

Visual Studio. Заготовки проектов

Прочитать статью на сайте

В последнее время я не просто много, а очень много экспериментирую с кодом, подходами, написаниями каких-то тестовых сценариев и приложений. Результат, конечно, всегда разный, ну да не об этом сегодня пойдет речь. Постепенно начинаешь уставать от настройки стандартных проектов. Для проекта библиотек надо удалить класс, создаваемый там по умолчанию, потому что все равно его будет лень переименовывать. Мне приятно еще перенаправлять весь вывод компиляции в одну папку в корне всего проекта из всех библиотек, а не шариться по папкам `..\bin\Debug\` для каждого составляющего проекта, выгребать оттуда `*.dll`. Для проектов Test удаление ненужного барахла особенно актуально! Было бы неплохо еще сразу настроить атрибуты для сборки и информацию о ней (Assembly). Ко всему

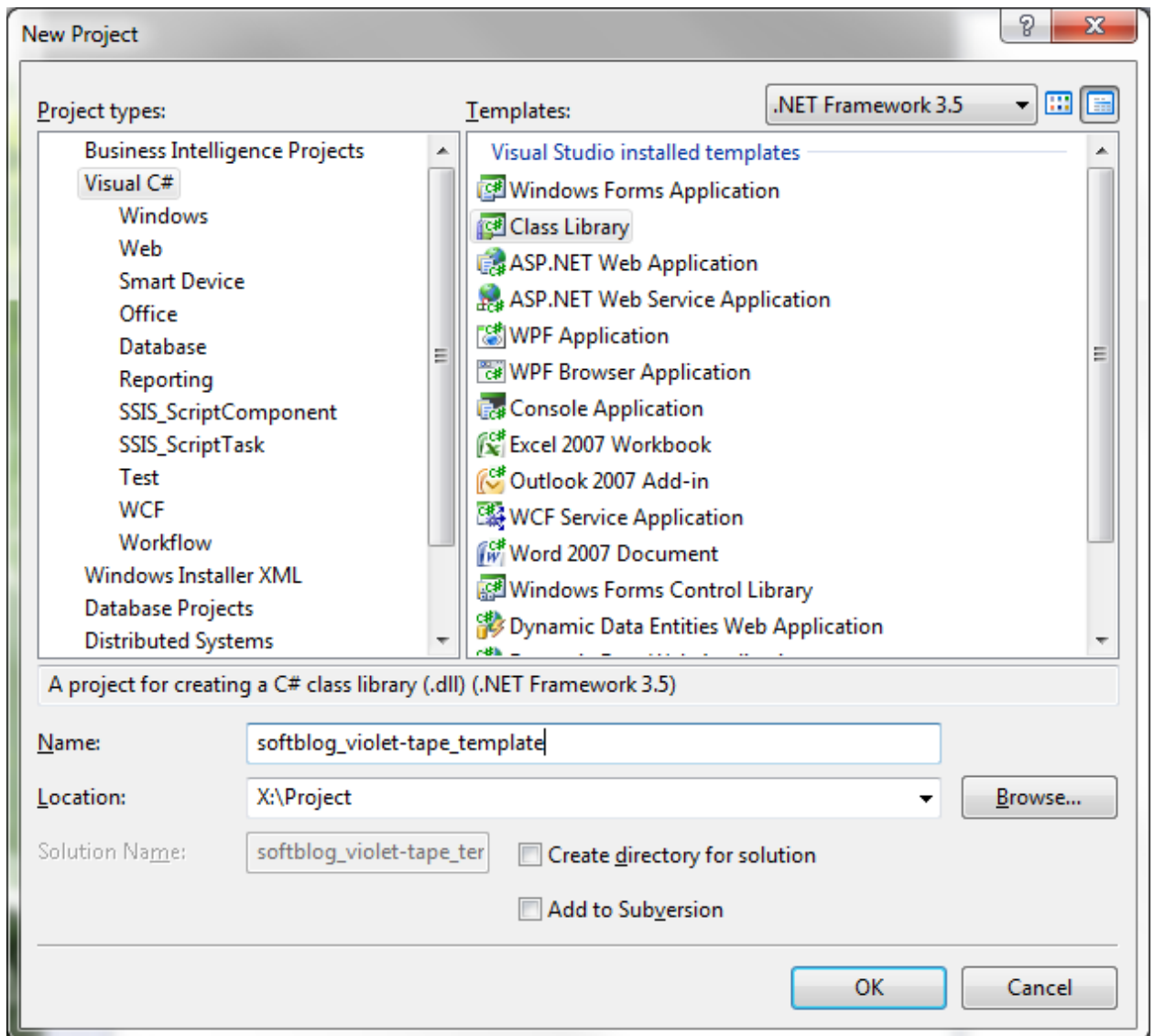
Хочу, чтобы при создании чего-то нового, выбрал проект – и все у тебя настроено! И общая папка вывода, и нужные значения в AssemblyInfo прописаны, и лишнего ничего не валяется в проекте. Подумал я так и вспомнил, что есть у студии опция «Get online Templates», а раз есть онлайн, то должна быть и оффлайн – для моих собственных шаблонов. Сказано сделано!

В результате я очень доволен, что потратил час времени для настройки всех нужных мне проектов и теперь не отвлекаюсь в процессе, а только радуюсь, как все здорово вышло и что не надо тратить время на тупую работу.

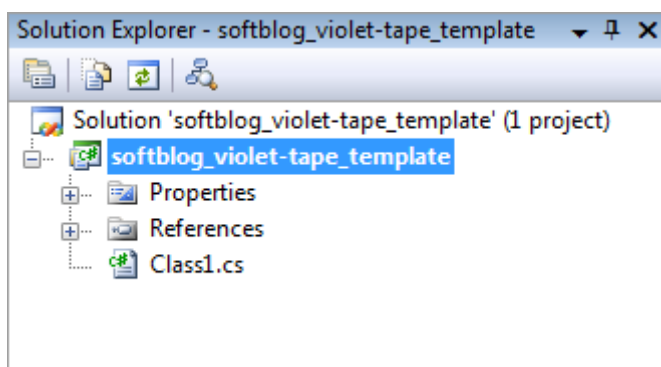
О том, как это сделать в Visual Studio 2008, 2010 и пойдет речь дальше. Так же будет рассмотрено, как удалять, и импортировать готовые шаблоны проектов в студию.

Visual Studio 2008

Начало более чем банально и знакомо до боли каждому – открываем студию и создаем желаемый проект.

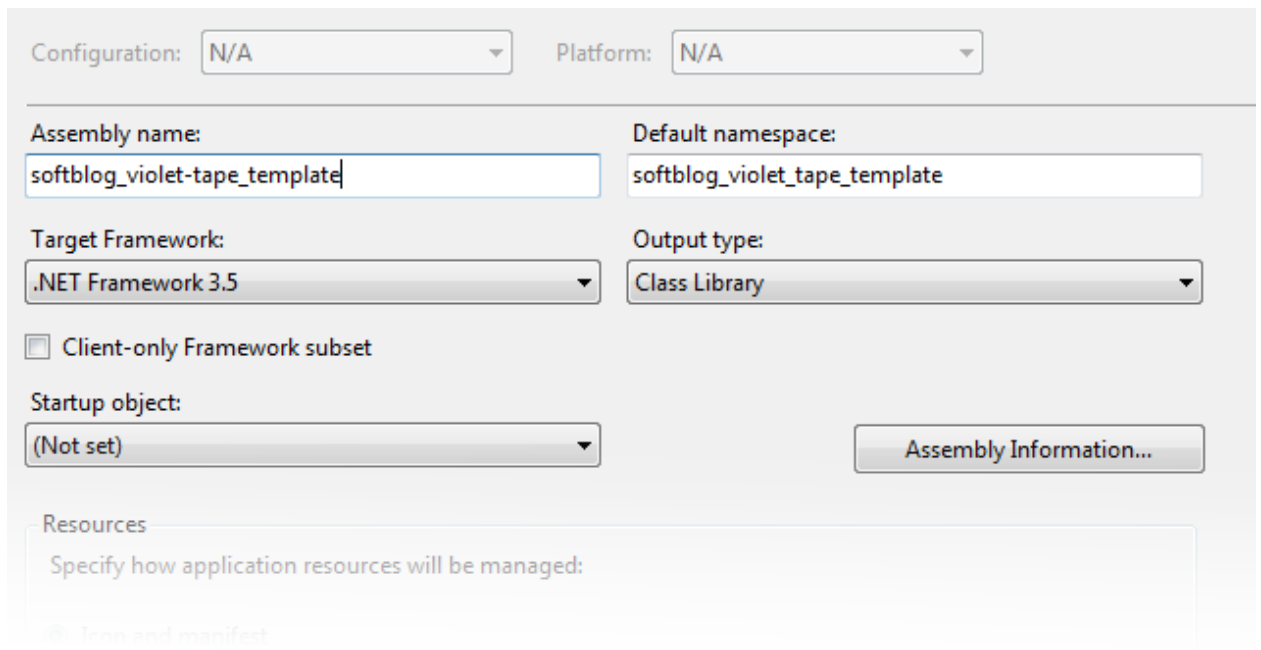


Тренироваться будем на пустой библиотеке, т.к. именно такие проекты чаще всего приходится создавать. После прохождения знакомого визарда по созданию проекта видим стандартную картину нового проекта.



Первым делом удаляем совершенно ненужный файл класса **Class1**. Позже, можно будет надеть шаблонов с какими-нибудь полезными файлами заготовками, но я пока не могу представить, что такого универсального может понадобиться всегда в новом проекте.

Следующим шагом будет настройка информации о текущей сборке проекта. Для этого открываем свойства проекта (Клик правой кнопкой по проекту > Properties) и на первой же закладке видим такую информацию:



Configuration: N/A Platform: N/A

Assembly name: softblog_violet-tape_template Default namespace: softblog_violet_tape_template

Target Framework: .NET Framework 3.5 Output type: Class Library

Client-only Framework subset

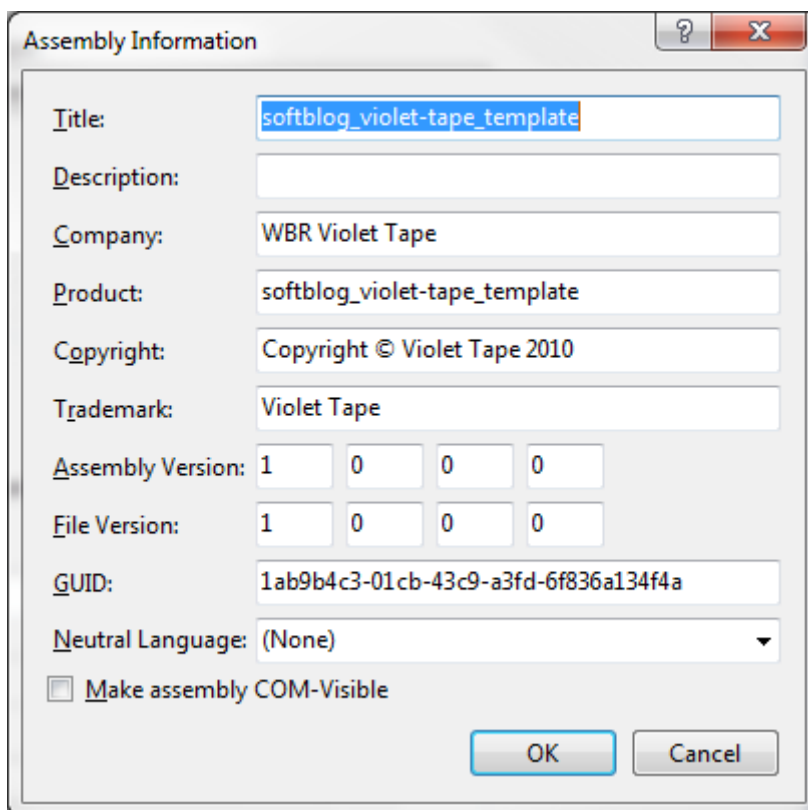
Startup object: (Not set) Assembly Information...

Resources

Specify how application resources will be managed:

[Link resources and manifest](#)

Можно выставить используемую версию .Net платформы, имя сборки, пространство имен, и нужная нам кнопка «**Assembly Information...**». Нажатие на нее вызовет к жизни новое диалоговое окно, где можно указать информацию которая отличить ваш MyClass от аналогичной сборки Васи Пупкина. ;) На этом вкладке есть простор для того, чтобы потешить свое самолюбие!



Assembly Information

Title: softblog_violet-tape_template

Description:

Company: WBR Violet Tape

Product: softblog_violet-tape_template

Copyright: Copyright © Violet Tape 2010

Trademark: Violet Tape

Assembly Version: 1 0 0 0

File Version: 1 0 0 0

GUID: 1ab9b4c3-01cb-43c9-a3fd-6f836a134f4a

Neutral Language: (None)

Make assembly COM-Visible

OK Cancel

После того, как закончили выдумывать имена своих будущих корпораций (тут вспомним поговорку про плохого солдата), можно настроить и более насыщенные темы. Переключаемся на

следующую по порядку закладку в свойствах и попадаем на настройки сборки проекта (закладка **Build**).

The image shows the 'Configuration' section of the Visual Studio Build Properties dialog box. At the top, there are two dropdown menus: 'Configuration:' set to 'Debug' and 'Platform:' set to 'Active (Any CPU)'. Below these are several sections of settings:

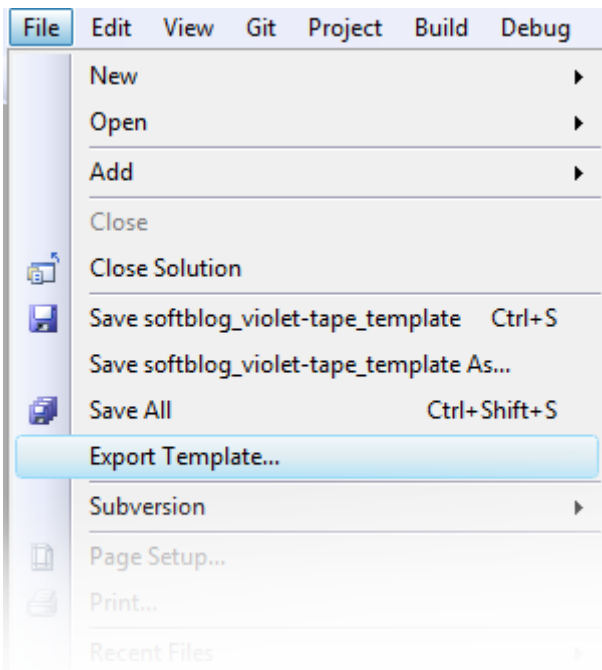
- Allow unsafe code:**
- Optimize code:**
- Errors and warnings:**
 - Warning level:** 4
 - Suppress warnings:** (empty text box)
- Treat warnings as errors:**
 - None
 - Specific warnings: (empty text box)
 - All
- Output:**
 - Output path:** ..\Output\Debug\ (with a 'Browse...' button)
 - XML documentation file: (empty text box)
 - Register for COM interop
 - Generate serialization assembly:** Auto

An 'Advanced...' button is located at the bottom right of the dialog.

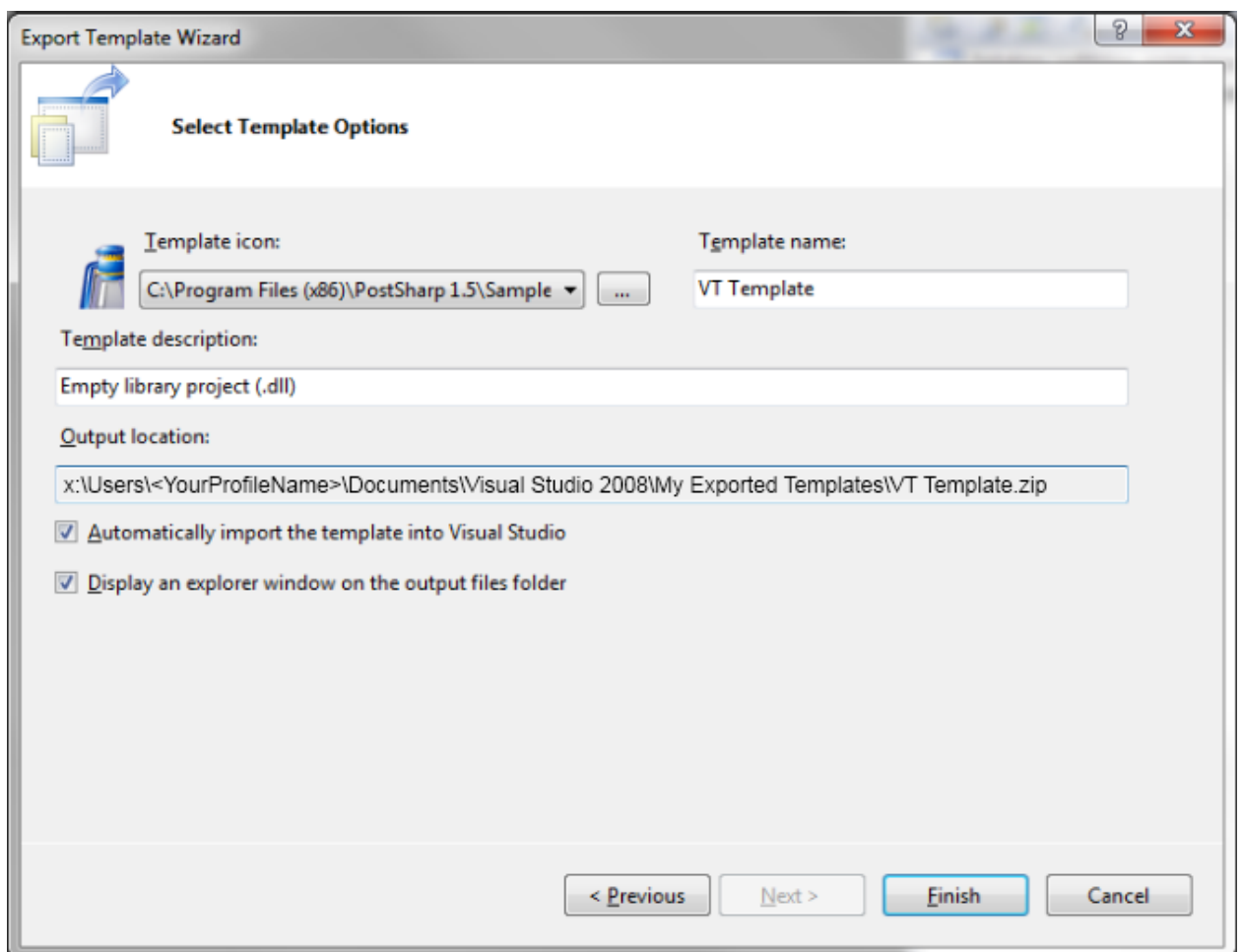
На данной закладке меня обычно интересовало то, куда складывается скомпилированный код. Мне нравится собирать и видеть его в одной папке в корне проекта. Для этого находим секцию **Output** и поле для ввода обозначенное как «**Output path:**». Можно сразу написать относительный путь до папки куда складывать готовые модули, а можно найти эту папку с помощью кнопки «**Browse...**». В этом случае студия покажет полный путь, но не волнуйтесь, при переоткрытии этой закладки вы увидите относительный путь от файла проекта.

Поле «**Output path:**» надо заполнить 2 раза: один раз для *debug* сборки, и еще раз для *release* сборки. Переключатся между режимами позволяет выпадающий список, обозначенный как «**Configuration**».

Для простейшего случая на этом настройку проекта можно закончить и перейти к экспорту проекта. Если вам надо что-либо еще (ссылки на другие проекты, специальные файлы в проекте), то настройте это и переходите к экспорту шаблона.



В основном меню, в опциях работы с файлами находим пункт «**Export Template...**», жмем и видим начало помощника по экспорту шаблонов файлов/проектов.

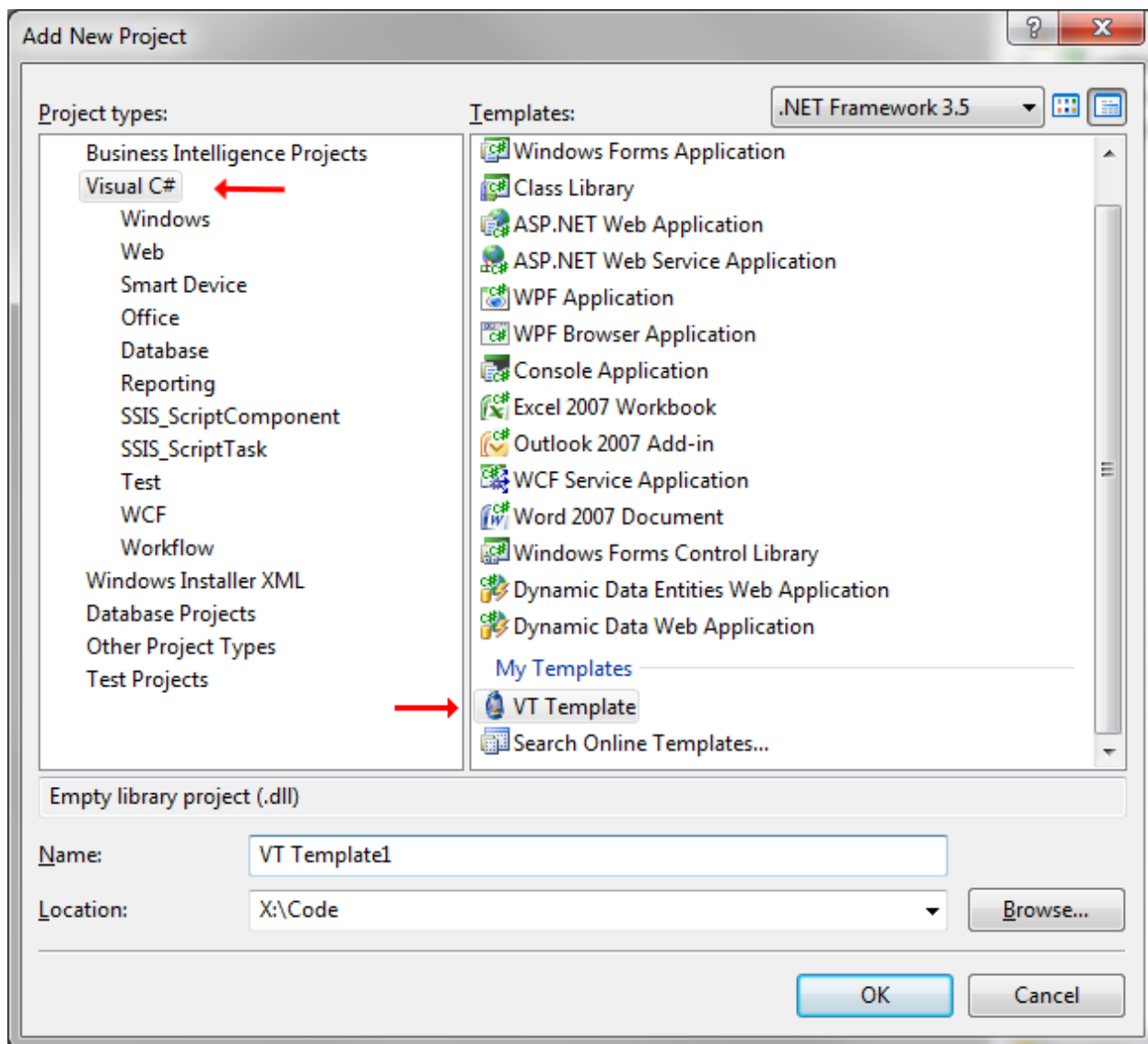


Для шаблона можно указать иконку, по которой вы сможете быстро узнать свой проект при добавлении его к решению или при создании нового чего-то. Задаете имя, чтобы не было путаницы и описание, чтобы исключить малейшие сомнения в назначении проекта.

Помощник экспорта шаблонов подскажет вам, где искать шаблон, для передачи его друзьям и коллегам, бэкапа или препарации. Далее я расскажу как можно будет установить шаблон, при переустановке студии и других несчастьях.

Не снимаем галочку с первой опции «*Automatically import the template into Visual Studio*» и можем снять со второй «*Display an explorer window on the output files folder*».

При нажатии на кнопку **Finish** работа по экспорту шаблона завершена и вы можете лицезреть результат своих трудов в диалоговом окне «*Add New Project*».



По умолчанию все шаблоны будут видимы только когда в левой части выбран пункт **Visual C#**, как настроить появление шаблонов в тематических папках будет чуть позже по тексту.

Visual Studio 2010

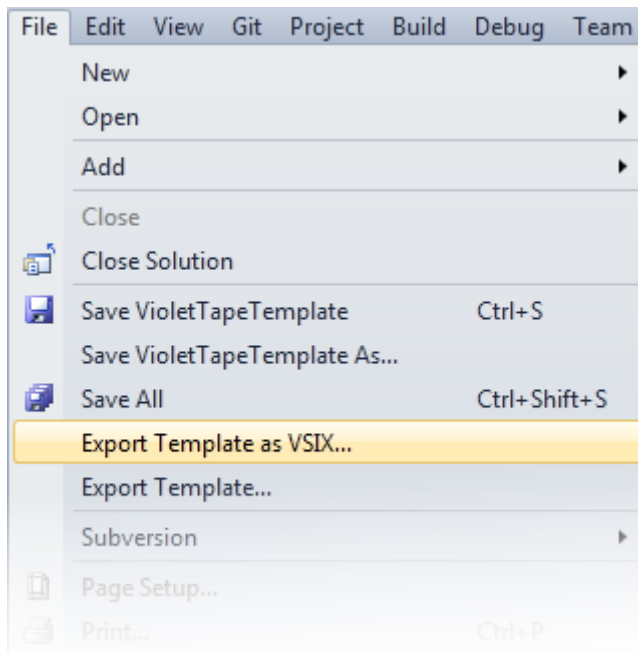
Все вышеописанные действия работают и для 2010 студии. Но в ней есть и еще один способ экспорта шаблонов проектов – в виде расширений студии, которые более гибкие в настройке. Для более того, чтобы осуществить такой фокус, необходимо скачать и установить [Export Template Wizard](http://visualstudiogallery.msdn.microsoft.com/en-us/57320b20-34a2-42e4-b97e-e615c71aca24) (<http://visualstudiogallery.msdn.microsoft.com/en-us/57320b20-34a2-42e4-b97e-e615c71aca24>)

Так же в арсенале полезно будет иметь [VSIX Explorer](http://visualstudiogallery.msdn.microsoft.com/en-us/e3d76792-d1b3-4d0c-9716-1f16c38be8b3)

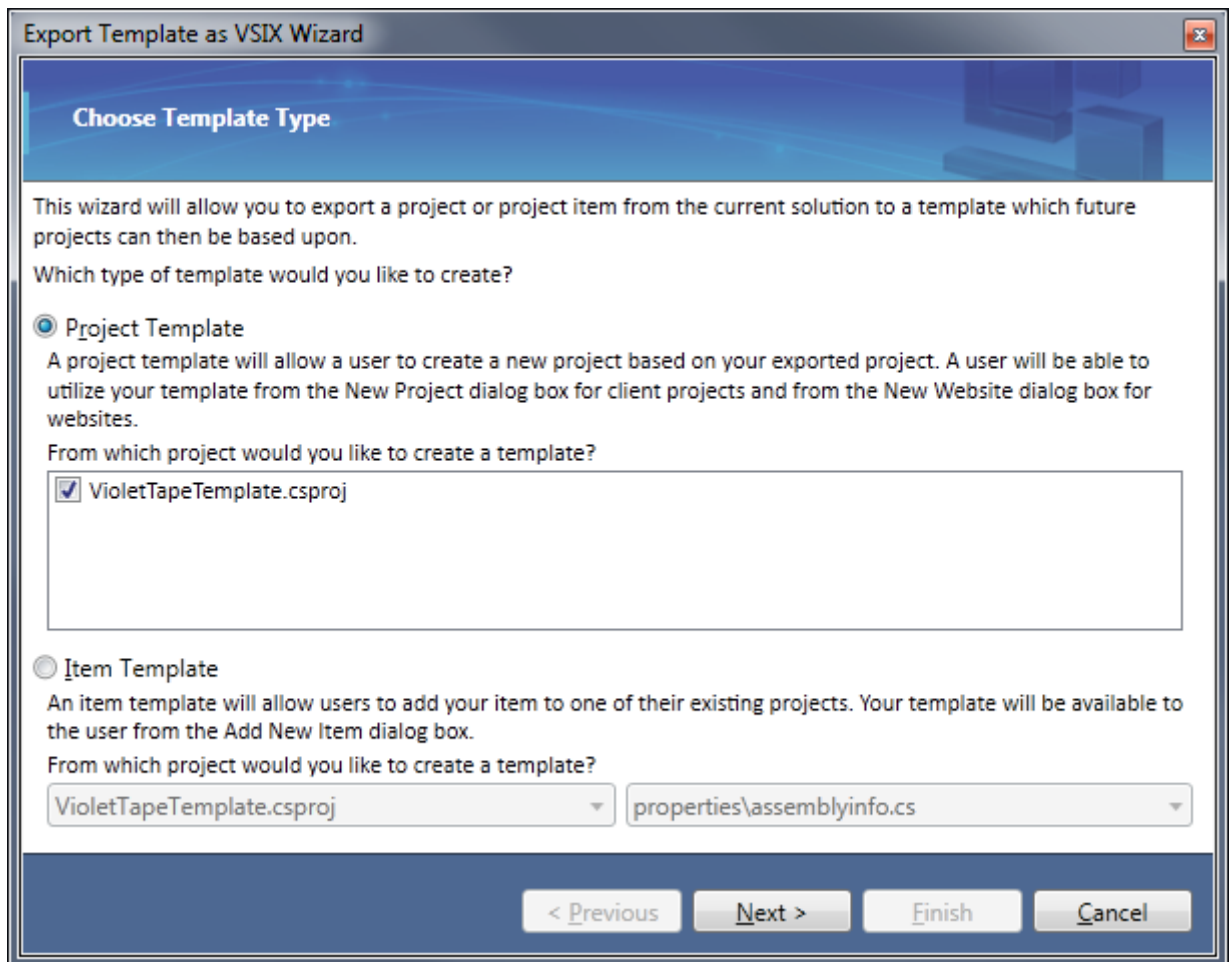
(<http://visualstudiogallery.msdn.microsoft.com/en-us/e3d76792-d1b3-4d0c-9716-1f16c38be8b3>

), если его по какой-то причине не оказалось в вашей студии. Проверить можно, посмотрев на элементы в меню **Tools**.

После того, как вы настроили проект для своих нужд, открываем основное меню на пункте *File* и видим новую опцию «**Export Template as VSIX...**».



Появляется помощник создания шаблонов в формате VSIX, местами сильно повторяющий оригинальный.



Выбираем проект для экспорта и ждем далее, тут ничего интересного нет.

На следующем шаге помощника мы видим новые опции и возможности. Можно задать красивую картинку с логотипом или диаграммой, чтобы производить впечатление на неофитов или на самого себя. Последние два поля намекают на новые интересные возможности при создании проекта, но я пока с ними не разобрался, скажу честно. Как разберусь и придумаю им полезное применение, так сразу расскажу. Наверно можно будет сделать так, чтобы при создании проекта для запуска тестов, было возможно указать ссылки на сборки, которые будут участвовать в тестировании.

The screenshot shows a Windows-style dialog box titled "Export Template as VSIX Wizard" with a close button in the top right corner. The main area has a blue header with the text "Select Template Options". Below the header are several input fields and buttons:

- Template Name:** A text box containing "VioletTapeTemplate".
- Template Description:** A text box containing "Violet Tape lib with PostSharp referene".
- Icon Image:** A text box containing "X:\<source>\Icon.png" with a red border. To its right is a "Browse..." button.
- Preview Image:** A text box containing "X:\<source>\PreviewImage.png" with a red border. To its right is a "Browse..." button.
- Wizard Assembly:** An empty text box with a "Browse..." button to its right.
- Welcome page:** An empty text box with a "Browse..." button to its right.

At the bottom of the dialog, there are four buttons: "< Previous", "Next >", "Finish", and "Cancel".

На следующем экране вы увидите информацию, которая полностью относится к новому виду расширений для студии:

- Id пакета расширения, чтобы на него могли ссылаться другие расширения
- Лицензии для использования
- Ссылка на руководства по использованию или настройке
- Поддержку для каких вариантов студии включить

А так же виднеется часть опций из стандартного помощника по экспорту.

The screenshot shows the 'Export Template as VSIX Wizard' dialog box, specifically the 'Select VSIX Options' step. The dialog has a blue header with the title 'Export Template as VSIX Wizard' and a close button. Below the header, the title 'Select VSIX Options' is displayed. The main area contains several input fields and checkboxes:

- VSIX Id:** VioletTapeTemplate.d046a014-659c-4078-a65f-86e1990aac33
- Product Name:** VioletTapeTemplate
- Company Name:** Violet Tape
- Version:** 1.0
- License:** (empty field) with a 'Browse...' button.
- Getting started guide URL:** (empty field)
- Supported VS Editions:** VS 10.0: Professional; with a 'Select...' button.
- Output Location:** C:\Users\...\.Documents\Visual Studio 2010\My Exported Templates\VioletTapeTemplate.vsix
- Automatically import the template into Visual Studio
- Display an explorer window on the output files folder

At the bottom, there are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

На этом шаге можно указать, что наш шаблон будет совместим со всеми редакциями студии 2010, для чего жмем на кнопку «**Select...**» рядом с полем **Supported VS Editions**, после чего появляется новое окошко, где ставим галочки напротив всех пунктов и подтверждаем свой выбор.

The screenshot shows the 'Select Editions' dialog box, which appears after clicking the 'Select...' button in the previous dialog. It contains a list of Visual Studio editions with checkboxes:

- Visual Studio Ultimate
- Visual Studio Premium
- Visual Studio Professional
- All Express Editions
 - Visual Basic Express
 - Visual C# Express
 - Visual C++ Express
 - Visual Web Developer Express

At the bottom, there are two buttons: 'Accept' and 'Cancel'.

После всех манипуляций можно жать на Finish и искать свой шаблон проекта в галерее установленных проектов!

Дальнейшие улучшения

Через некоторое время мне надоело, что все мои шаблоны лежат в одной куче под главным пунктом **Visual C#** в добавлении проектов, а не по категориям: Windows, Test, DataBase и т.д. К сожалению, стандартных средств, для того, чтобы переместить проект в другую категорию нет. Но можно это все сделать руками!

Берем свой шаблон проекта и идем по адресу **C:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\IDE\ProjectTemplates\CSharp** там находим нужную категорию, открываем ее, и добираемся до папки, где лежат zip архивы и помещаем туда свой шаблон. Перезапускаем студию и наблюдаем результат.

Если результата нет, то надо поудалять кэши проетов. Для этого идем по адресу **C:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\IDE\ProjectTemplatesCache** и удаляем файл **cache.bin**. После этого студия снова опросит все свои папки, найдет ваш проект и включит его в соответствующий раздел.

Для студии 2010 все то же самое, за исключением того, что номер студии 10.0.

Установка

Установка происходит очень просто: необходимо скопировать шаблон либо как указано выше, либо в **%Documents%\Visual Studio 2008\Templates\ProjectTemplates\Visual C#** (**%Documents%\Visual Studio 2010\Templates\ProjectTemplates\Visual C#** соответственно).

Для 2010 студии, если вы сделали шаблон проекта в виде расширения студии VSIX, то по двойному клику все само установится.

Как видите особых проблем при установке шаблонов заново возникнуть не должно.

Удаление

А вот это совсем неочевидный момент, как оказалось. В процессе экспериментирования я насоздавал много шаблонов, но вот как их удалять... Быстрой кнопки для удаления шаблонов найти не удалось. Да видимо и нет такой =) Интернет щедро делится информацией о том, как создавать шаблоны, но вот как удалить их из коллекции проектов, почти ни слова. Пришлось все пробовать методом научного тыка.

VS 2008

Удаление из 2008 студии происходит достаточно безболезненно и быстро. Необходимо найти папку **%Documents%\Visual Studio 2008\Templates\ProjectTemplates\Visual C#**, там свой шаблон и просто удалить его, после чего перезапустить студию. Так же желательно прибить кэш проектов. Для этого идем по адресу **C:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\IDE\ProjectTemplatesCache** и удаляем файл **cache.bin**.

Собственно и все!

VS 2010

Если вы создавали шаблон проекта стандартно в виде zip архива, то надо будет повторить все те же шаги, что и для 2008 студии, с поправкой пути на номер версии студии. Если же экспортировали как vsix, то действия будут несколько иными.

Для начала надо пройти по адресу `%user%\AppData\Local\Microsoft\VisualStudio\10.0\Extensions` и там найти свое расширение. Удалить его. Потом надо в обязательном порядке прибить кэш проектных шаблонов. Эта вещь находится по адресу `%user%\AppData\Local\Microsoft\VisualStudio\10.0\ComponentModelCache`. Если вы не пропихивали свой шаблон в категории к предустановленным шаблонам, то все должно исчезнуть. Я этого у себя добился))

Поле для дальнейшего развития

Надо подробнее изучить возможность создания сразу шаблонов решений на несколько проектов за раз. Интересно было бы посмотреть на новые возможности vsix пакетов, как создаются для них помощники создания и все в таком духе.

Вообще я советую вам потренироваться на этой ниве, пригодится!

Hard'n'heavy!