NET StockTrader that illustrated Configuration Service implementations in non-StockTrader applications have been replaced by this VS 2010 Solution Wizard/Project type.

- \configuration\ConfigServiceVSTemplate\ConfigService5Template.zip is the template that should be installed in your [user]\Documents\Visual Studio 2010\Templates\ProjectTemplates folder. You will need to run "gacutil -i
 \stocktrader\Configuration\ConfigServiceVSTemplate\Template_Wizard\bin\x86\Release\ConfigService.RepositoryCreate.dll"
 Using: \Program Files\Microsoft SDKs\Windows\v7.1\Bin\NETFX 4.0 Tools\gacutil.exe
- You will also need to make sure you have created the Environment Variable: 'StockTraderInstall' and set value to base stocktrader install directory for example c:\stocktrader.
- Note the StockTrader setup program does these two steps automatically as part of the install.

Capacity Planner Solution

The solution with all the projects and source code to the Capacity Planner (including the .NET implementation of the WSTest benchmark workload) is in the \capacityplanner subfolder. Open the solution in this subfolder, which contains all the individual projects (for both the controller and agent elements).