|  |
| --- |
| Apworks™ |
| Visual Studio 项目模板介绍 |
| 本文档将对Apworks所提供的Visual Studio项目模板进行详细介绍。 |

6/29/2012

Table of Contents

[版本信息 2](#_Toc328751188)

[概述 2](#_Toc328751189)

[安装 2](#_Toc328751190)

[管理 3](#_Toc328751191)

[使用项目模板创建应用程序 4](#_Toc328751192)

[项目模板 5](#_Toc328751193)

[Layered Service Application (Entity Framework) 5](#_Toc328751194)

[Layered Service Application (NHibernate) 10](#_Toc328751195)

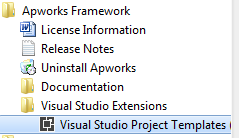
# 版本信息

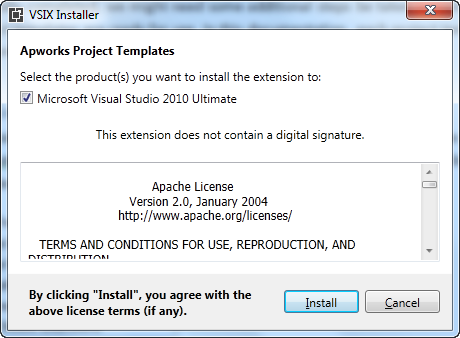
|  |  |
| --- | --- |
| 产品 | 版本 |
| Apworks | 2.5.4563.21309 |
| Visual Studio | 10 SP1 (VS2010 SP1), 11 (VS2012) |
| .NET Framework | 4.0 Full |

# 概述

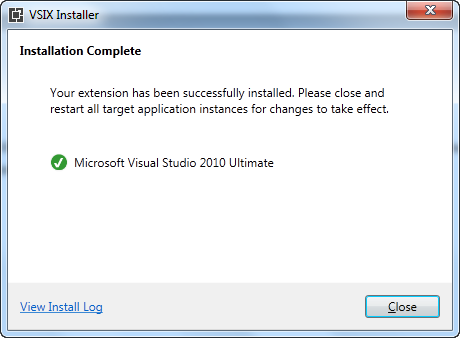
为了帮助开发人员能够更快地使用Apworks进行应用程序开发，从2.5版开始，Apworks将在其安装包内包含Visual Studio的应用程序项目模板。由于Apworks安装程序是用InstallShield LE（Limited Edition，限制版）创建的，由于许可证的约束，在创建Apworks安装程序时并不能完全使用InstallShield所提供的强大功能。为了在Apworks安装成功后，开发人员能够方便地使用Visual Studio应用程序项目模板，我们可能需要执行一些额外的操作。在本文档中，将对每个项目模板进行详细介绍。

## 安装

Apworks安装程序无法直接安装包含在Visual Studio扩展（VSE）中的项目模板，用户应该通过在开始菜单中访问Apworks目录以手动安装Visual Studio扩展，如下： 

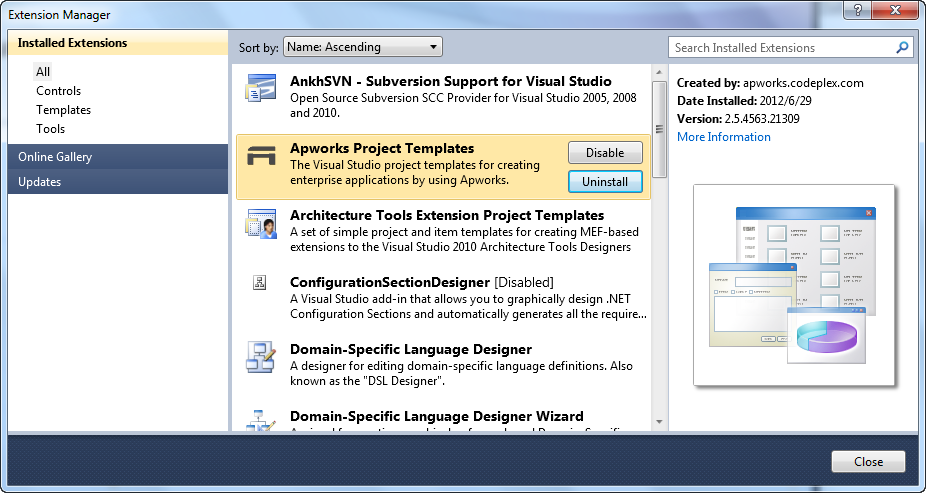
点击这个菜单项后，会出现VSIX Installer对话框，用户可以选择相应的Visual Studio版本进行安装。

现在开始安装Visual Studio扩展，通过点击对话框上的“安装”按钮，如果安装成功完成后，您会看到以下画面：



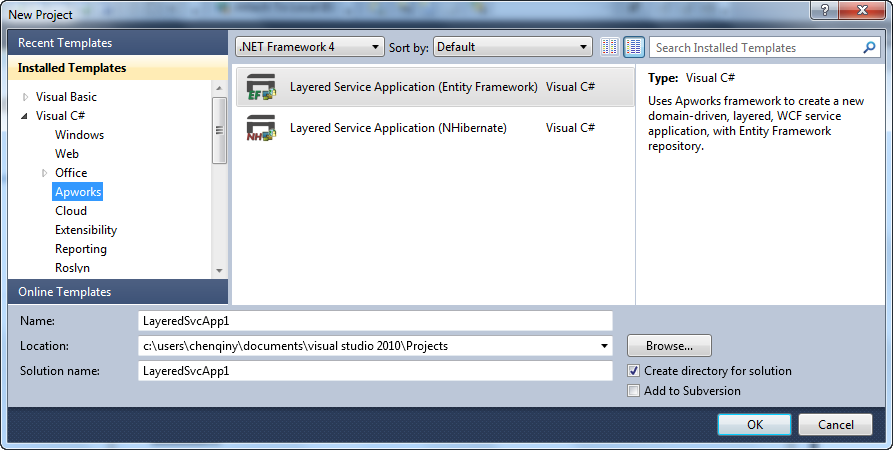
## 管理

安装完成后，您可以启用/禁用扩展，通过使用Visual Studio的扩展管理器对话框（在Visual Studio中选择“工具” - >“扩展管理器”菜单）。您还可以卸载扩展。



## 使用项目模板创建应用程序

在Microsoft Visual Studio中，单击“文件 - >新建 - >项目”菜单，这将打开“新建项目”对话框。在Installed Templates的Visual C#分类下，您可以找到一个名为“Apworks”的子类，所有与Apworks相关的项目模板都在这个子类下。注意，Apworks项目模板只支持Visual C#编程语言，所以您无法在其它语言的分类下找到Apworks子类。



选择您想使用的项目模板，像创建其他类型的应用一样，给您的解决方案取个名称，并单击“确定”按钮。 Visual Studio会为您创建的项目。

# 项目模板

在本节中，将对所有的项目模板进行介绍。

## Layered Service Application (Entity Framework)



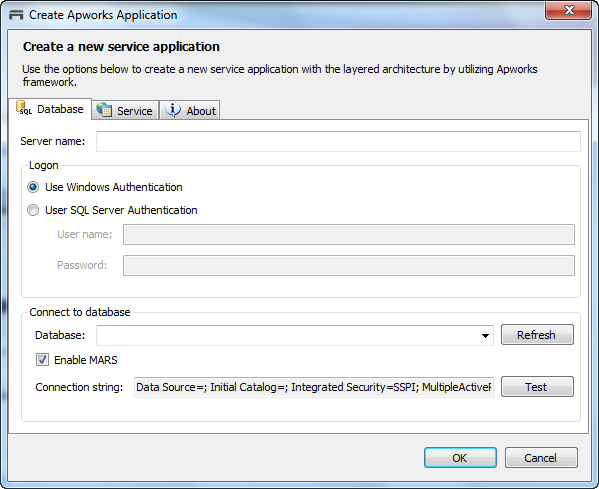
此项目模板将创建一个基于传统的分层架构的示例服务应用解决方案。所创建的示例应用程序基于一个非常简单的场景：用户能够维护他们的笔记条目。这个应用程序将使用微软的ADO.NET Entity Framework作为仓储的实现技术。

请注意，所创建的应用程序是一个WCF服务应用程序，项目模板不提供任何用于呈现应用程序界面的GUI逻辑。开发人员应根据项目的实际情况将GUI部分的代码引入解决方案中。这是因为与用户界面相关的技术发展迅速，开发平台以及框架变化多端，版本各异，让Apworks所创建的应用程序绑定到一个特定的界面技术显然不是一个很好的选择。

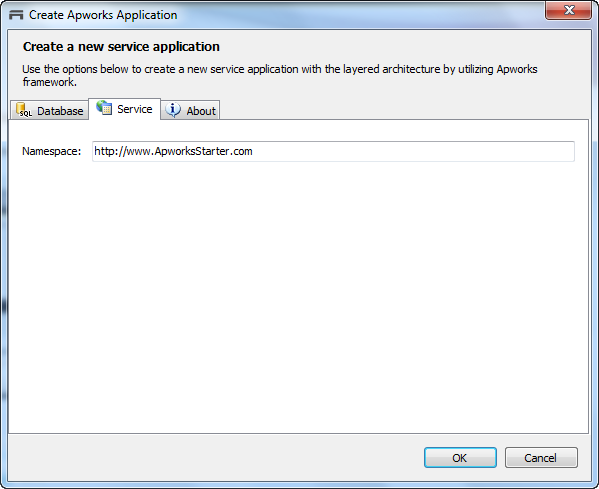
要创建Layered Service Application (Entity Framework)，在“新建项目”对话框中，选择该模板，并提供项目名称，然后单击“确定”，并按照以下步骤：

1. 在“新建项目”对话框中点击OK按钮后，出现Create Apworks Application对话框。在此对话框中有三个标签页：

* Database - 提供数据库中的信息，该数据库将被新的应用程序用作后端数据存储。请注意，目前此项目模板只支持SQL Server数据库。如果您想使用其他数据库（如Oracle，或者是NoSQL数据库），您可以更改Services项目的Web.config文件中的连接字符串。这是项目模板的限制，Apworks框架允许您根据实际项目需求使用任何一种数据库

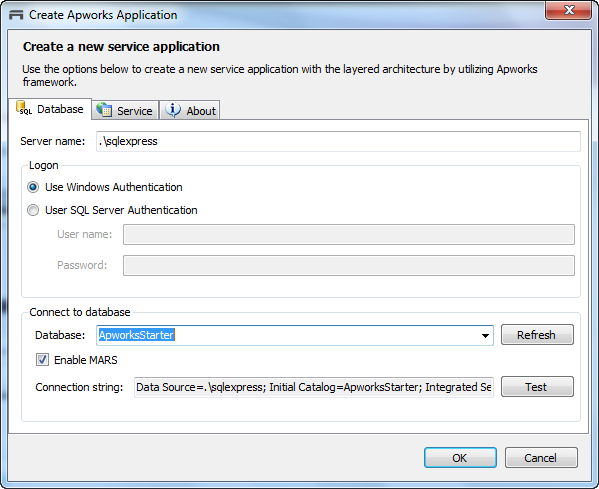


* Services –这里提供与WCF服务相关的配置。目前，您只需要指定WCF服务的命名空间。默认情况下，它会使用<http://www.XXXX.com>作为命名空间，其中XXXX为您的解决方案的名称



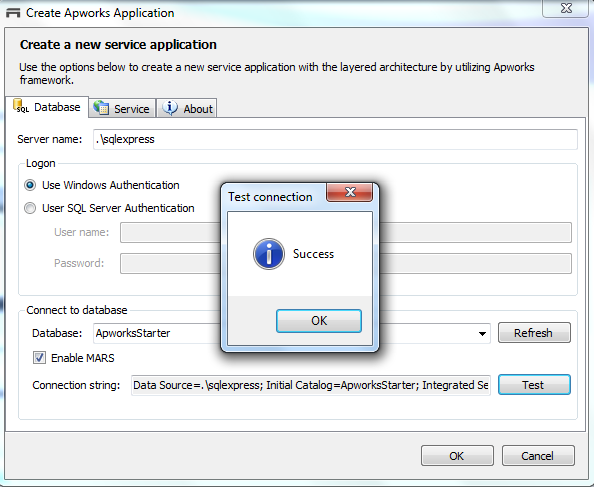
* About – 关于项目模板

1. 在“数据库”选项卡，选择您想使用的SQL Server数据库：

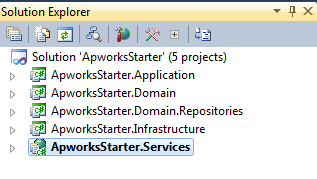


确保“Enable MARS”选项被选中。

您还可以点击“测试”按钮来测试连接：

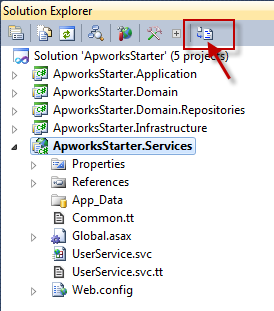


1. 在“服务”选项卡，输入合理的命名空间
2. 点击OK按钮，Visual Studio会为您完成以下任务：
   1. 在您选定的数据库中创建表和数据
   2. 编译整个解决方案 - 这是必需的，因为在转换Services项目下的文本模板时，需要引用其它几个项目的编译结果
3. 在解决方案资源管理器中，将创建下列项目：

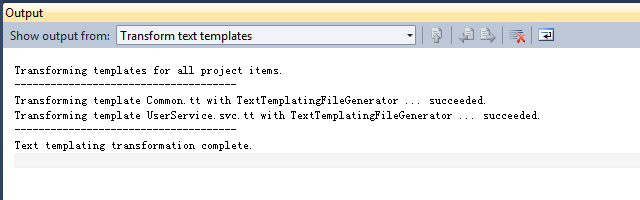


* ApworksStarter.Application –此C#类库项目包含应用程序接口和应用服务的实现
* ApworksStarter.Domain – 此C#类库项目包含领域模型、仓储接口、规约和O / R映射架构
* ApworksStarter.Domain.Repositories – 此C#类库项目包含仓储的实现
* ApworksStarter.Infrastructure – 此C#项目类库包含基础结构层的类型。目前，此项目模板创建的项目只提供了一个用于定义扩展方法的类型。您可以向此项目添加更多的基础结构类型
* ApworksStarter.Services – WCF服务应用程序，用以初始化应用程序，并向客户端公开服务接口

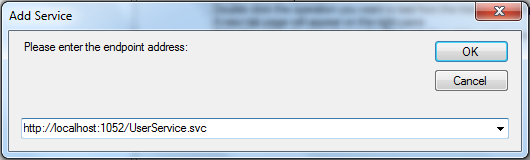
1. 在解决方案资源管理器中点击ApworksStarter.Services项目，然后单击“转换所有模板”工具箱按钮，这将转换这一项目下的所有定义的模板：



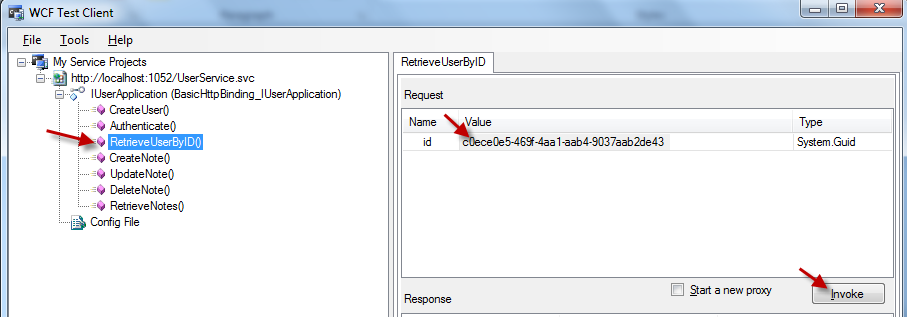
转换的结果将显示在Visual Studio的输出窗口：



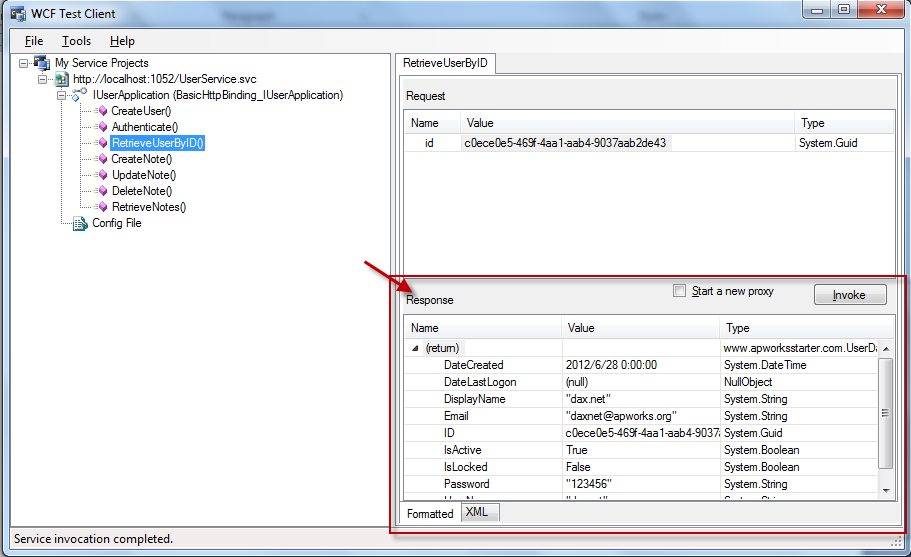
1. 按F5来启动调试器
2. 启动WCF测试客户端
3. 在WCF测试客户端，点击“文件 - >添加服务”菜单，将弹出添加服务的对话框
4. 在添加服务对话框，输入endpoint如下：



1. 在Request面板中，双击RetrieveUserByID，在“id”字段输入“c0ece0e5-469f-4AA1-aa​​b4-9037aab2de43”，然后单击Invoke按钮：



1. 服务被成功调用后，Response面板中显示如下：



1. 现在您可以在解决方案中创建您的GUI项目，并调用此服务提供的功能

## Layered Service Application (NHibernate)



此模板将创建与Layered Service Application (Entity Framework)相同的示例应用程序，不同之处在于，此模板所创建的应用程序将使用NHibernate框架来实现仓储架构。

有关此项目模板的其他信息，请参阅上面的“Layered Service Application (Entity Framework)”部分。