

Table of Contents

Overview.....	2
Functional Updates.....	2
Prerequisites.....	2
Prerequisites for Setup.....	2
Prerequisite for Source Code	2
Installation.....	3
Visual Studio Project Templates.....	5
Version Information	5
Introduction.....	6
Installation.....	6
Manage.....	7
Create Applications with Project Templates	8
Project Templates.....	8
Layered Service Application (Entity Framework)	8
Layered Service Application (NHibernate)	14

Overview

Last week I published a new version of Apworks framework, the version number was 2.5.4563.21309. You can download the installer and source code from <http://apworks.codeplex.com>.

Functional Updates

Apworks 2.5.4563.21309 has the following enhancements and functional updates:

1. Visual Studio 2010 project templates are available for easily creating Apworks based applications (Will discuss later)
2. Changed the type of the entity ID from System.Long to System.Guid
3. New WcfPerRequestLifetimeManager added
4. Added another Repository implementation by using Microsoft Entity Framework v4.3.1
5. Repository now supports "Eager Loading", used for avoid lazy loading on some specific entity properties
6. Pagination is supported in Repositories
7. Added a new namespace: Apworks.Services.ApplicationServices. DataTransferObject class is defined under this namespace to support the design of Data Transfer Objects
8. Updated all the 3rd party components to their latest versions
9. Fixed the GC exception that occurs when NHibernateContext calls its Rollback method after transaction fails
10. Fixed the session unavailable issue that happens when commit fails in NHibernateContext and the NHibernate session was not closed properly
11. Fixed the XML documentation for Repository implementation classes

Prerequisites

Prerequisites for Setup

If you want to install Apworks directly from the installer, please make sure the machine meets the following prerequisites:

- Windows XP Professional SP3 (Or higher)
- Microsoft .NET Framework 4.0 Full (Or higher)
- Microsoft Visual Studio 2010 SP1 Ultimate/Professional (Note: If you wish to use the project templates)

Prerequisite for Source Code

If you want to open the source code in Visual Studio 2010, build the source code and run the unit test cases, please make sure the machine meets the following prerequisites:

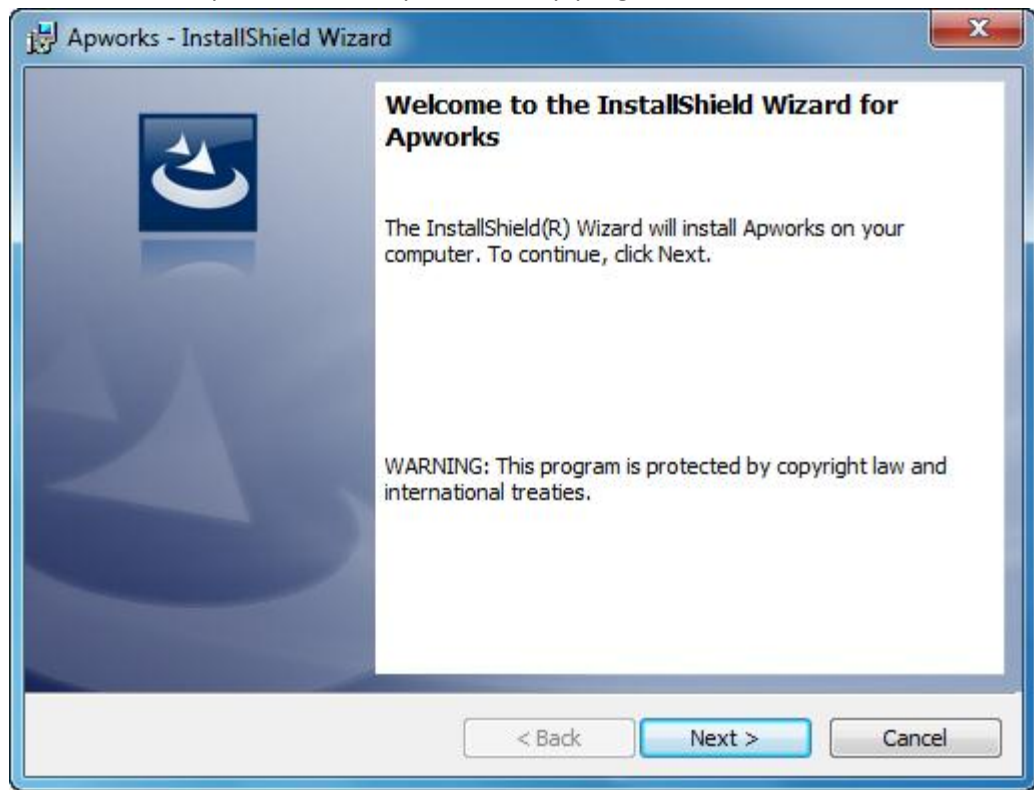
- Windows XP Professional SP3 (Or higher)

- Microsoft .NET Framework 4.0 Full
- Microsoft Visual Studio 2010 SP1 Ultimate/Professional
- Microsoft SQL Server (2005/2008/Express/Standard/Enterprise)
- Oracle MySQL
- Configuration Section Designer Visual Studio Extensions
- WiX Toolset v3.6
- InstallShield Limited Edition
- Microsoft MSMQ
- Message Queuing service started properly
- Microsoft Distributed Transaction Coordinator service started properly

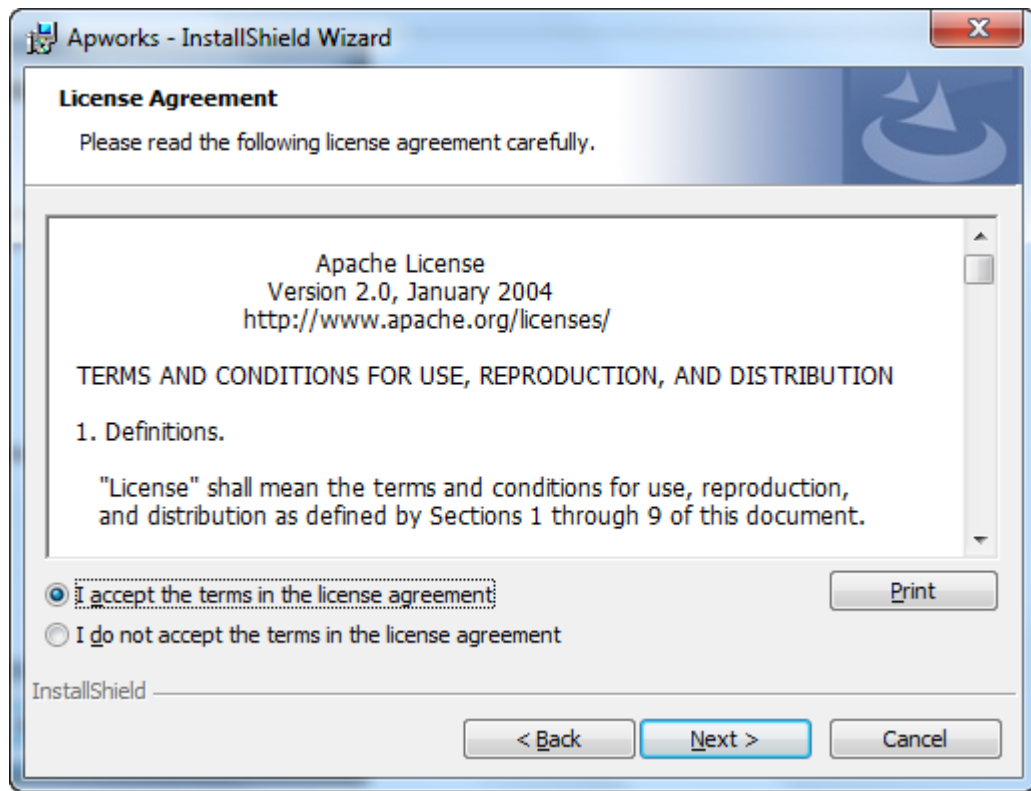
Installation

From version 2.5, Apworks uses the setup program generated by InstallShield, so you may see a different user interface compared with the early versions.

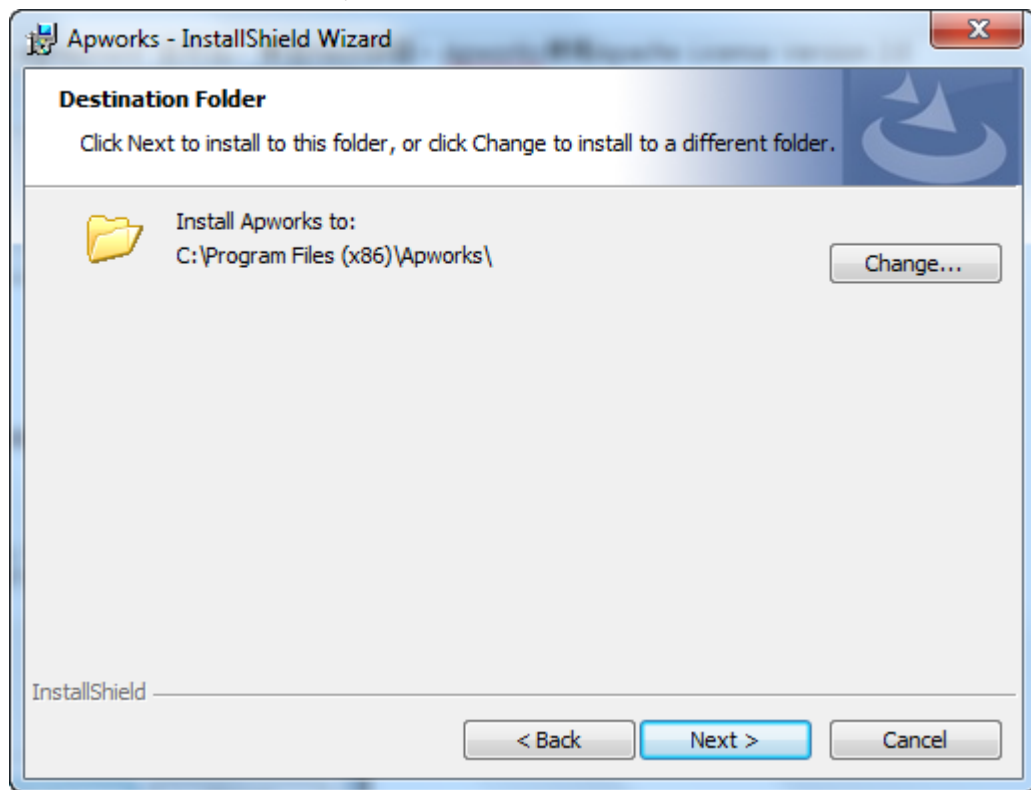
1. Download the setup.exe, you can click the following link to start the downloader:
<http://apworks.codeplex.com/downloads/get/418848>
2. Double click setup.exe, this will open the setup program:



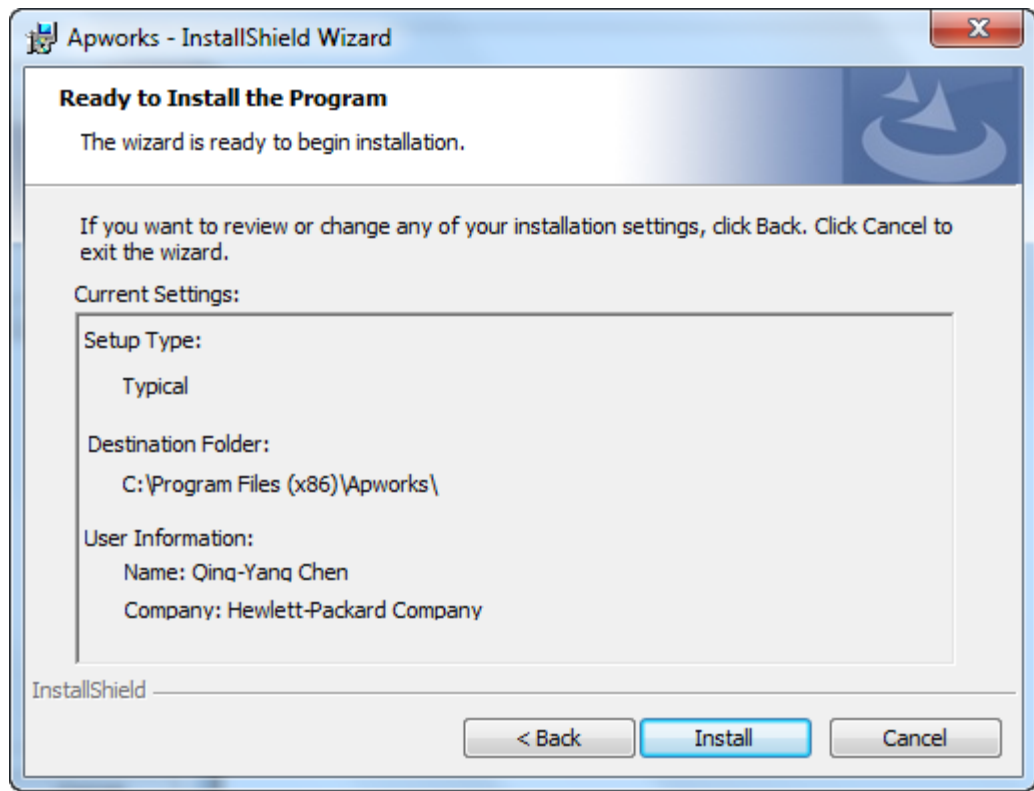
3. Click the Next button, check the "I accept the terms in the license agreement" check box, then click Next button. Apworks uses Apache License v2.0



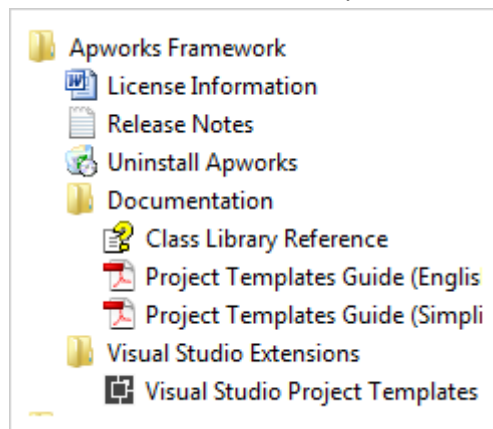
4. Select the installation folder, then click Next



- Click the Install button to start installation



- Click the Finish button
- From Windows Start menu, you can find the shortcuts for Apworks:



Visual Studio Project Templates

Version Information

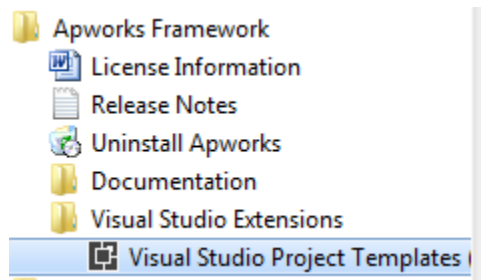
Product	Version
Apworks	2.5.4563.21309
Visual Studio	10 SP1 (VS2010 SP1)

Introduction

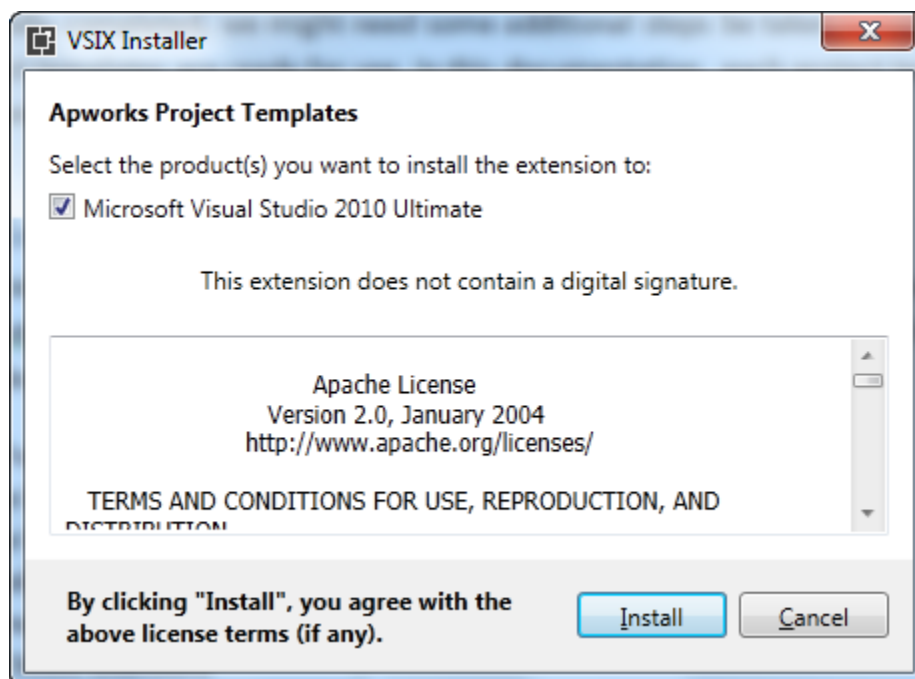
In order to help developers getting started with Apworks application development much faster, from version 2.5, Apworks carries lots of Visual Studio project templates along within its installation package. Since Apworks installer was built with InstallShield LE (Limited Edition), many wonderful functionalities was disabled by its limited license, so after the setup program was successfully completed, we might need some additional steps be taken before the Visual Studio project templates are ready for use. In this documentation, each project template will be introduced and more detailed information is provided.

Installation

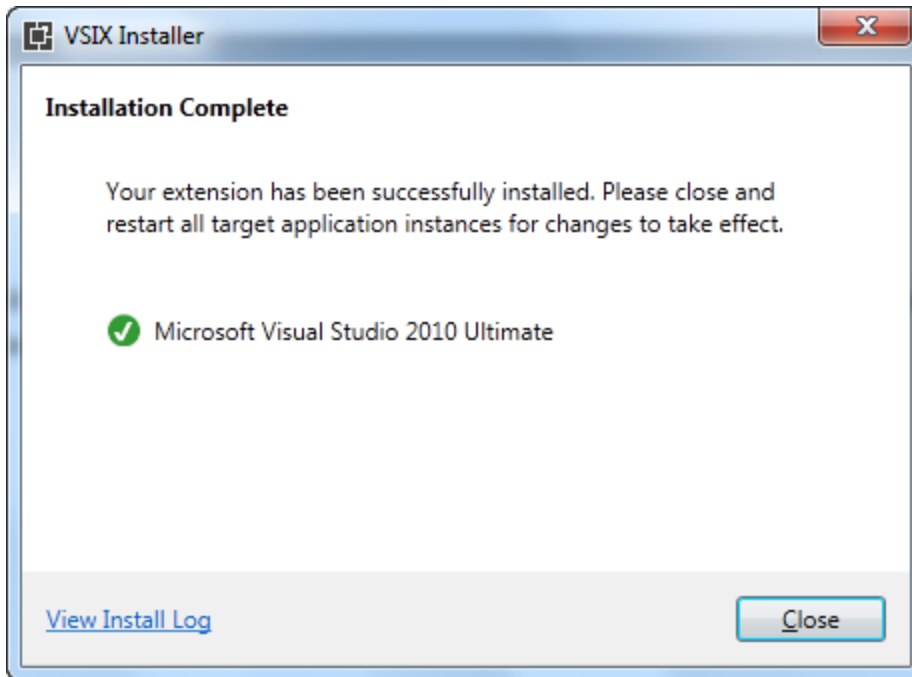
Apworks installer won't install the Visual Studio Extensions (VSE) that contains the project templates automatically, users should install the VSEs manually by accessing the Start menu shortcut under Apworks Framework folder, just like below:



After clicking this menu item, the VSIX Installer will appear, users are able to select the available Visual Studio versions to install.

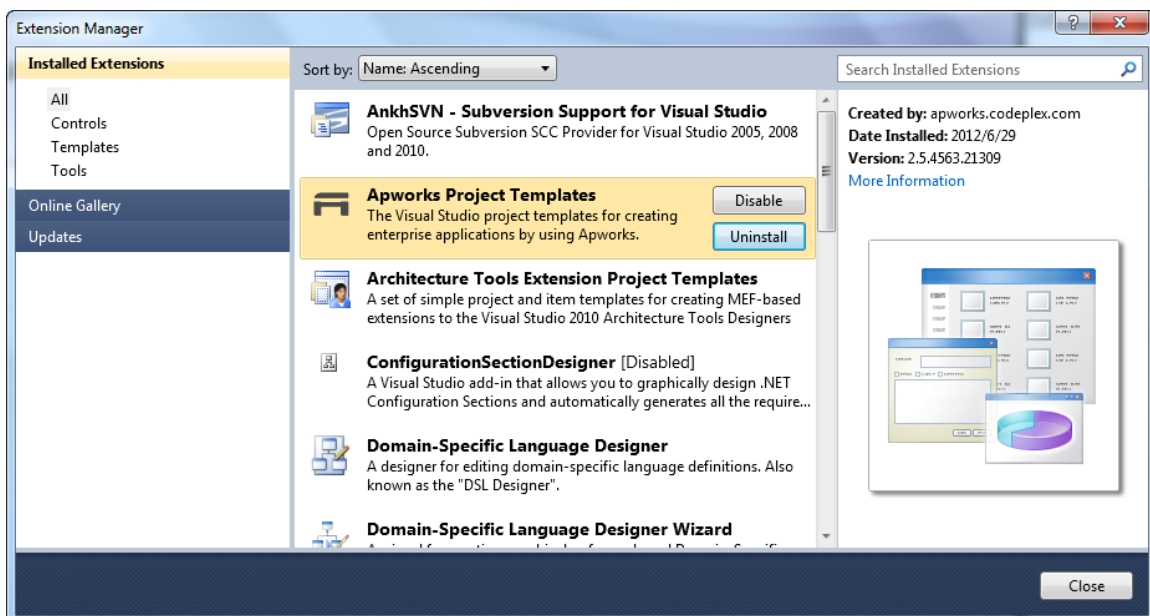


Now start the Visual Studio Extension installation by clicking the Install button on the dialog box, if the installation has successfully completed, you will see the following screen:



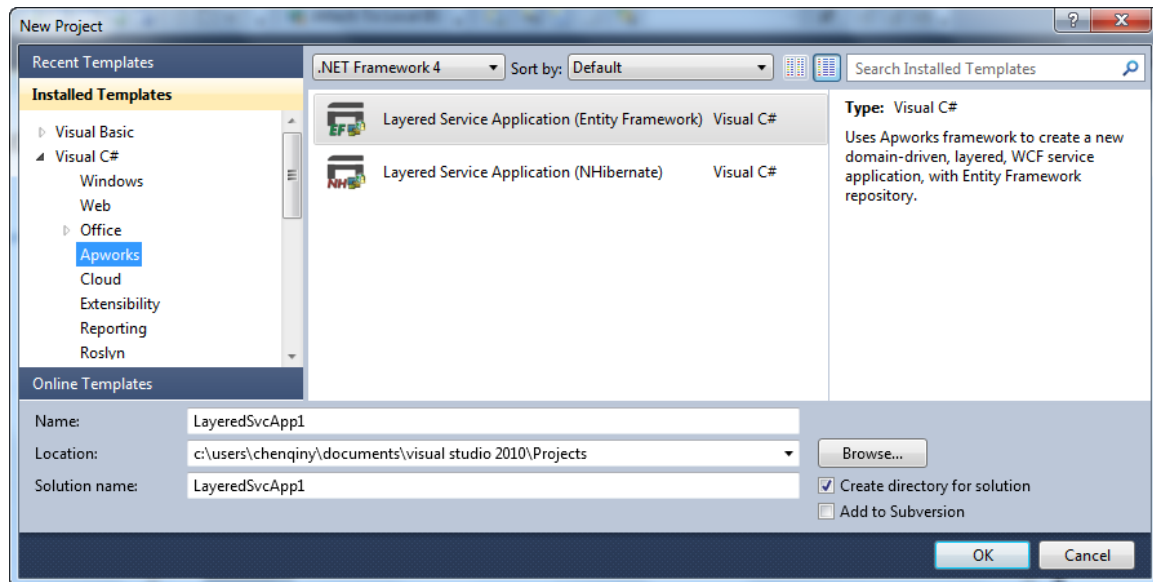
Manage

After the Visual Studio Extension has been installed, you can Enable/Disable the extension by using Visual Studio's Extension Manager dialog box (From Tools -> Extension Manager menu in Visual Studio). You can also uninstall the extension there.



Create Applications with Project Templates

In Microsoft Visual Studio, click File -> New -> Project menu, this will open the New Project dialog box. Under Installed Templates group, under Visual C# category, you can find a sub category named “Apworks”, all the Apworks related project templates are available here. Note that Apworks project templates only support Visual C# programming language, so you cannot find this sub category under other language groups.



Select a project template you wish to use, and like creating other types of applications, simply give your solution a name and click OK button. Visual Studio will create the project for you.

Project Templates

In this section, all available project templates will be introduced.

Layered Service Application (Entity Framework)



This project template will create a sample service application solution with the traditional layered architecture. The created sample application is based on a very simple scenario: Users are able to maintain their Notes, and it will use the Microsoft ADO.NET Entity Framework as the implementation technology of the repositories.

Note that the created application is a WCF service application, the project template doesn't provide any GUI logic to present the sample application. Developers should handle the GUI logic by themselves. This is because GUI development technology grows rapidly and it is hard to make Apworks targeting a specific version of some kind of UI frameworks.

To create a Layered Service Application (Entity Framework), in the New Project dialog, select this template and provide a project name, then click OK and follow the steps below:

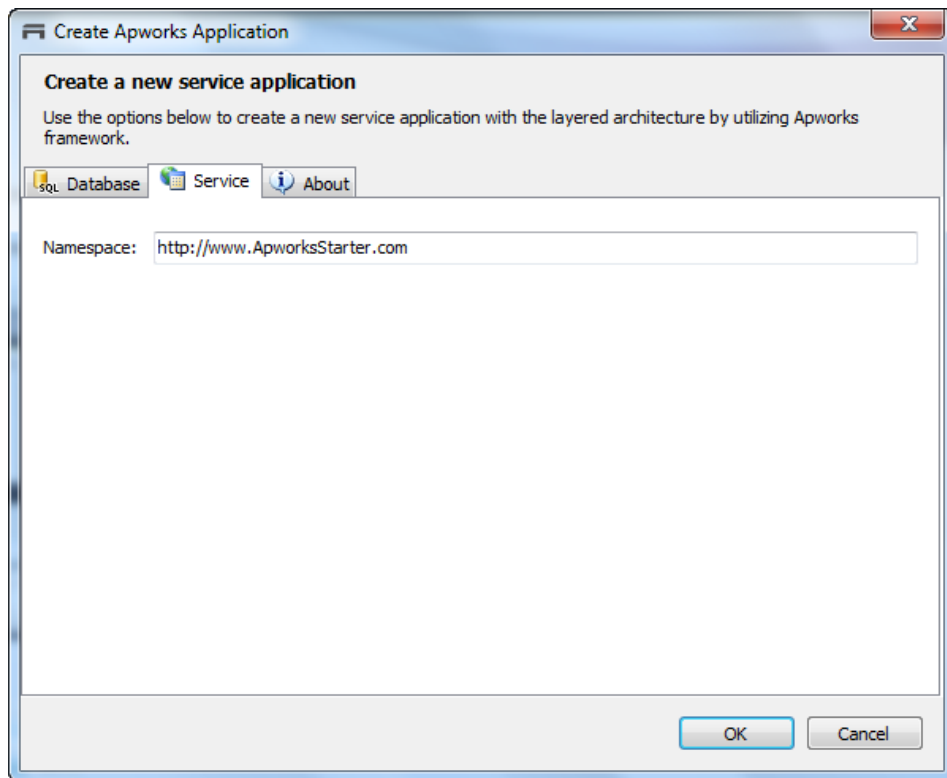
1. After OK button is clicked on the New Project dialog, the Create Apworks Application dialog box appears. There are three tabs on this dialog box:
 - Database – Provides the database information here, the database will be used as the backend data store for the new application. Note that currently this project template only supports SQL Server databases. If you want to use other databases (Like Oracle, or even NoSQL databases), you can change your connection string in the Service project's Web.config file. This is the project template's limitation, Apworks framework allows you to use any kind of database per your requirement

The screenshot shows a Windows-style dialog box titled "Create Apworks Application". It has three tabs: "SQL Database" (selected), "Service", and "About". The "SQL Database" tab contains the following fields and controls:

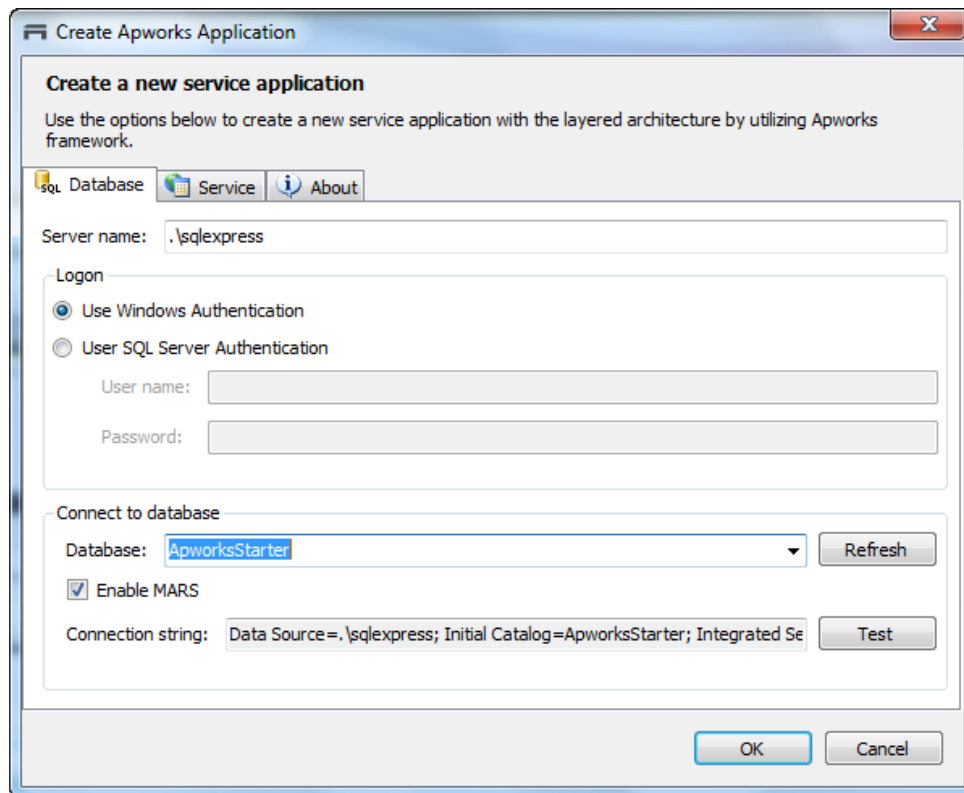
- Server name:** A text input field.
- Logon:** A section with two radio buttons: "Use Windows Authentication" (selected) and "User SQL Server Authentication". Below these are "User name:" and "Password:" text input fields.
- Connect to database:** A section with a "Database:" dropdown menu, a "Refresh" button, and a checked checkbox for "Enable MARS".
- Connection string:** A text area containing the default string: "Data Source=; Initial Catalog=; Integrated Security=SSPI; MultipleActive". A "Test" button is to its right.

At the bottom right of the dialog are "OK" and "Cancel" buttons.

- Services – Provides the WCF service-related configuration here. For now you only need to specify the Namespace for the service. By default, it will use <http://www.XXXX.com> as the Namespace, where XXXX stands for your solution name

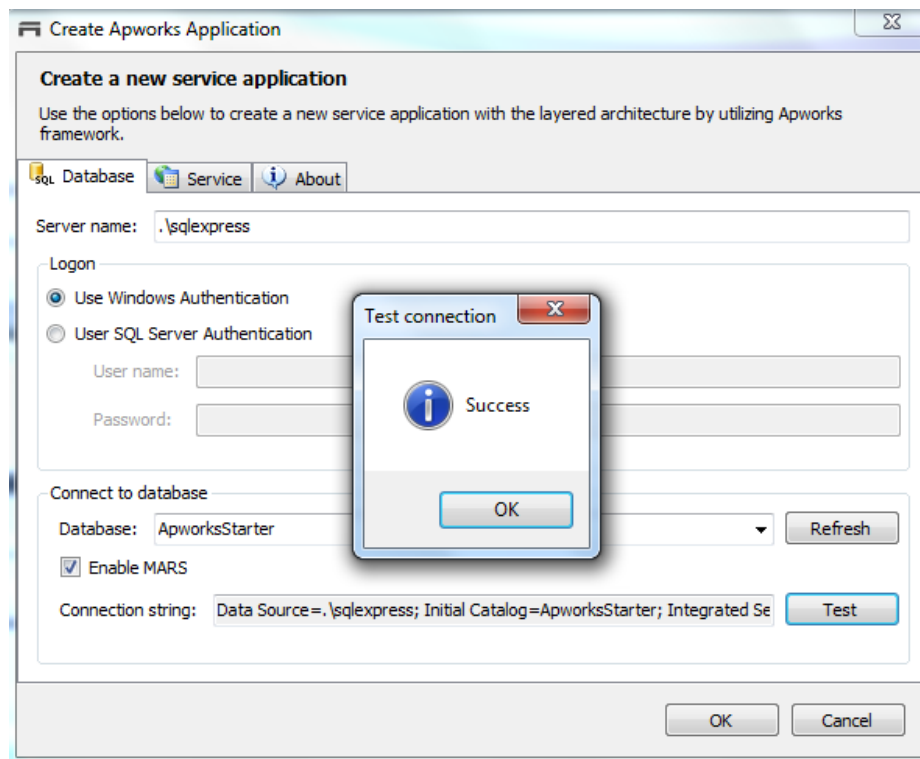


- About – About the project template
2. Under the Database tab, select the SQL Server database you wish to use for the new application:



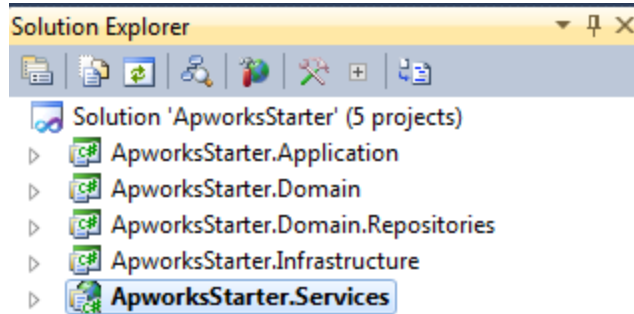
Make sure that the Enable MARS option is checked.

You can also click the Test button to test the connection:

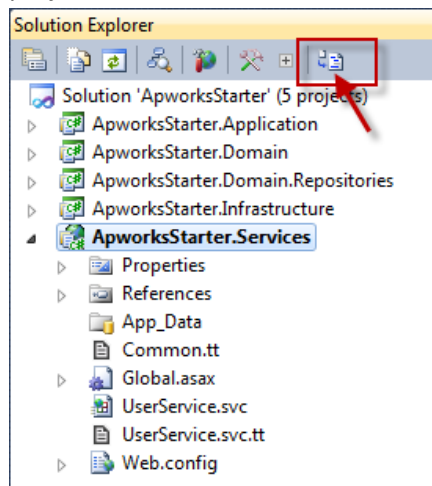


3. Under the Service tab, enter the proper namespace to use for the WCF service

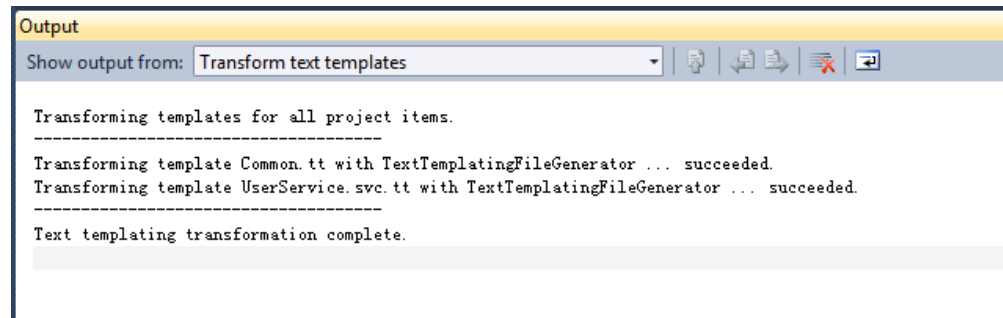
4. Click OK button, Visual Studio will do the following tasks for you:
 - a. Create the Tables and Data in your selected database
 - b. Compile the entire solution – This is required because the Text Templates under the generated Service project will need to use the compile output from other projects to generate the source code
5. Following projects will be created in Solution Explorer:



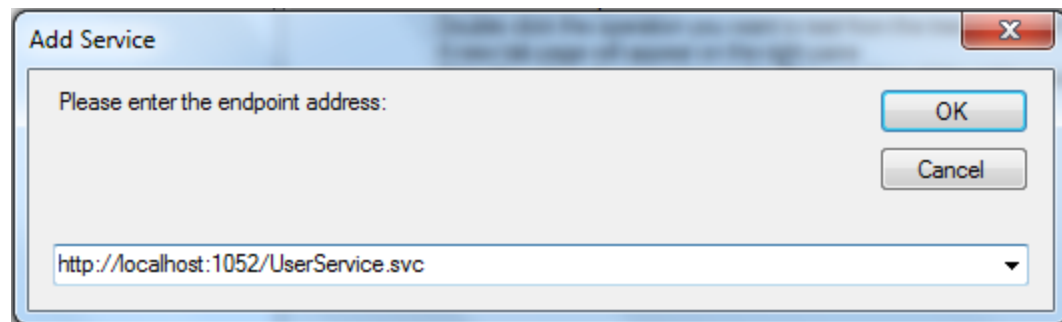
- ApworksStarter.Application – The C# Class Library project that contains application interfaces and application service implementations
 - ApworksStarter.Domain – The C# Class Library project that contains the Domain Model, repository interfaces, specifications and O/R mapping schemas
 - ApworksStarter.Domain.Repositories – The C# Class Library that contains the repository implementations
 - ApworksStarter.Infrastructure – The C# Class Library that contains all the facilities at infrastructure level. Currently the project created by this project template only has a method extender class defined. You can add other utility classes here based on the requirement
 - ApworksStarter.Services – The WCF Service application which initializes the application and exposes the interfaces to the clients
6. Click the ApworksStarter.Services project in the Solution Explorer, then click Transform All Templates toolbox button, this will transform all the templates defined under this project:



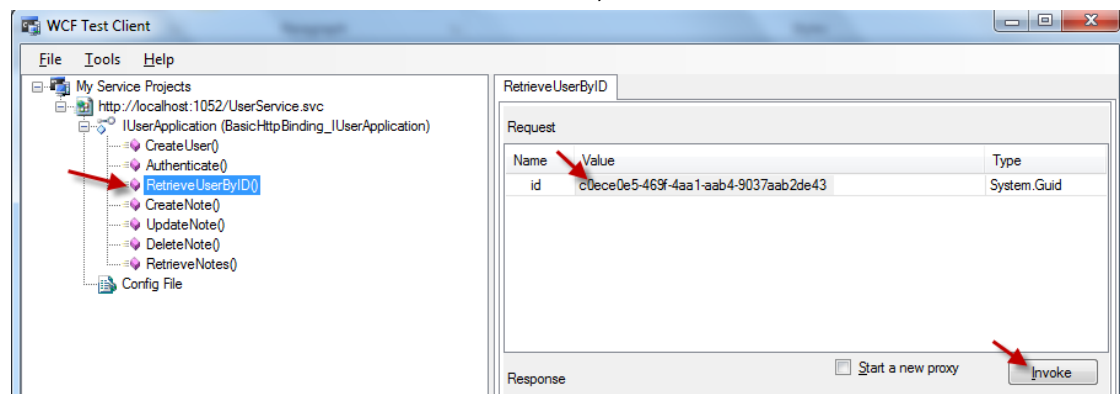
Transform result will be displayed in Visual Studio's Output window:



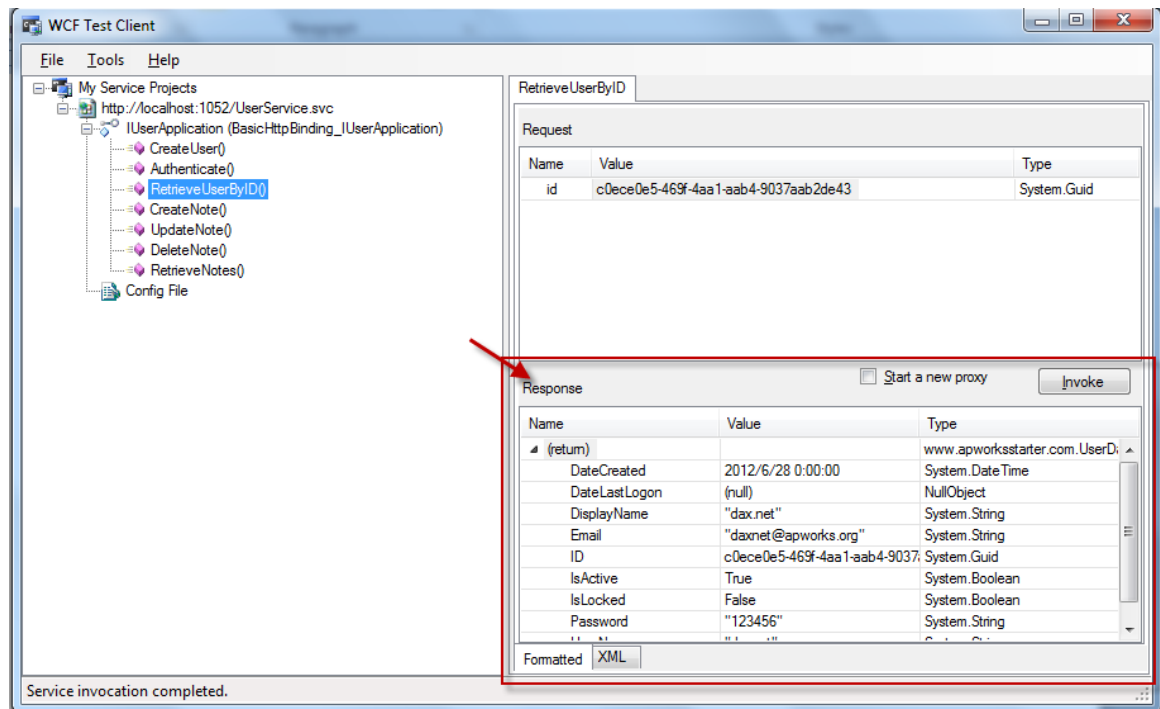
7. Press F5 to start the debugger
8. Start WCF Test Client
9. In WCF Test Client, click File -> Add Service menu, it will bring up the Add Service dialog box
10. In the Add Service dialog box, enter the endpoint address as follows:



11. Double click the RetrieveUserByID operation, in the Request panel, enter "c0ece0e5-469f-4aa1-aab4-9037aab2de43" for the "id" field, then click the Invoke button:



12. After the service has been called successfully, data will be displayed in the Response panel as below:



- Now you can create your GUI project under the solution and invoke the service to provide functionalities

Layered Service Application (NHibernate)



This template will create the same sample application as the Layered Service Application (Entity Framework) does, except that this template will use the NHibernate as the repository implementation technology.

For other information about this project template, please refer to "Layered Service Application (Entity Framework)" in the above section.