

Ноябрь 2011 №11 (03)



журнал developer.RU.

Разработчикам про технологии Microsoft

## Постоянныe подписчики

**MFCMAPI: траблшутим почтовые ящики MS Exchange Server**

**Шпаргалка по сертификации приложений на Windows Phone 7**

**Обзор библиотеки Kendo UI**

**ASP.NET MVC + jQuery AJAX**

**Управление руководителями проектов**

**Фоторепортаж «День студента Microsoft 2011»**

**Работа с базой данных в WP7 для ленивых**

## Репортаж

Microsoft

**tech·ed**  
Russia | 2011

## Интервью



Интервью с  
Михаилом Черномордиковым  
руководителем направления клиент-  
ских платформ Microsoft Россия



Интервью с  
Дмитрием Сошниковым  
координатором академических  
программ Microsoft Россия



Интервью с  
Яной Курановой  
Microsoft Student Partner

# FastVPS.ru

**VPS от 129 рублей**

- 100 MB RAM
- 300 MHz CPU
- 6 GB HDD



**Dedicated от 2199 рублей**

- Intel® Core™ i7-920 Quad-Core incl
- 8 GB DDR3 RAM
- 2 x 750 GB HDD

- один из лучших дата-центров Германии**
- служба поддержки на русском языке**
- уникальная система восстановления**
- удаленный reboot/reinstall**
- установка сервера в течение 24 часов**

# Заводим блог компании на сайте журнала MSDeveloper.RU

Доброго времени суток!

Друзья, рады вам сообщить, что теперь любая компания, имеющая отношения к технологиям и продуктам Microsoft, может абсолютно бесплатно завести свой блог на сайте журнала MSDeveloper.RU. У нас нет никаких ограничений по созданию и ведению таких тематических блогов компаний. Если вы компания, которая активно использует инструменты и технологии Microsoft в своей повседневной жизни, то мы рады предложить вам свою площадку для публикации технических статей, тематических новостей, анонсов и других событий, подходящих под тематику журнала MSDeveloper.RU.

Все сообщения или посты из блогов компаний, равно как и сообщений из блогов пользователей, без учета какого бы то ни было рейтинга, автоматически попадают на главную страницу сайта. У нас на проекте отсутствует рейтинг на сообщения, и мы не проводим специальную выборку для главной страницы. Все сообщения будут опубликованы на главной странице сайта, а также автоматически попадут в наши социальные сети и основной RSS-канал. Иначе говоря, будут доступны всем нашим читателям без исключения. Думается такой подход в формировании ленты статей и новостей исключает возможность разных спекуляций и выпрашивания «плюсануть мой пост». Каждое сообщение будет услышано!



Напоследок хочется привести небольшой список компаний и команд, которые уже с нами. Это:

[блог команды русского MSDN](#),  
[студенческий блог Microsoft Russia](#),  
[блог команды Xbox 360 Россия](#),  
[хостинговая компания Infobox](#),  
[облачный провайдер Parking.ru](#),  
[блог команды Windows Phone 7 Rocks!](#)  
[блог команды PVS-Studio](#),  
[блог команды w7phone.ru](#)

И многие, многие другие не менее интересные компании и команды. Кроме этого мы ожидаем в ближайшее время приход еще ряда серьезных компаний, которые дадут много интересной и свежей информации, что будет интересно всем нашим читателям. Приглашаем компании, а также отдельных разработчиков создавать и вести свои блоги на сайте журнала MSDeveloper.RU.

Приятного Вам времяпрепровождения в компании журнала MSDeveloper.RU!

Как создать  
персональный блог

Как создать блог  
компании

Хотите быть в курсе всех событий?  
 Подпишитесь на нашу RSS-ленту новостей, Twitter, ВКонтакте  
 или страницу в Facebook.  
 Мы вещаем круглые сутки семь дней в неделю



Свежие новости  
в Twitter'e



RSS-лента  
новостей



Свежие новости  
в Facebook'e



Свежие новости  
в ВКонтакте



**Издательская группа**

ООО «Издательство «Стангор»  
[www.stangor.ru](http://www.stangor.ru)  
 Журнал MSDeveloper.RU  
[www.msdeveloper.ru](http://www.msdeveloper.ru)

**Издатель и руководитель проекта**  
 Станислав Горнаков [sg@stangor.ru](mailto:sg@stangor.ru)

**Зам. главного редактора**  
 Игорь Редько [ir@stangor.ru](mailto:ir@stangor.ru)

**Арт-директор**  
 Светлана Петрова  
[art-dis@stangor.ru](mailto:art-dis@stangor.ru)

**Редакторы**  
 Сергей Рубан  
 Ирина Войкова  
 Андрей Коваль  
 Василий Сенявский  
 Марат Ягудин  
 Игорь Войков  
 Игорь Периодов

**Корректоры**  
 Ксения Рубова  
 Максим Веров  
 Роман Ветрин  
 Максим Злобин

**Авторы**  
 Андрей Гордиенков  
 Борис Романов  
 Валерий Воробьев  
 Никита Львов  
 Александр Рагель  
 Сергей Карташев  
 Филипп Панфилов  
 Сергей Рогачев  
 Дмитрий Соловьев  
 Андрей Соловьев  
 Владимир Колесников  
 Станислав Горнаков

**Отдел маркетинга и рекламы**

**Руководитель отдела**  
 Виктор Прудников [pr@stangor.ru](mailto:pr@stangor.ru)

**Реклама в журнале**  
[msd@msdeveloper.ru](mailto:msd@msdeveloper.ru)

**Реклама на сайте журнала**  
[msd@msdeveloper.ru](mailto:msd@msdeveloper.ru)

Издание зарегистрировано в  
 Комитете Российской Федерации  
 по Печати  
 Свидетельство № ФС 77 - 3905

**Журнал издает**  
 ООО «Издательство «Стангор»

**Для пресс-релизов и информации о**  
**пресс-конференциях**  
[msd@msdeveloper.ru](mailto:msd@msdeveloper.ru)

**Авторам**  
[www.msdeveloper.ru/about/autors.aspx](http://www.msdeveloper.ru/about/autors.aspx)

За достоверность рекламной информации  
 ответственность несут рекламируемые. Рекламные материалы не редактируются и  
 не корректируются. Редакция ждет ваших  
 откликов и писем читателей. Фотографии,  
 рукописи и другие печатные материалы не  
 редактируются и не корректируются. При  
 цитировании или перепечатывании материа-  
 лов ссылка на сайт [www.msdeveloper.ru](http://www.msdeveloper.ru) и название журнала MSDeveloper.RU  
 обязательна. Полное или частичное вос-  
 произведение материала журналов воз-  
 можно только с письменного разрешения  
 Издательства Стангор. Мнение редакции  
 журнала может не совпадать с мнением  
 авторов статей публикуемых в журнале.  
 Все товарные знаки принадлежат их вла-  
 дельцам.

**Интервью**

Интервью с Михаилом Черномордиковым  
 Руководителем направления клиентских платформ Microsoft  
 Россия

19

Интервью с Дмитрием Сошниковым  
 Координатором академических программ департамента стратегических технологий Microsoft Россия

37

Интервью с Яной Курановой  
 Microsoft Student Partner

76

**Репортаж**

Резюме докладов на Microsoft Tech•Ed Russia 2011

07

Фоторепортаж Tech•Ed Russia 2011

11

Взгляд из-за кулис на Tech•Ed Russia 2011

15

Фоторепортаж «День студента Microsoft 2011»

69

**Разработка**

Отображение сложносоставных данных в WPF

26

Постоянные подписчики

29

**Microsoft Exchange Server**

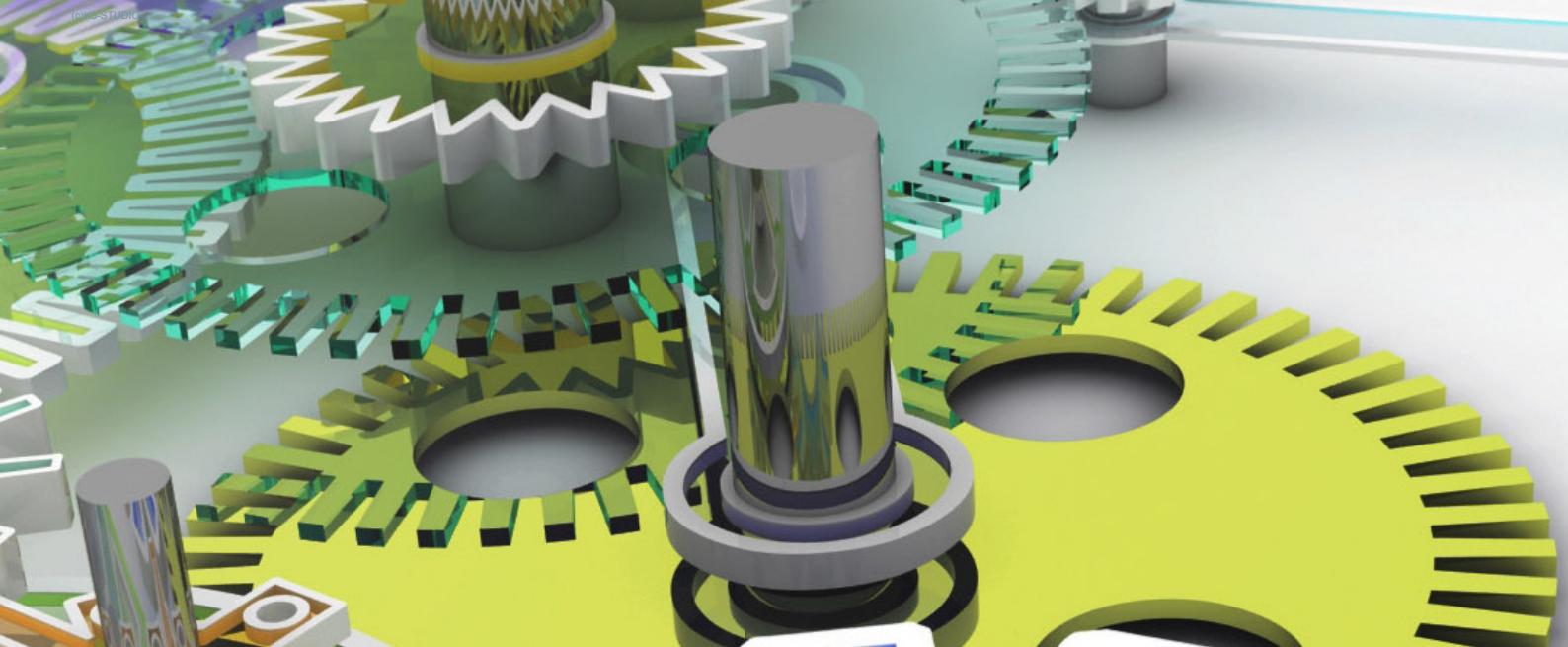
MFCMAPI: траблшутим почтовые ящики MS Exchange Server

33

**Windows Phone 7**

Шпаргалка по сертификации приложений на Windows Phone 7

42



## Kendo UI

Обзор библиотеки Kendo UI

45

## A SP.NET

ASP.NET MVC + jQuery AJAX

53

## Проекты

Управление руководителями проектов

64

## Администрирование

Развертывание операционных систем Windows при помощи флэш-накопителя

79

Работа с базой данных в WP7 для ленивых

82

Использование VHD в OC Windows 7 и Windows Server 2008 R2

85

## Программы

1С.Net:Предприятие  
многопоточный http-сервер с обработкой запросов в коде 1С

88

## Веб-сервисы

Зачем и как можно создать свой персональный блог на сайте журнала MSDeveloper.RU

92



Говорят, что Алёнка Коршунова  
из отдела оплаты даст каждому,  
кто позвонит и скажет ей  
кодовое слово “MSDEVELOPER”...

50% скидку на  
Windows VPS

+7 495 647-20-05 в Москве, +7 812 331-29-99 в Петербурге, 8 800 333-2005 - федеральный  
Акция действует до 31 октября 2011 года включительно.

# Резюме докладов на Microsoft Tech•Ed Russia 2011



## Автор

Сергей Рогачев  
 rogachev@list.ru  
<http://rsn81.wordpress.com>  
 ЗАО «Астерос», Директор R&D

9 -10 ноября в Москве прошла главная конференция Microsoft, посвященная ее технологиям и продуктам – [Microsoft TechEd Russia 2011](#). В России конференция этого международного формата была проведена впервые, заменив локальное мероприятие «Платформа: определяя будущее», которое проходило в течение последних 12 лет. Судя по всему, крупнейшая корпорация пересмотрела свое отношение к России, что не может не радовать.

Помимо непосредственного участия, была возможность наблюдать за происходящим с помощью прямой трансляции. Причем, судя по отзывам в Twitter, качество трансляции [#ruteched](#) было отличным. Если вы не смогли выделить деньги или время на участие в этом событии, не расстраивайтесь, скорее всего, в течение месяца записи докладов опубликуют на [TechDays](#).

Цель и формат данной заметки аналогичны описанию докладов предыдущей крупной конференции Microsoft: [Резюме докладов на Microsoft DevCon'11](#) – я расскажу, какие доклады посетил и что интересного там узнал. Вероятно, это поможет вам выбрать, какие записи докладов стоит посмотреть. Разумеется, все доклады посетить я не смог, так как они одновременно шли по 16 секций. Соответственно, я смогу рассказать только о той 16-ой части докладов, которую выбрал. Вы можете ориентироваться на мои субъективные оценки содержания доклада и мастерства докладчика, которые я указываю ниже после краткого описания содержания доклада.

Кстати, на пленарном докладе озвучили дату проведения [DevCon'12](#), в следующем году конференция Microsoft в России по разработке программного обеспечения пройдет 23-24 мая, регистрация будет открыта уже со следующего месяца. А в курилке случился инсайд о месте проведения – это, как и ранее, московская область, более точно место узнать не удалось.

9 ноября, среда

## Современные тенденции информационной безопасности

Андрей Бешков, Microsoft

Андрей рассказывает про взаимосвязанные тенденции: «консьюмеризацию» и мобилизацию современных технологий в ИТ, и приводит рекомендации по изменению информационных политик, которые могут обеспечить сохранение уровня безопасности на прежнем уровне.

Сотрудники стали не только приносить на работу личные мобильные устройства – телефоны, ноутбуки, планшеты и т. п., но и подключать их тем или иным способом к корпоративной сети. Вследствие этого специалисты по безопасности и администраторы, которые пользуются старыми методами запретов, так или иначе начинают терять управление. Напрашивается разумный вывод: если не можешь победить какое-то движение, остается только его заглушить. К примеру, была интересной информация о том, что в Microsoft сотрудникам не запрещают пользоваться на работе социальными сетями, но регламентируют политику их использования, то есть что и в каком формате можно публиковать. Проблема убытков корпораций из-за использования сотрудниками на работе социальных сетей напомнила мне историю, услышанную на другом мероприятии. На заводе запрет социальных сетей и программ обмена мгновенными сообщениями привел к увольнению по собственному желанию 15% кадрового состава. Проблему удалось решить организацией корпоративной социальной сети с помощью Microsoft Office SharePoint Server 2010.

Сотрудники нуждаются в том или ином общении, достаточно только направить его в необходимое рабочее русло. В обоих случаях сработал подход айки-

до: не можешь остановить движение, возьми управление им на себя и направь его в нужном тебе направлении.

Интересная новость: в Microsoft все-таки решили включить предустановленное антивирусное решение в Windows 8. Насколько понимаю, не очень приятное известие для таких компаний, как Kaspersky Lab. Не все знают, что Microsoft развивает собственный антивирусный продукт – Microsoft Security Essentials, который, кстати, является сейчас самым популярным бесплатным антивирусом: [Free Microsoft Security Essentials Is Most Used Antivirus Worldwide](#). Сейчас Microsoft Security Essentials доступен пользователям через Windows Update, но в качестве необязательного обновления, и соответственно, доходит далеко не до всех. Кроме того проблему защиты пользователей усугубляют OEM-ы, устанавливая на поступающие в продажу устройства временные версии платных антивирусных продуктов. После завершения временного периода антивирус перестает обновлять свои базы, но далеко не все пользователи понимают, что это практически равнозначно удалению антивируса. Соответственно, они спокойно живут в ложном ощущении защищенности от угроз, рано или поздно становясь, например, членами ботнет-сети. Поэтому в Microsoft решили предоставить своим пользователям защиту по умолчанию с включением своего антивирусного решения в свою операционную систему.

Посмотрите доклад, чтобы узнать о прогнозах по мобилизации технологий в России, продуктах Microsoft, которые обеспечивают управление устройствами, а также о готовящихся изменениях в Exchange Active Sync.

Содержание доклада: **5**

Мастерство докладчика: **5 ▶**

## Успеть все! Секреты управления временем для ИТ-руководителей

Алексей Галков, Альберт Аюпов,  
Владимир Еронин, Microsoft

Эйзенхауэр предложил технику управления временем, согласно которой все дела разбиваются на группы в зависимости от их срочности и важности, что позволяет выделить важные и существенные задачи и решить, что делать с остальными.

**1. Важные и срочные.** Неотложные дела, критические ситуации, проекты с «горящим» сроком необходимо выполнять без промедления, при этом пытаться предвидеть и по возможности предотвращать их появление.

**2. Важные и несрочные.** Самая важная категория, в которую попадают все дела, ради которых мы живем: нужно стараться, чтобы наши дневные дела были подчинены этому жизненному проекту.

**3. Неважные и срочные.** Якобы неотложные дела, в действительности таковыми не являющиеся: например, вас просят сделать что-то сейчас и быстро, но это не относится к вашим непосредственным обязанностям.

**4. Неважные и несрочные.** Ежедневные дела, которые практически не делаются вклада в улучшение вашей жизни и не приносят никакого результата. Такие поглотители времени необходимо выбрасывать из своего распорядка дня. Даже если такие дела пытаются представить небольшими радостями нашей жизни, нужно уметь их отличить от задач пункта 2, которые только и могут приносить настоящее удовольствие и радость.

Докладчики рассказывают и демонстрируют, как можно применить технику матрицы Эйзенхауэра с помощью наиболее популярного сейчас средства поступления задач Microsoft Office Outlook.

Ежедневный рабочий процесс включает следующие шаги: разбор непрочитанной почты, выполнение задач на сегодня, контроль выполнения задач в конце дня и просмотр задач на завтра. При этом постоянный контроль имеющихся и вновь поступающих задач опирается на отсутствие непрочитанных сообщений, на так называемом критерии Zero Inbox. При разборе почты рекомендуется руководствоваться принципом 4D: «Do, Defer, Delegate, Delete», а иные варианты приведут к тем или иным проблемам. Что нужно или можно сделать сразу – делать сразу. Если есть необходимость или просто можно, то делегировать выполнение задачи другому. Остальное отложить, прочее – в корзину. Этот процесс по технике матрицы Эйзенхауэра можно проводить путем раскладывания всех новых писем по мере прочтения по соответствующим папкам, устанавливая, кроме того, на письма флагги «К исполнению» для получения своевременных оповещений. Посмотрите в записи доклада рекомендуемые сотрудникам в Microsoft подпапки Inbox и технику работы с ними. Единственное, что мне здесь осталось непонятным: почему для категоризации дел рекомендовали применять ручное раскладывание по папкам, а не установку соответствующей категории на элемент и использование папки поиска, собирающей данные сообщения?

Кроме того, вы узнаете из доклада как применять в Outlook такие инструменты, как правила, условное форматирование, быстрые действия, поисковые папки.

Откровением для меня была демонстрация применения Microsoft Office OneNote для быстрого создания протокола непосредственно во время совещания. Как много вашего времени отьедают совещания, болтовня которых заканчивается тем, что все, начиная от участников, заканчивая организатором встречи, благополучно забывают и суть обсуждения и принятые в рамках него решения? Если много, то посмотрите демонстрацию этого сценария. Странно, что докладчики не дополнили этот сценарий возможностью интеграции почты с порталом Microsoft Office SharePoint, при которой происходит автоматическое размещение протокола совещания в общедоступной (в отличие от почтовых ящиков) папке портала.

Содержание доклада: 5

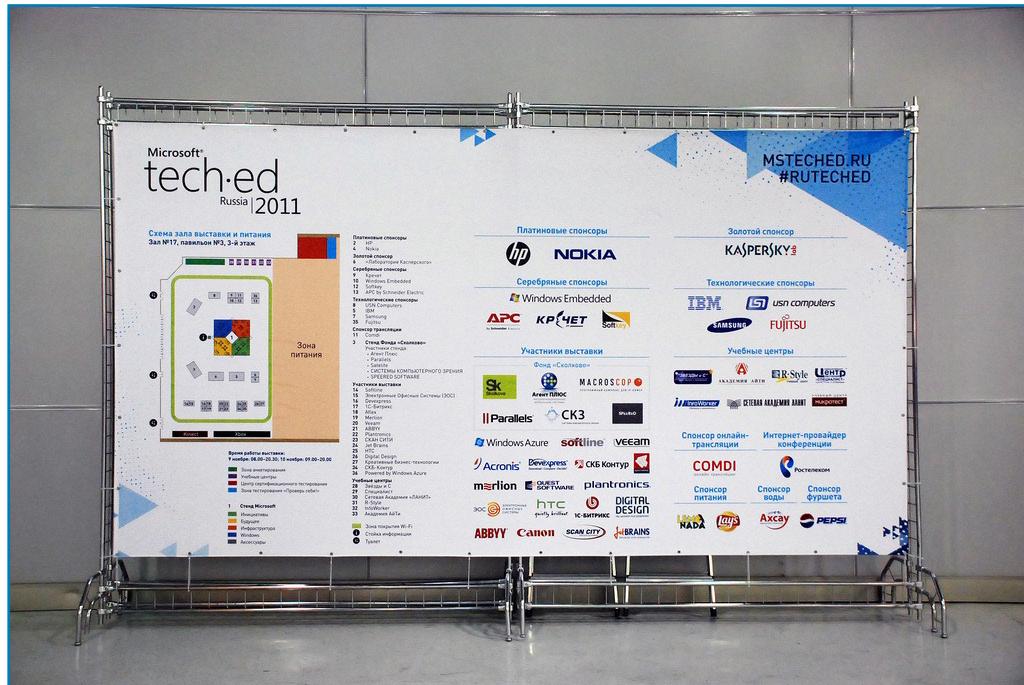
Мастерство докладчика: 4

## Ключевые нововведения в инструментах Visual Studio 11 для управления жизненным циклом приложений (ALM)

Максим Гольдин, Microsoft

Максим демонстрирует новые возможности, которые принесет Microsoft Team Foundation Server 2011 разработчикам, тестировщикам, аналитикам и менеджерам проектов. Видно, что в новой версии сделан упор на методики гибкой разработки и предоставление инструментария всем членам команды разработки.

Аналитики получат инструмент Storyboarding, который представляет собой Add-In для Microsoft Office PowerPoint и ►



позволяет быстро создать макет решения (прототип), обсудить его и собрать отзывы пользователей.

Разработчики получат обновленный Team Explorer (новый интерфейс, упрощающий работу с элементами), штатную поддержку рецензирования кода (code review), поддержку различных платформ модульного тестирования, средство поиска дублирования кода и прочее.

Отличная новость, что менеджеры проектов смогут теперь полноценно работать, пользуясь только веб-интерфейсом. При этом добавленный функционал позволяет проводить планирование спринтов и отслеживать загрузку ресурсов, что раньше можно было сделать, только используя дополнительно Microsoft Office Project или Excel. Там же появилось представление задач в виде доски с бумажками, наклеенными в области доски, соответствующей статусам прогресса выполнения задач: печальная судьба продуктов сторонних поставщиков, которые сейчас представляют такой функционал поверх Microsoft Team Foundation Server 2010.

*Содержание доклада: 4*

*Мастерство докладчика: 3*

## Технологии идентификации на основе заявок (claims-based identity)

Дэвид Чеппел, Chappel & Associates

Необычно, что доклад по такой нетривиальной теме определили на конец дня, когда слушатели уже перегружены информацией. Но талант рассказчика все компенсировал: сколько лет я читал материалы по федеративной безопасности и управления доступом на основе заявок в журналах Architects Journal и MSDN Magazine, слушал доклады по этой теме на различных конференциях и встречах Клуба архитекторов в Microsoft, но впервые увидел, что такой относительно сложный вопрос, оказывается, можно доходчиво осветить всего за час. Под занавес Дэвид иронично пошутил: «Учтите, что на самом деле мир не такой простой, как это может показаться из моих рассказов о нем».

Примерно половину времени Дэвид потратил на освещение теоретических основ идентификации на основе заявок:

1. Определения идентификации, аутентификации и авторизации.
2. Проблемы разрозненной идентификации.
3. Принцип, достоинства, и основные определения технологии идентификации на основе заявок: маркер (удосто-

верение, token), заявки (утверждения, claims), сервис выпуска удостоверений (Security Token Service), поставщик идентификации (Identity Provider) и прочие.

4. Дополнительные варианты использования заявок, к примеру, для определения прав доступа, групп или ролей.

После этого Дэвид показал основные сценарии идентификации на основе заявок в корпоративной сети Windows и облачной инфраструктуре Windows Azure, а также федеративной идентификации, и в конце перечислил продукты Microsoft, которые обеспечивают данные сценарии.

Если тема федеративной безопасности и управления доступом на основе заявок для вас нова, посмотрите запись доклада – это будет легким стартом. Если же вы искушенный в данной теме специалист, посмотрите, какими простыми словами можно объяснить суть этой технологии, например, вашим коллегам.

*Содержание доклада: 5*

*Мастерство докладчика: 5+*

10 ноября, четверг

## Построение корпоративной системы управления проектами на основе Project 2010

Андрей Стуленков, Юрий Звягинцев, Microsoft

Вначале доклада освещаются общие вопросы корпоративных систем управления проектами: какие компоненты они включают, и какую пользу приносит их внедрение. Затем уже описываются возможности Microsoft Project Server 2010 по управлению финансами, ресурсами, программами и портфелем проектов, отчетности и совместной работе над проектом. Сотрудники Microsoft похвастались тем, что в этом году, по мнению аналитиков Gartner, система Microsoft Project Server 2010 обошла остальные приложения для управления проектами и портфелями (Project and Portfolio Management) и принесла Microsoft статус «Strong Positive».

Во второй половине доклада были

рассмотрены сценарии использования «от простого к сложному». В простейшем случае можно ограничиться использованием Microsoft Office Project Professional, совместную работу над проектом может обеспечить внедрение Microsoft Office SharePoint и наиболее функциональный вариант приносит внедрение Microsoft Project Server. Для последнего варианта приводится рекомендация Microsoft последовательность поэтапного внедрения функциональных групп продукта. Здесь я нашел возможный ответ на вопрос, почему в свое время у нас провалилось внедрение расписаний (так любимые всеми timesheets) на основе Microsoft Project Server 2010. Тогда не был завершен первый этап внедрения коробочного решения, по итогам которого должно начаться управление сроками, назначение ресурсов на работы и контроль исполнения, сформирован реестр проектов и хранилище документов, должна была появиться стандартная отчетность. Внедрение же управления расписанием, заявками, ресурсами, проблемами и рисками, портфелем проектов, финансами и т. п. идет только вторым этапом, который опирается на достижения первого этапа.

Среди нового функционала Microsoft Project Server 2010 отметил следующие интересные возможности:

- полнофункциональный веб-интерфейс;
- департаменты, которые могут иметь собственный реестр проектов и ресурсы;
- сценарии, управляющие жизненным циклом проекта;
- делегирование полномочий.

В конце докладчики рассказали про опыт внедрения Microsoft Project Server 2010 в Kaspersky Lab. Хорошо, что в конце повествования было уточнено, что внедрение некоснулось подразделения Research and Development, которое, к слову, в Kaspersky Lab является самым крупным. А то я, как бывший сотрудник Kaspersky Lab, был вначале очень обескуражен этой информацией: в R&D там внедрена иная система PPM.

*Содержание доклада: 4*

*Мастерство докладчика: 4 ►*

	<i>Identity Provider STS</i>	<i>Federation Provider STS</i>	<i>Identity Library</i>
<i>Cloud</i>	Windows Live ID	Windows Azure AppFabric Access Control	Windows Identity Foundation
<i>On-premises</i>	Active Directory Federation Services 2.0	Active Directory Federation Services 2.0	Windows Identity Foundation

## Подходы облачного проектирования в Windows Azure

Гайдар Магдануров, Microsoft

Стандартный доклад про сценарии использования облачной инфраструктуры для разработчиков, который я слышал уже множество раз:

- временем график использования собственных вычислительных ресурсов, показывающий неэффективность их использования из-за неизбежных простое;
- 4 возможных сценария аренды вычислительных ресурсов на основании характеристики необходимой нагрузки;
- описание технологической реализации облачной инфраструктуры, в частности Windows Azure;
- требования к архитектуре приложений, работающих в облачной инфраструктуре: слабая связанность, масштабируемость, отказоустойчивость, параллелизм и сохранение целостности данных;

сценарии последовательной миграции сервисов в облачную инфраструктуру Windows Azure: размещение данных в облаке, размещение фоновой обработки в облаке и полный перенос приложения в облако.

Но если ранее такой объем информации давался в течение продолжительного семинара, то Гайдар уложился всего в час, при этом не пострадал ни объем, ни уровень детализации вопросов. И как всегда, Гайдар разбавил все серьезные темы юмором.

Если вы уже знакомы с технологической стороной Windows Azure, скорее

всего, вы все равно найдете для себя интересную информацию в конце доклада, где освещаются особенности и рекомендации по миграции: данных (горизонтальное, вертикальное и гибридное разбиение данных), фоновой обработки (решение проблем повторной обработки и «ядовитых» сообщений), а также рекомендации по оптимизации использования ресурсов), приложений (карусельная балансировка, работа состояния представлений ASP.NET Web-Forms, рекомендации по обеспечению мультитенантности).

Содержание доклада: 4

Мастерство докладчика: 5

## Управление проектами разработки программного обеспечения на платформе Microsoft Project Server и Visual Studio Team Foundation Server

Елена Осипова, Microsoft

Елена повторила доклад, который я уже слышал от нее же год назад на конференции Microsoft «Платформа: определя будущее». К сожалению, мои ожидания узнать что-то новое по этой теме не оправдались: мне показалось, что даже слайды презентации были ровно те же, что и год назад. Разве что я узнал, где скачать виртуальную машину, демонстрирующую интеграцию Microsoft Team Foundation Server 2010 и Microsoft Project Server 2010.

Содержание доклада: 3

Мастерство докладчика: 3

## Архитектурные особенности создания и миграции приложений для Windows Azure

Алексей Федоров, Гайдар Магдануров, Дмитрий Мартынов, Microsoft

На круглом столе эксперты отвечали на вопросы аудитории по ценообразованию Windows Azure и использованию его сервисов Data Sync и AppFabric (Service Bus, Caching Service). Как видите, обсуждение вышло за рамки темы, озвученной в названии круглого стола: сейчас уже разработчикам интересны не столько вопросы планирования миграции, сколько вопросы «правильной жизни» в облачной инфраструктуре Windows Azure.

На круглом столе «Почему я завтра начну использовать Облако» конференции Microsoft DevCon'11, которая проводилась в начале этого года, также поднимался вопрос ценообразования Windows Azure. Но если тогда все члены круглого стола (кстати, там были Алексей и Дмитрий) согласились, что ценообразование Windows Azure пока очень запутанно и туманно, то сейчас Алексей показал, что появились вполне конкретные ответы на вопрос: «А сколько это будет стоить?». Во-первых, в блоге группы российского представительства Microsoft по работе с компаниями-разработчиками недавно появилась статья с описанием схемы ценообразования при использовании платформы Windows Azure: [Windows Azure. Ценообразование](#). Во-вторых, можно получить конкретные цифры, воспользовавшись калькулятором стоимости использования платформы Windows Azure для вашего сценария: [Windows Azure Pricing Calculator](#).

Содержание доклада: 4

Мастерство докладчика: 5 ■



# Фоторепортаж Tech•Ed Russia 2011

**Автор**

Станислав Горнаков  
@Gornakov  
gornakov@hotmail.com  
www.gornakov.com



Первым на сцену в этот день с вступительным словом вышел Николай Прянишников, президент Microsoft Россия. В коротком вступительном слове он рассказал о будущем конференции Tech•Ed Russia 2011, после чего пригласил на сцену главного операционного директора Microsoft Кевина Тернера.



Кевин Тернер, рассказал про большую партнерскую сеть и огромное количество реализованных стартапов в России. В своем выступлении Кевин затронул огромный спектр вопросов: облачные технологии, развитие инфраструктуры, консьюмиризацию информационных технологий, игровое направление, Windows Phone 7 и многое другое.



После выступления Кевина Тернера, на сцену вышел корпоративный вице-президент С. Сомасегара, который в течение полутора часов выдал массу огромной и полезной информации о Windows 7-8, Windows Phone 7.5, инструментах разработчиков и многом другом. Кроме всего прочего Сомасегара сообщил о старте продаж в России смартфонов Nokia под управлением Windows Phone 7.5 с 10 ноября.

## Конференция



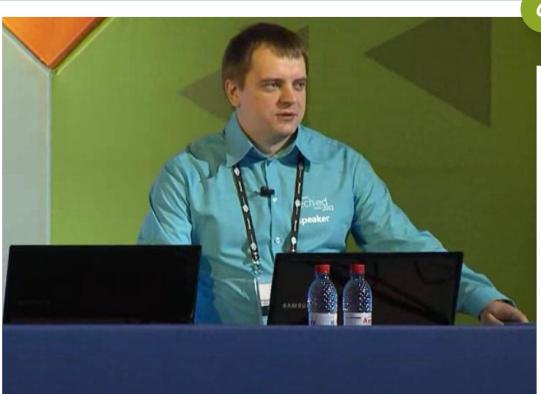
4

В процессе своего доклада Сомасегара пригласил на сцену Михаила Черномордикова, руководителя направления клиентских программ Microsoft Россия. Михаил кратко рассказал о новых возможностях Visual Studio 2011, буквально на коленке, создав и запустив показательный пример в среде Windows 8.



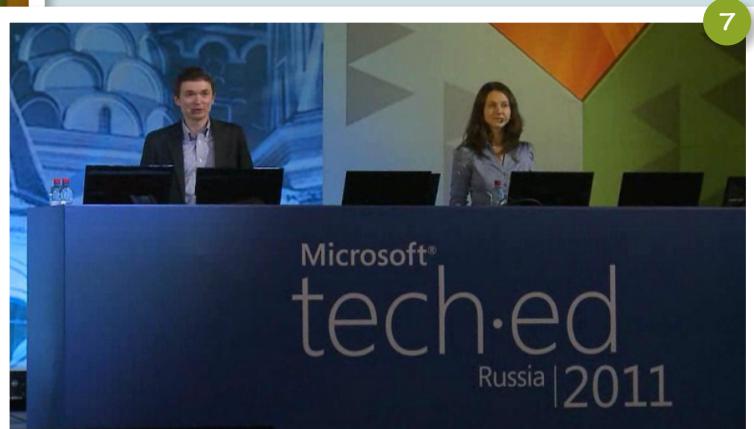
5

Под конец своего выступления Сомасегара пригласил на сцену Дмитрия Андреева, который рассказал много интересного о новом Team Foundation Server.



6

Также Сомасегара за время своего выступления еще много раз приглашал на сцену разных участников. Так Александр Шаповалов в своем коротком выступлении рассказал о развертывании простейшей облачной инфраструктуры.



7

А Владимир Колесников и Анна Марьина представили Офис 365, показав удобство и простейшие принципы работы с этим сервисом.



8

В это время фойе комплекса Крокус Сити работал Гайдар Магдануров, который брал интервью у разных людей и участников конференции.



9

## Конференция

10



После перерыва пошла основная серия докладов. В зале А1 выступал Дэвид Чеппелл с докладом «Представление широкого доступа к данным протоколом OData».

11



В зале А2 свой доклад «SQL Server 2010 – новая комплексная платформа управления данными предприятия и повышения эффективности бизнеса» представил Алексей Шуленин.

12



Зал В2 порадовал всех выступлением Метью Бенке с докладом «Экосистема разработчиков Windows Phone».

13



В свою очередь в зале В2 выступали сразу три докладчика: Дмитрий Мартынов, Юрий Ларин и Лутц Зиоб. Ребята представили классный доклад «Облако Microsoft. Индивидуальный подход к глубине и скорости перехода в облако».

14



Дмитрий Сергеев с докладом «System Center AppController 2012: между облаками и землей».

15



Александр Павлов из Российского представительства HP с докладом «Управление электронными архивами с использованием продуктов Microsoft и HP TRIM».

## Конференция



16

Доклад Бари Шиловера «Встречайте новое поколение решений по управлению ИТ-инфраструктурой – System Center 2011» был достаточно интересным и познавательным.



17

После перерыва в зале А1 вновь появился Дэвид Чеппелл с докладом «Что такое платформа-как-сервис (PaaS), и почему это правильный выбор для новых приложений». В этот день Дэвид был словно заряженная батарейка, отработав сразу четыре доклада подряд.



18

Ярослав Помазков со своим докладом «Управление качеством данных на базе SQL Server Master Data Services и Data Quality Services200».



19

В зале В2 выступали Брайан Сюраси и Сергей Члек с докладом «Hyper-V Cloud Fast Track: референсная архитектура частного облака200».



20

В конце этого года должен состояться выход Exchange Server 2010 SP2 и Скотт Шноль выступил с докладом «Что нового в Microsoft Exchange Server 2010 SP2».



21

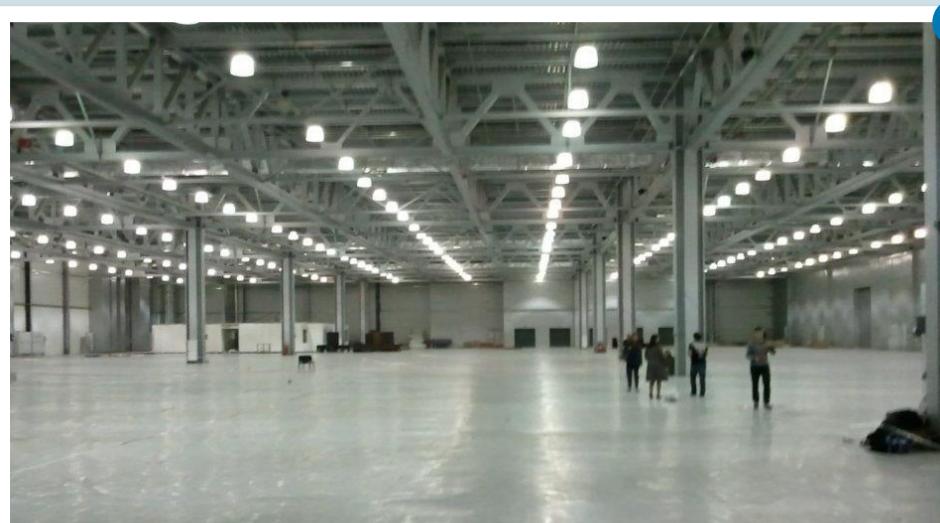
Сергей Копорулин представил интересный доклад «Магазин ИТ-сервисов на базе System Center Service Manager 2012» с демонстрацией процесса создания новых услуг для пользователей и их автоматической обработки с помощью новой линейки продуктов System Center 2012.

# Взгляд из-за кулис на Tech Ed•Russia 2011



## Автор

Владимир Колесников  
Microsoft Россия



Сентябрь месяц. Пустой зал.

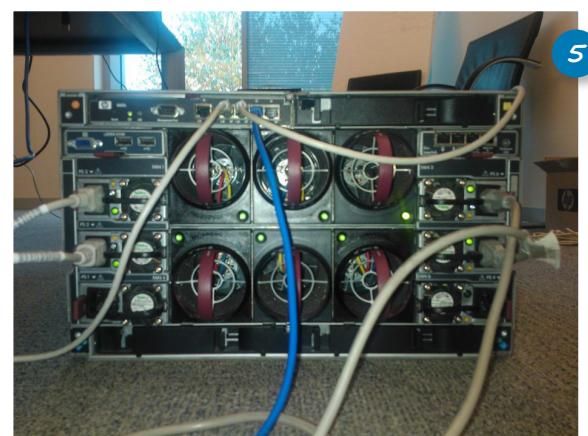
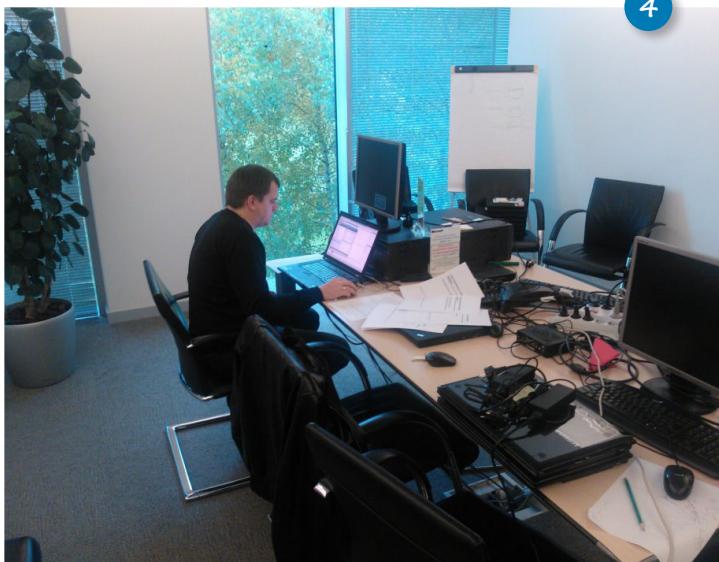


Nokia Lumia 800 на Windows Phone 7 –  
подопытный экземпляр.



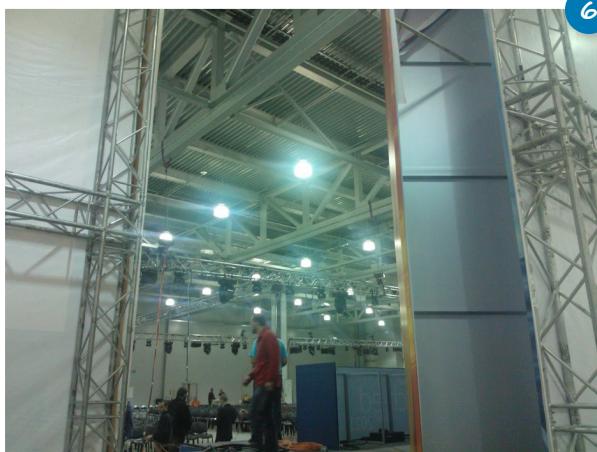
Готовимся.

## Конференция



Железо для демонстрации System Center 2012 и других программ.

Александр Шаповалов еще не знает, что его ждет впереди с этой демой.



7 ноября – монтаж сцены в самом разгаре.



Все выше, и выше, и выше...



Сцена давно готова. Идут тесты демок.

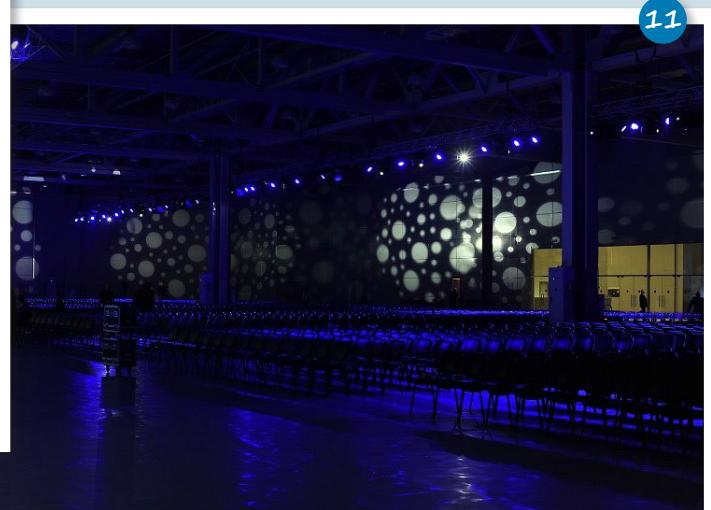


Тестируем, пробуем и решаем проблемы.

## Конференция



На бэкстейдже еще спокойно и малолюдно.



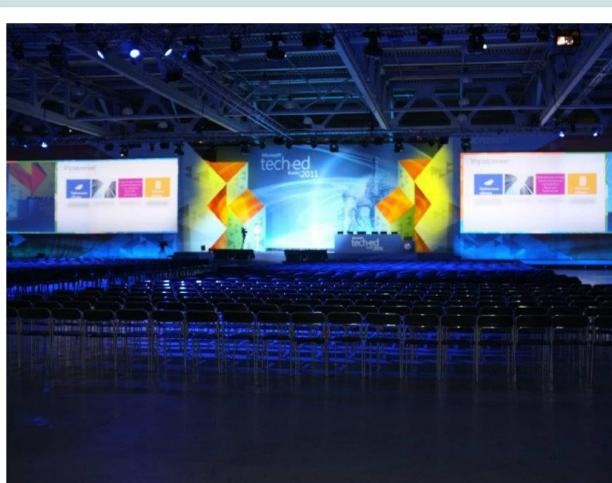
Зал готов и ждет своих первых посетителей.



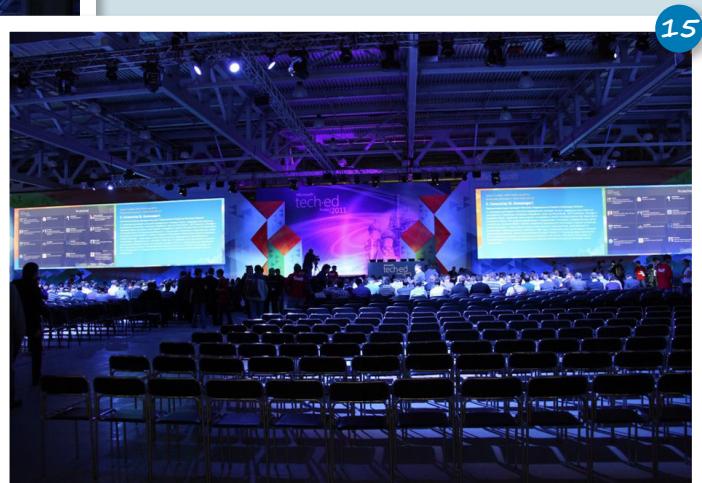
До кейнота примерно 12 часов.



Еще один подопытный экземпляр от Nokia.



Финальный прогон тестов. Участники только-только заходят в Крокус.



Посетители начали рассаживаться, нервы на пределе.

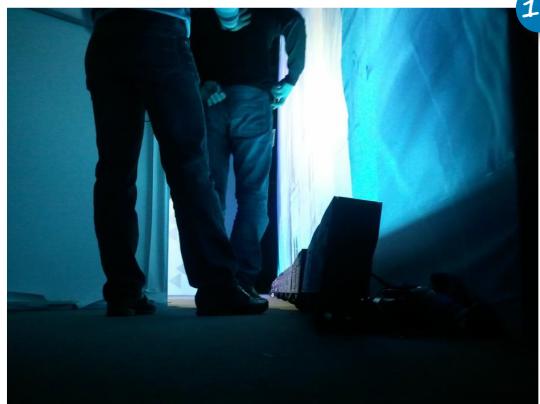
## Конференция



Началось!



На сцене Михаил Черномордиков со своим демо.



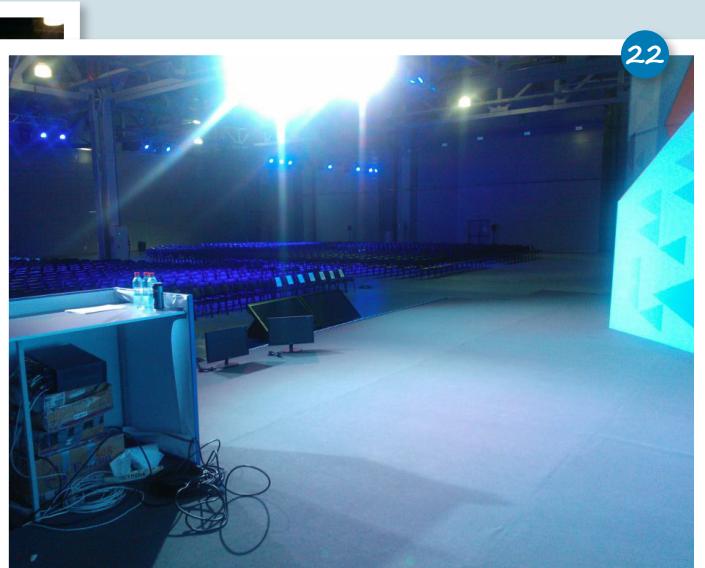
Александр Шаповалов готовится к выходу на сцену.



Я и Анна Марынина готовы выйти на сцену. Это будет последнее демо кейнота. Сомасегару осталось всего 3 слайда.



Вот все и закончилось.



Зал опустел, оборудование разбрали. Всем спасибо!



# Интервью с Михаилом Черномордиковым руководителем направления клиентских платформ Microsoft Россия

Беседовал Станислав Горнаков

**Станислав Горнаков:** Здравствуйте, Михаил. Спасибо за согласие дать интервью и спасибо за то, что вы поддерживаете наш проект всеми силами. Первый вопрос хочется задать по программе Microsoft Academy for College Hires (МАСН), с помощью, который вы смогли прийти в Microsoft. Расскажите, пожалуйста, немного о самой программе, о том, как вам удалось ей воспользоваться и как в итоге «закрепились» на одной из позиций в Microsoft.

**Михаил Черномордиков:** Здравствуйте, Станислав. Спасибо за интересный проект! МАСН – это одна из самых интересных программ в Microsoft. Она позволяет прийти в компанию молодым людям, которые только что окончили университет или вот-вот собираются это сделать. Это международная программа, которая позволяет максимально

быстро погрузиться в принципы и атмосферу Microsoft, получить знания через тренинги и реальную работу. Одновременно с вами в ней участвуют недавние студенты из Европы, Азии, Америки, с которыми вы знакомитесь уже в начале работы в компании. Получается очень дружное сообщество, с участниками которого я с большой радостью я общаюсь до сих пор.

Важно понимать, что МАСН – это, по сути, ключ к двери под названием карьера в Microsoft. Она помогает вам даже без опыта работы, но с нужными знаниями и умениями сразу попасть в одну из самых успешных компаний в мире. В то же время все остальное зависит уже от вас – от того, насколько хорошо вы сможете работать в новых условиях, какой результат принесете и какие проекты реализуете.

Я очень сильно благодарен МАСН за то, что она открыла мне дверь в компанию, где я работаю до сих пор. Так же, как и сотням моих коллег по всему миру.

Кстати, российское подразделение Mi-

crosoft набирает в МАСН каждый год – следите за информацией на [сайте](#).

**Поиск работы в Microsoft было осознанным выбором или рассматривались разные варианты?**

Мой пapa – программист, и мне всегда хотелось идти по его стопам. Компьютером я стал пользоваться с 6 лет (вот такое несчастливое детство), и желание связать свою жизнь с этой профессией появилось очень давно.

Для меня всегда Microsoft являлся безусловным лидером этого рынка, и мне, конечно, хотелось здесь работать. Но я никогда не ставил себе такую конкретную цель, тем более что и уезжать из Мурманска я особенно не собирался. Потом, правда, передумал.

С третьего курса университета я стал студентом-партнером Microsoft (MSP), что несомненно сильно конкретизировало мои дальнейшие жизненные планы.

## Наше досье

В 2006 году окончил с отличием политехнический факультет Мурманского государственного технического университета по специальности «Инженер-программист».

Начал работу с первого курса. Первая должность – системный администратор в туристической компании «Транстур», далее в 2005-06 гг. инженер-программист в Мурманском морском пароходстве.

В 2006 году переехал в Москву и работал разработчиком ПО в выделенном центре разработки корпорации DELL в компании Luxoft.

С января 2007 года по настоящее время – сотрудник корпорации Microsoft в департаменте стратегических технологий:

- 2007 – 2009, Developer Evangelist, эксперт по технологиям разработки ПО
- 2009 – 2011, Architect Evangelist, эксперт по стратегическим технологиям
- 2001 – н. в., Client Evangelism Lead, руководитель направления клиентских платформ

В 2010-2011 гг. работал в штаб-квартире Microsoft в Редмонде, США, в должности Worldwide Field Strategy Manager, менеджером по международной стратегии.

**Переезд из Мурманска в Москву и обустройство на новом рабочем месте дались сложно или особых проблем не возникло?**

Переезжал из Мурманска я на следующий день после получения диплома. Устроился разработчиком в центр разработки DELL в компании Luxoft. Сам поиск работы и собеседования проходили полностью удаленно (кстати, вакансию я нашел простым поиском в «Моем Круге»).

Luxoft сильно помогает при переезде – от встречи поезда до первого аванса для съема квартиры.

Конечно, было непросто работать на полную ставку, до этого я много лет ►

полдня учился. Но в целом и коллектив, и атмосфера международной компании, и интересные задачи помогли войти в нужный ритм достаточно быстро.

Параллельно с этим я участвовал в разных собеседованиях в Microsoft. Мне предложили поехать разработчиком в группу Windows Live в Редмонде, но я понял, что хочу работать в Москве. Об этом выборе не пожалел.



**Когда только пришли в Microsoft, что входило в круг ваших обязанностей и чем конкретно занимались?**

Как мы обсуждали выше, в Microsoft я пришел по программе MACH. Моя должность называлась Developer Evangelist, и первое время я в большей степени был сконцентрирован на работе со студентами и разработчиками. Начались презентации в университетах и «Днях разработчика» – Москва, Самара, Екатеринбург, другие города.

В рамках MACH у нас была серия тренингов Intro School, которая в то время проходила в португальском Лиссабоне.

Являясь новым сотрудником, я помогал моим коллегам в выполнении наших общих целей. В то же время, мне нужно было найти свою нишу, свое направление и специализацию. Ей стала технология Silverlight, которой я впоследствии занималась многие годы. Она как раз была выпущена в 2007 году, когда я пришел в компанию.



**Не так давно вы достаточно продолжительный период времени провели на стажировке в главном офисе в Редмонде США. Это был запланированный визит?**

Да, я вновь решил участвовать в программе для сотрудников, и компания меня здесь сильно поддержала. В те-



чение пяти месяцев я жил и работал в Редмонде, получал новый опыт и делился своим. Это был очень интересный и полезный период моей работы и жизни.

вью с Александром Ложечкиным в прошлом номере.



**Расскажите, пожалуйста, за что именно вы отвечаете в компании теперь.**

Я работал в департаменте маркетинга на должности менеджера по международной стратегии.

Цель поездки была, прежде всего, в получении нового опыта для последующего его применения в российском офисе. Поэтому, в отличие от моих проектов здесь, это была сугубо внутренняя работа, связанная с планированием работы между штаб-квартирами и офисами Microsoft по всему миру, стратегические инициативы, налаживание деловых связей с коллегами в США.

То, что происходило прошлой зимой, помогает мне до сих пор. Это касается и личных связей с коллегами из разных стран, умений и навыков работы, и последних анонсов, связанных с платформой разработки Windows и Windows Phone.

В своей нынешней должности я отвечаю за направление клиентских платформ. Это означает, что мы вместе с коллегами помогаем российским разработчикам быть успешными на таких платформах, как Windows, Windows Phone, Internet Explorer и Kinect. То есть тех платформах, которые работают на клиентской стороне у пользователей, а не на сервере.

Мы работаем со всем спектром разработчиков – от начинающих студентов, которые создают свои первые творения, до крупнейших европейских Интернет-компаний, которыми теперь официально являются Яндекс и Mail.Ru Group.

Наша цель – чтобы российские разработчики создавали самые успешные и известные приложения на нашей платформе для пользователей как в России, так и во всем мире. В этом случае будут счастливы и разработчики, и пользователи, и Microsoft.



**Круг ваших обязанностей на данной позиции – это?**

**С первого июля у вас поменялась должность, до этого вы занимали позицию Microsoft Architect Evangelist, теперь вы являетесь руководителем направления клиентских платформ в Microsoft. Как я понимаю, это должность по-прежнему, что называется, приписана к департаменту DPE (Developers & Platforms Evangelism)?**

Да, я по-прежнему работаю в DPE. Более подробно об устройстве нашего департамента можно почитать в интер-

Прежде всего, это координация работы с коллегами внутри нашего департамента, с другими отделами в российском Microsoft и за его пределами.

Важно понимать, что мы не занимаемся, например, продажей коробок с Windows или маркетингом Windows Phone в магазинах электроники. Этим занимаются коллеги из других департаментов, с которыми мы плотно взаимодействуем. ▶

Также я активно занимаюсь внешней деятельностью – встречаюсь с партнерами, выступаю на мероприятиях, планирую новые совместные решения.

В моей работе мне нравится отсутствие жестких рамок. У нас есть цель, нам нужен результат, а в остальном – свобода творчества. Ничто не мешает завтра придумать что-то совсем новое и заняться этим активно и результативно.

**Вы покрываете достаточно большой круг интересов, успеваете или 48 часов в сутках было бы в самый раз?**

Да, это тоже часть моей работы. При переходе на новую должность сложно отказываться от привычки все делать самому. Но с этим обязательно нужно бороться. И если тебе не хватает времени – значит, ты плохо работаешь.

Кстати, такой же совет мне дали при переезде в Москву на должность разработчика. Важно правильно находить приоритеты и повышать эффективность в любой деятельности. Для этого созданы все условия.

**А можно небольшую зарисовку скажем одного рабочего дня из жизни Михаила Черномордикова в Microsoft Россия?**

Сейчас открою Outlook и посмотрим... У нас свободный график, но я начинаю в 9 утра. Проверка почты, чтение новостей, срочные задачи. Затем впремешку идут встречи с коллегами и решение запланированных задач. Пара звонков партнерам, кому удобнее общаться голосом вместо почты. После обеда две внешние встречи, подобранные так, чтобы между ними можно было быстро проехать. Потом можно



работать из кафе, дома или вернуться в офис.

А еще я конечно Интернетозависим. Поэтому в течение всего дня будут много твитов и записей в Facebook. Утром можно написать что-то интересное на Хабр, почитать профильные ресурсы про Microsoft и наших конкурентов.

**Поговорим об Internet Explorer 9. Вам не кажется, что продукт получился несколько не однозначным, с одной стороны вполне себе аксессуарный и удобный дизайн, а с другой стороны большое количество страниц в Интернете элементарно не корректно отображаются в нем, хотя в других браузерах все нормально.**

Честно скажу, что я очень доволен Internet Explorer 9. Это огромный шаг вперед – причем не только для Microsoft, но для рынка браузеров в целом. Я специально каждую неделю меню браузеры, чтобы следить за тем, что происходит у наших друзей из Google, Mozilla, Apple и Opera. У нас отличный продукт, который, например, на поря-

док быстрее визуализирует графику на HTML5 за счет аппаратного ускорения.

Не могу сказать, что как пользователь часто встречаюсь с некорректным отображением сайтов. Такое действительно возможно, если разработчик пытается «подсунуть» версию сайта, специально сделанную для старых версий IE. Мы всячески пытаемся улучшить эту ситуацию.

При этом встречаются сайты, некорректно работающие и в других браузерах.

Использование HTML5 позволяет разрабатывать решения, которые будут одинаково хорошо отображаться на разных настольных и мобильных браузерах, облегчая жизнь и разработчикам, и пользователям.

**К слову, лично меня достают все уведомления, которые почему-то нельзя отключить, плюс еще мешок достаточно прогнозируемых и ненужных граблей, на которые не стоило наступать, изучив внимательно рынок браузеров. Создается такое впечатление, что команда IE9 прибывает в каком-то вакууме и оторвана от реальных пользователей. Вам так не кажется или все-таки мы сгущаем краски?**

А о каких уведомлениях идет речь? Любые уведомления можно выключить в настройках браузера. Могу предложить, что речь об отключении плагинов. Его можно легко отключить. Но лучше отключить сами плагины, которые в среднем больше всего тормозят работу браузера. Поэтому мы и рекомендуем их отключать, показывая, насколько они мешают быстродействию, если это действительно так.

Что касается процесса разработки браузера, то вы можете следить за ним,



устанавливая промежуточные версии IE Platform Preview. В частности, можно загрузить предварительную версию IE10 на сайте <http://ietestdrive.com>. Там же можно сравнить его работу с другими браузерами.

А по поводу реальных пользователей – посмотрите на долю IE9. Она стремительно растет.



**Вопрос безопасности Internet Explorer 9 особая тема, здесь претензий по сути ноль. Браузер Internet Explorer 9 по праву носит звание самого защищенного браузера. Расскажите, пожалуйста, немного о безопасности IE9 и других не менее полезных функций браузера.**

Да, безопасность в Сети – одна из наиболее важных направлений. К сожалению или к счастью, но количество интернет-пользователей постоянно увеличивается, и к нам присоединяются люди, которые не привыкли настороженно относиться к сомнительным ресурсам.

В Internet Explorer 9 встроен SmartScreen Filter, который предотвращает миллионы атак и блокирует вредоносные сайты. Также, напомню, что именно в IE (тогда еще IE8 Beta) впервые появился разделение вкладок на отдельные процессы в системе, InPrivate-режим и использование технологии DEP/NX. Все это обеспечивает максимально безопасную работу пользователей.



**Кстати, а что с цифрами? Какова доля рынка у IE9 на данный момент?**

Доля IE9 постоянно растет с момента выхода продукта на рынок. В конце октября доля IE9 превысила долю всех остальных браузеров вместе взятых на Windows 7. В США это произошло еще летом.

Почему так важно смотреть именно за пользователями на Windows 7? Потому что все возможности Internet Explorer работают именно в ней. В частности, это аппаратное ускорение, которое реализовано именно благодаря тесной интеграцией с системой.



**Вы также курируете игровое направление Microsoft, а это Xbox 360 и Kinect. Вопросов по игровой приставке Xbox360 лично у меня нет – это лучшая игровая платформа на рынке, я не могу понять, почему люди до сих пор играют на компьютерах. Разница**



в игре на Xbox 360 и ПК, это то же самое, что сравнивать поездку в дождь на машине и велосипеде, очевидно, что на машине поездка в данном случае будет комфортней. Вы сами увлекаетесь играми по-настоящему или играете время от времени?

Я был достаточно активным игроком в школьные годы – это был период первого UFO, первого StarCraft и третьего Mortal Kombat. После этого я долгие годы серьезно не играл.

Сейчас время от времени играю на Xbox 360 и все чаще в новые игрушки на Windows Phone.

Что касается игрового направления, то на самом деле я его не курирую. Просто с недавнего времени мы поддерживаем разработку приложений для Windows, к которым подключен сенсор Kinect. Мы сейчас начинаем помогать разработчикам в создании и поддержке таких решений.



**Kinect – это очень сильный продукт и главное, что это то самое будущее, которое мы все так ждали. Как оцениваете перспективу этого устройства?**

Это действительно революция, которая пришла из чистых исследований и дошла до полки супермаркета. Kinect в первый год существования попал в книгу рекордов Гиннеса, и мне посчастливилось во время его запуска быть в США – он уходил даже быстрее, чем горячие пирожки. Уверен, что это только начало.



**К сожалению сейчас не так много действительно хороших тайтлов под Kinect. В основном это фитнессы и танцы, а хочется каких-то**

мощных франчайзов от больших студий. В чем тут дело? Проблемы с реализацией, технологии еще не те или проблема в нежелании разработчиков?

Посмотрите любую последнюю игровую конференцию – например, E3. Новинок под Kinect все больше и больше. Но здесь важно понимать, что в самом большом восторге от него те, кого принято называть казуальными, а не хардкорными геймерами. Им кстати стал я. Я не хочу несколько часов подряд играть в стрелялку – лучше полчаса погоняться или поиграть в виртуальный теннис.

Это те люди, которые раньше не покупали приставки, а теперь стали. И таких людей очень много!

Означает ли это, что настоящих стрелялок и хитов станет меньше? Конечно нет. Но будут и менее тяжелые игры. И их тоже станет больше.



**Хорошо, перейдем к следующей не менее интересной теме – Windows Phone 7. Продажи на этом этапе, судя по всему не очень большие и порой складывается такое впечатление, что кроме как .NET программистам этот продукт никому не нужен.**

Это не так. Лично я встречаю пользователей Windows Phone и в Америке, и в Европе, а теперь и в России.

Недавно президент подразделения Windows Phone в Microsoft официально заявил, что за первые 12 месяцев продажи смартфонов на Windows Phone превысили продажи Android за аналогичный период. Это очень хороший показатель!

А по прогнозам аналитиков из IDC и Gartner, к 2015 году система Windows Phone займет второе место по доле ►

рынка в мире, опередив iOS.

При удачных продажах Microsoft обычно раскрывает внутреннюю статистку продаж, а вот по Windows Phone 7, если мне не изменяет память какой бы то ни было статистики так и нет. Неужели все так плохо?

Помните, что прошел всего год. Год для абсолютно новой системы. И мы очень довольны тем, что уже за первый год мы смогли с нашими партнерами выпустить целый набор устройств, а количество приложений превысило 30000. Это самый быстрорастущий магазин приложений в истории.

От независимых источников как-то всплыала цифра в 1.7 млн. отгруженных устройств. Цифры хоть какие-то примерные могли бы озвучить по продажам или это сложно?

Во-первых, как я писал выше, я не занимаюсь продажами и продвижением аппаратов. Во-вторых, сами устройства Microsoft не продает – этим занимаются производители и операторы.

Американский *Forbes* недавно написал, что уже треть своих доходов HTC получает именно от аппаратов на Windows Phone. И это за первый год!

В принципе да, год маленький промежуток времени, чтобы судить об успехе или провале платформы, можно вспомнить первый год продаж Андроида, который тоже был не особо удачным. Будем надеяться, что у Windows Phone 7 все еще впереди, тем более что обновление Mango привнесло много нового и интересного. Кстати, появляется разная информация о следующем обновлении Windows Phone 7, прокомментируете что-то по этой теме?

Всегда интересно, когда люди любят заглядывать в будущее. Mango вышло меньше месяца назад всего! У нас запланирован целый цикл обновлений в будущем – следите за новостями.

Вы удовлетворены стартом продаж Windows Phone 7 на российском рынке? Сейчас



в целом ощущается даже недостаток того же HTC 7 Mozart в магазинах.

Мы крайне довольны стартом в России. Все аппараты HTC Mozart раскупили, хотя их количество прогнозировали опытные игроки на нашем рынке – HTC, МТС и Связной. Сейчас уже доступна новая партия, а также HTC Radar и HTC Titan, которые у нас появились сразу же за мировой премьерой.

Уже доступен предзаказ Samsung Omnia W. До конца года появятся Nokia Lumia 710 и Lumia 800 – снова вместе с мировой премьерой.

Российский рынок обещает быть успешным для Windows Phone, и начало действительно удалось!

А как работаете с партнерами в направлении продвижении Windows Phone 7 на рынке? Смотрю много рекламы по телевидению, а например, у нас в Ессентуках купить Windows Phone 7 дело не реальное, был один, да и тот оказался выставочным экземпляром.

Этим активно занимаются мои коллеги, которые взаимодействуют с производителями устройств, операторами связи и ритейлерами. Например, HTC Mozart был запущен вместе с HTC и салонами МТС и «Связной».

Была и реклама по телевидению, и множество биллбордов в Москве и других городах.

Но это не значит, что мы хотим за один день выставить аппарат в каждом городе России. Так вообще ритейл не работает, особенно с абсолютно новым продуктом.

Соглашусь, в отличие от мировых продаж на российском рынке, как кто-то выразился в Твиттере «движуха» пошла, во многом это конечно заслуга евангелистов. Насколько важна сейчас роль евангелистов в IT-отрасли и в Microsoft в частности?

Нам всем хочется верить, что наша работа очень важна. Я уверен, что мы помогаем и разработчикам, и партнерам, и даже конечным пользователям понять все возможности наших продуктов. Лично мне очень приятно, что роль разработчиков в успехе продуктов только возрастает. Это касается и продуктов Microsoft, и нашей индустрии в целом.

Михаил и последний вопрос, ваши планы на будущее, какие активности намечаются, где будете выступать в ноябре и декабре?

Мы сейчас планируем очень много новых активностей по Клиентскому направлению, которые вы сможете увидеть в начале 2012 года. В этом году к нам придут первые смартфоны Nokia на Windows Phone. Один из них я сейчас держу в руках и уверен, что это будет знаковое событие на нашем рынке.

Михаил, спасибо за интервью, очень рад был с вами пообщаться, надеемся на плодотворное сотрудничество с вами.

Спасибо за приглашение! ■

# Октябрьский бюллетень обновлений центра загрузки Microsoft

Предлагаем вашему вниманию ежемесячный список обновлений центра загрузки Microsoft. Весь месяц мы следили за обновлениями, которые регулярно появляются в центре загрузки Microsoft. В каждом новом номере журнала мы будем подытоживать месячный список, и сообщать вам полный перечень поступивших обновлений и других новинок, доступных для скачивания в центре загрузки Microsoft.

- ⇒ [Microsoft® Visual Studio® Agents 11 Developer Preview \(ISO\)](#)
- ⇒ [Windows Embedded Compact 7 Update 3](#)
- ⇒ [Business continuity management for Microsoft SharePoint Server 2010](#)
- ⇒ [Technical reference for Microsoft SharePoint Server 2010](#)
- ⇒ [Remote BLOB storage for Microsoft SharePoint Server 2010](#)
- ⇒ [Microsoft Dynamics CRM Server 2011](#)
- ⇒ [Microsoft Lync for Mac 2011 14.0.1 Update](#)
- ⇒ [Microsoft Lync Server 2010 Resource Kit](#)
- ⇒ [Utilities and SDK for Subsystem for UNIX-based Applications in Microsoft Windows 7 and Windows Server 2008 R2](#)
- ⇒ [MSDN Forum Assistant, TechNet Forum Assistant](#)
- ⇒ [Microsoft Web Farm Framework Version 2.2 for IIS7 \(KB 2627276\)](#)
- ⇒ [Risk and Health Assessment Program for Configuration Manager \(CMRAP\) - Scoping Tool v1.1.3](#)
- ⇒ [Microsoft Lync Server 2010 Administration Guide & Windows PowerShell Supplement](#)
- ⇒ [Windows Phone 7 Guides for IT Professionals](#)
- ⇒ [Windows Phone 7.5 Guides for IT Professionals](#)
- ⇒ [Walkthroughs: Microsoft «Roslyn» CTP](#)
- ⇒ [Microsoft “Roslyn” CTP](#)
- ⇒ [Whitepaper: Roslyn Project Overview](#)
- ⇒ [Windows Server Operating System Management Pack for Operations Manager 2007](#)
- ⇒ [SQL Server Denali CTP3 Demo VHD](#)
- ⇒ [Microsoft® SQL Server® code name ‘Denali’, Community Technology Preview 3 \(CTP 3\) Product Guide](#)
- ⇒ [Microsoft Office Communications Server 2007 R2 Group Chat SDK](#)
- ⇒ [Microsoft iSCSI Software Target 3.3](#)
- ⇒ [Visual Studio 11 Developer Preview Training Kit](#)
- ⇒ [Microsoft Visio 2010 Templates for Weather Conditions](#)
- ⇒ [Microsoft Visio 2010 Template for Rack Server Virtualization](#)
- ⇒ [Microsoft adCenter Desktop \(Beta\)](#)
- ⇒ [DirectX 11 DirectCompute—A Teraflop for Everyone - GDC 2010](#)
- ⇒ [Unified Communications Phones and Peripherals Datasheets](#)
- ⇒ [Windows Embedded Compact 7 Evaluation Edition](#)
- ⇒ [Windows Embedded Compact 7 ATL Update for Visual Studio 2008 SP1](#)

- ⇒ [Windows Embedded Compact 7 Update 3](#)
- ⇒ [TPL Dataflow Community Technology Preview](#)
- ⇒ [Microsoft WCF Data Services October 2011 CTP](#)
- ⇒ [Microsoft Dynamics AX 2012 White Paper: Report Programming Model](#)
- ⇒ [Microsoft Dynamics AX 2012 White Papers: Code Upgrade](#)
- ⇒ [Microsoft Web Farm Framework Version 2.2 for IIS7 \(KB 2627276\)](#)
- ⇒ [Reactive Extensions \(Rx\) v1.1.11011 \(Experimental Release\)](#)
- ⇒ [Windows CE 5.0 Monthly Update September 2011](#)
- ⇒ [Security Update for Unified Access Gateway 2010 Service Pack 1 \(KB 2522485\)](#)
- ⇒ [SQL Server 2012 Developer Training Kit Web Installer Preview](#)
- ⇒ [Upgrading to Microsoft SharePoint Server 2010](#)
- ⇒ [Deployment guide for Microsoft SharePoint Server 2010](#)
- ⇒ [Profile synchronization guide for Microsoft SharePoint Server 2010](#)
- ⇒ [Diagnosing Application Compatibility Issues Affecting Windows Power Management](#)
- ⇒ [Security Update for Microsoft .NET Framework 1.1 Service Pack 1 on Windows XP, Windows Server 2003 \(64-bit\), Windows Vista, and Windows Server 2008](#)
- ⇒ [Microsoft JDBC Driver 4.0 for SQL Server CTP3](#)
- ⇒ [Update for Windows Mail Junk E-mail Filter \[October 2011\] \(KB905866\)](#)
- ⇒ [Update for Windows Mail Junk E-mail Filter for x64-based Systems \[October 2011\] \(KB905866\)](#)
- ⇒ [Microsoft® Windows® Malicious Software Removal Tool \(KB890830\)](#)
- ⇒ [Microsoft® Windows® Malicious Software Removal Tool \(KB890830\) x64](#)
- ⇒ [Security Update for Microsoft .NET Framework 1.0 Service Pack 3 on Windows XP Service Pack 3 Tablet PC Edition and Media Center Edition](#)
- ⇒ [Security Update for Microsoft .NET Framework 1.1 Service Pack 1 on Windows Server 2003 Service Pack 2 \(32-bit\)](#)
- ⇒ [Security Update for Microsoft .NET Framework 2.0 Service Pack 2 on Windows Vista Service Pack 2 and Windows Server 2008 Service Pack 2](#)
- ⇒ [Security Update for Microsoft .NET Framework 3.5.1 on Windows 7 Service Pack 1 and Windows Server 2008 R2 Service Pack 1](#)
- ⇒ [Security Update for Microsoft .NET Framework 2.0 Service Pack 2 on Windows Server 2003 and Windows XP](#)
- ⇒ [Security Update for Microsoft .NET Framework 4 ►](#)

**Примечание:** каждый понедельник в 10 часов утра мы публикуем на сайте журнала еженедельный бюллетень обновлений центра загрузок Microsoft.

- ⇒ [Security Update for Microsoft .NET Framework 3.5.1 on Windows 7 and Windows Server 2008 R2](#)
- ⇒ [Microsoft Dynamics AX 2009 White Paper: Customer Returns](#)
- ⇒ [Security Update for Microsoft Silverlight \(KB2617986\)](#)
- ⇒ [Security Update for Windows 7 for x64-based Systems \(KB2579686\)](#)
- ⇒ [Security Update for Windows Server 2003 \(KB2592799\)](#)
- ⇒ [Security Update for Windows 7 \(KB2579686\)](#)
- ⇒ [Security Update for Windows Embedded Standard 7 for x64-based Systems \(KB2579686\)](#)
- ⇒ [Update for Windows XP \(KB943232\)](#)
- ⇒ [Security Update for Windows XP x64 Edition \(KB2592799\)](#)
- ⇒ [Security Update for Unified Access Gateway 2010 UP2 \(KB2522484\)](#)
- ⇒ [Update for Windows 7 \(KB2604521\)](#)
- ⇒ [Security Update for Windows Vista \(KB2579686\)](#)
- ⇒ [Security Update for Windows Server 2003 for Itanium-based Systems \(KB2605295\)](#)
- ⇒ [Update for Windows Server 2008 for Itanium-based Systems \(KB2604521\)](#)
- ⇒ [Security Update for Unified Access Gateway 2010 UP1 \(KB2522483\)](#)
- ⇒ [Security Update for Windows Vista Media Center TVPack 2008 \(KB2579692\)](#)
- ⇒ [Security Update for Windows Server 2003 for Itanium-based Systems \(KB2592799\)](#)
- ⇒ [Security Update for Windows Embedded Standard 7 \(KB2579686\)](#)
- ⇒ [Windows Embedded Compact 7 Update 3](#)
- ⇒ [Tiled Resources for Xbox 360 and Direct3D 11](#)
- ⇒ [Modern Texture Content Pipelines](#)
- ⇒ [Microsoft® SQL Server® 2008 R2 Feature Pack](#)
- ⇒ [Update Rollup 6 for Exchange Server 2010 Service Pack 1 \(KB2608646\)](#)
- ⇒ [Microsoft Operations Framework \(MOF\) 4.0](#)
- ⇒ [System Center 2012 Orchestrator Integration Pack for VMware vSphere RC](#)
- ⇒ [System Center 2012 Orchestrator Integration Pack for IBM Tivoli Netcool/OMNIbus RC](#)
- ⇒ [Windows Azure Platform Training Kit - October Update](#)
- ⇒ [System Center 2012 Configuration Manager and System Center 2012 Endpoint Protection Release Candidates](#)
- ⇒ [System Center Service Manager 2012 Beta](#)
- ⇒ [System Center App Controller 2012 Beta](#)
- ⇒ [System Center Orchestrator 2012 Release Candidate](#)
- ⇒ [Microsoft Media Platform Content Manager](#)
- ⇒ [Microsoft Exchange Server MAPI Client and Collaboration Data Objects 1.2.1](#)
- ⇒ [OData Service for Team Foundation Server 2010 \(v1.0\)](#)
- ⇒ [System Center Monitoring Pack for Microsoft Data Warehouse Appliance](#)
- ⇒ [Microsoft® Visual Studio® 2010 and Team Foundation](#)

- [Server® 2010 RTM virtual machine for Windows Server 2008 Hyper-V](#)
- ⇒ [Microsoft Office 365 Service Descriptions and Service Level Agreements for Dedicated Subscription Plans](#)
- ⇒ [Microsoft Speech Platform - Runtime Languages \(Version 11\)](#)
- ⇒ [Microsoft Speech Platform - Software Development Kit \(SDK\) \(Version 11\)](#)
- ⇒ [The 2007 Microsoft Office Suite Service Pack 3 \(SP3\)](#)
- ⇒ [Service Pack 3 for Microsoft Office Proofing Tools 2007 \(KB2526293\)](#)
- ⇒ [Microsoft Dynamics CRM 2011 Software Development Kit \(SDK\)](#)
- ⇒ [Service Pack 3 for Visio Language Pack 2007 \(KB2526291\)](#)
- ⇒ [Microsoft Office Access Runtime and Data Connectivity 2007 Service Pack 3 \(SP3\)](#)
- ⇒ [Service Pack 3 for Microsoft Office Visio 2007 \(KB2526291\)](#)
- ⇒ [Excel Viewer 2007 Service Pack 3 \(SP3\)](#)
- ⇒ [Microsoft Office Compatibility Pack Service Pack 3 \(SP3\)](#)
- ⇒ [Service Pack 3 for Office Language Pack 2007 \(KB2526086\)](#)
- ⇒ [The 2007 Microsoft Office Servers Language Pack \(SP3\), 86-Bit Edition](#)
- ⇒ [The 2007 Microsoft Office Servers Language Pack \(SP3\), 64-Bit Edition](#)
- ⇒ [Service Pack 3 for Windows SharePoint Services 2007 \(KB2526305\), 64-Bit Edition](#)
- ⇒ [Microsoft® SQL Azure Labs Privacy Statement](#)
- ⇒ [Update for Windows Embedded POSReady 2009](#)
- ⇒ [Update for Windows Embedded POSReady 7](#)
- ⇒ [Governance guide for Microsoft SharePoint Server 2010](#)
- ⇒ [Microsoft SQL Server Connector for Apache Hadoop](#)
- ⇒ [Microsoft Exchange Server Protocol Documentation](#)
- ⇒ [Microsoft Exchange and Microsoft Outlook Standards Documentation](#)
- ⇒ [HPC Pack 2008 R2 SDK with Service Pack 2](#)
- ⇒ [Microsoft XNA Framework Redistributable 4.0 Refresh](#)
- ⇒ [BizTalk Server 2010 Documentation](#)
- ⇒ [Microsoft SQL Server 2008 Service Pack 3 Feature Pack](#)
- ⇒ [Microsoft SQL Server 2008 SP3 Reporting Services Report Builder](#)
- ⇒ [Microsoft SQL Server 2008 SP3 Reporting Services Add-in for Microsoft SharePoint Technologies](#)
- ⇒ [Microsoft XNA Game Studio 4.0 Refresh](#)
- ⇒ [SharePoint 2010 Podcasting Hands On Lab](#)
- ⇒ [System Center Monitoring Pack for Active Directory \(AD\)](#)
- ⇒ [SQL Server 2008 Service Pack 3](#)
- ⇒ [Microsoft SQL Server 2008 SP3 Express Edition](#)
- ⇒ [Microsoft Exchange Server 2010 SP2 Web Services SDK September 2011](#)
- ⇒ [Improving Microsoft Dynamics CRM Performance and Securing Data with Microsoft SQL Server 2008 R2](#)
- ⇒ [Microsoft BitLocker Administration and Monitoring](#)

# Отображение сложносоставных данных в WPF



## Автор

Борис Романов  
[boris.romanov@projectsprofiler.ru](mailto:boris.romanov@projectsprofiler.ru)  
 Компания: Projects Profiler LLC  
<http://projectsprofiler.ru>  
 г. Москва

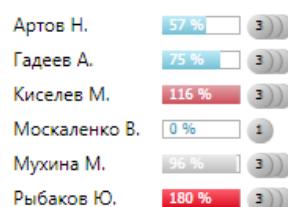
Технология WPF предлагает удивительно гибкие механизмы связывания данных, позволяющие отображать информацию в пользовательском интерфейсе максимально гибко, обновляя отображаемые данные в ответ на их изменения в программе. Однако в некоторых случаях изменения данных в программе могут происходить слишком часто, а для отображения информации может потребоваться выполнение дополнительных вычислений. В этом случае бездумное использование связывания данных может привести к сильным потерям производительности, интерфейс приложения будет «зависать» в ответ на действия пользователя.

## Пример задачи

Рассмотрим следующий пример. Пусть у нас есть сотрудники и проекты. Сотрудники заняты в проектах. Квант занятости сотрудника в проекте описывается объемом работ и непрерывным интервалом времени. Полная занятость сотрудника в проекте – это объединение всех квантов занятости этого сотрудника в конкретном проекте.

Теперь предположим, что нам нужно отображать пользователю статистическую информацию об участии каждого сотрудника в проектах: в каких проектах он участвует, какой объем работ выпол-

няет и т. п. Здесь все просто – нужно просто просуммировать соответствующие величины по набору занятостей и вывести результат в виде числа (или графического индикатора).



А теперь предположим, что в процессе выполнения пользовательских операций набор квантов занятостей каким-то образом изменяется – какие-то элементы добавляются, какие-то удаляются. Одна пользовательская операция может включать в себя множество элементарных операций по изменению исходного набора. В соответствии с шаблоном реализации привязки к данным, каждая элементарная операция должна приводить к генерации события изменения данных. Если элементы пользовательского интерфейса связаны с данными непосредственно и реагируют на события изменения данных, то обновления интерфейса будут происходить в ответ на каждое элементарное действие. Это приводит к необходимости выполнения большого количества вычислений:



$$K_{\text{выч}} = K_{\text{сот}} * K_{\text{знач}} * K_{\text{обн}}$$

где

$K_{\text{выч}}$  – общее количество вычислений,

$K_{\text{сот}}$  – количество отображаемых сотрудников,

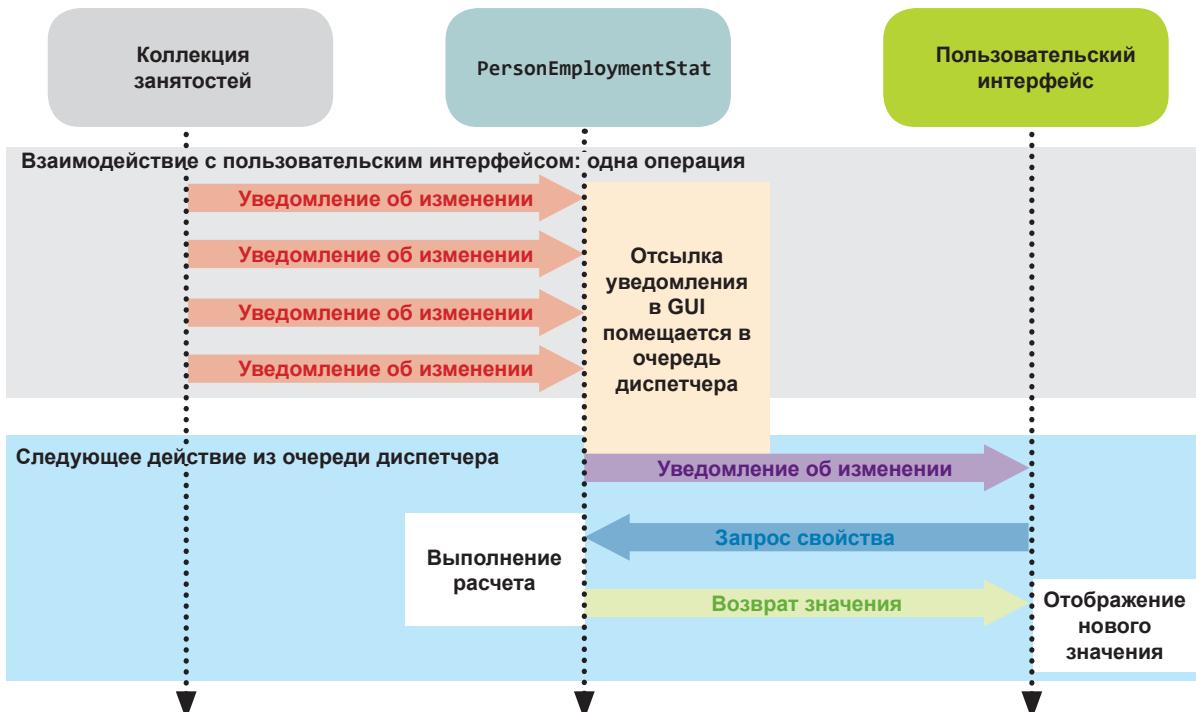
$K_{\text{знач}}$  – количество значений (отображаемых свойств) для каждого сотрудника,

$K_{\text{обн}}$  – количество обновлений интерфейса = количество элементарных операций.

Повлиять на количество сотрудников и отображаемых значений мы не можем, однако, в случае сложных операций, можно значительно уменьшить количество обновлений интерфейса – до одного на пользовательскую операцию. Тогда количество обновлений пользовательского интерфейса будет ограничено необходимым количеством.

## Решение задачи

Для уменьшения количества промежуточных вычислений при отображении статистических данных, создадим класс `PersonEmploymentStat`, который будет «накапливать» информацию о происходящих изменениях, передавая интерфейсу только необходимые уведомления. Для простоты у класса `PersonEmploymentStat` оставим только одно свойство – количество проектов, в которых участвует сотрудник. Для отложенного вызова уведомления используем `Dispatcher.BeginInvoke` с низким приоритетом. Асинхронный низкоприоритетный вызов начнет выполняться после того, как текущий вызов из диспетчера будет завершен. Т. е. гарантировано после завершения пользовательской операции, вызванной через кнопку, пункт меню или любой другой элемент интерфейса. Пользовательский интерфейс при получении ►



уведомления об изменении данных обратится к экземпляру класса PersonEmploymentStat за новыми данными. И только в этот момент выполнится фактическое вычисление.

Для отслеживания факта изменения исходных данных, класс PersonEmploymentStat подписывается на событие их изменения. Заметим, что обычные события C# здесь не подходят, так как они будут приводить к утечкам памяти, поскольку класс PersonEmploymentStat конструируется неявно через конвертер и не может отследить факт завершения своего использования. Поэтому в нашем случае уместно применить реализацию событий с помощью слабых ссылок (см. статью о реализации паттерна «наблюдатель» с использованием слабых ссылок в предыдущем номере журнала (1)).

```

public class PersonEmploymentStat : INotifyPropertyChanged
{
    /// <summary>
    /// Вычисляемое (и запомненное) количество проектов
    /// </summary>
    int? _projCount;

    /// <summary>
    /// Для какого сотрудника выполняется вычисление
    /// </summary>
    Person _who;

    /// <summary>
    /// Перечисление всех квантов занятости (для
    /// отбора среди них нужных)
    /// </summary>
    IEnumerable<EmploymentAtom> _allEmployments;

    /// <summary>
    /// Сильная ссылка на обработчик слабого события
    /// изменения коллекции квантов занятости
    /// </summary>
    EventReceiver _updater;

    /// <summary>
    /// Признак того, что операция расчета уже
    /// помещена в очередь
    /// </summary>
    private bool _RecalcInQueue;
  
```

```

    /// Квантов занятостей</param>
    /// <param name=>changeEvent</param>Интерфейс
    /// <param name=>слабоссыльного</param> события изменения
    /// коллекции</param>
    public PersonEmploymentStat(Person who, IEnumerable<EmploymentAtom> allEmployments,
        IWeakEvent changeEvent)
    {
        if (who == null)
            throw new ArgumentNullException("who");
        if (allEmployments == null)
            throw new ArgumentNullException("all-Employments");

        //запоминаем в классе сотрудника и коллекцию
        _who = who;
        _allEmployments = allEmployments;
        //подписываемся на уведомление об изменении коллекции
        changeEvent.AddReceiver(_updater = new Event-
        Receiver(new Action(OnSourceChanged)));

        _projCount = null;
    }

    /// <summary>
    /// Помещение операции перерасчета в очередь диспетчера
    /// </summary>
    private void OnSourceChanged()
    {
        _projCount = null;

        if (!_RecalcInQueue)
        {
            //помещаем операцию отсылки уведомления
            //в очередь диспетчера
            //только если ее там еще нет
            _RecalcInQueue = true;
            var oper = Dispatcher.CurrentDispatcher
                .BeginInvoke(NotifyCountChanged,
                DispatcherPriority.Background);
            //при завершении операции снимаем
            //отметку о ее наличии в очереди
            oper.Completed += new EventHandler((x,
                y) => { _RecalcInQueue = false; });
        }
    }
  
```

```

/// <summary>
/// Уведомление об изменении свойства
/// </summary>
private void NotifyCountChanged()
{
    if (PropertyChanged != null)
        PropertyChanged(this, new PropertyChangedEventArgs("ProjectCount"));
}

/// <summary>
/// Количество проектов (с выполнением расчета
/// при необходимости)
/// </summary>
public int ProjectCount
{
    get
    {
        if (_projCount == null)
            _projCount = _allEmployments.Count(x => _who.Equals(x.Person));

        return _projCount.Value;
    }
}

#region INotifyPropertyChanged Members

public event PropertyChangedEventHandler PropertyChanged;

#endregion
}

```

Данные будем отображать с помощью ContentControl, передавая ему экземпляр класса в качестве содержимого. Причем экземпляр класса PersonEmploymentStat будем конструировать с помощью конвертера. Это даст возможность использовать этот класс непосредственно в XAML, а не в коде. Для управления отображением данных используем соответствующий ему шаблон.

```

<!--Список сотрудников-->
<ListBox x:Name="Persons" ItemsSource="{Binding AllPersons}" />
<!--Элемент для отображения составных данных-->
<ContentControl x:Name="UserStatistics">
    <ContentControl.Resources>
        <!--Конвертер для конструирования экземпляра
        PersonEmploymentStat-->
        <local:StatisticsBuilder x:Key="StatConverterBuilder" />
        <!--Шаблон для показа данных-->
        <DataTemplate DataType="{x:Type local:PersonEmploymentStat}">
            <StackPanel>
                <TextBlock Text="Количество проектов:"/>
                <TextBlock Text="{Binding ProjectCount}"/>
            </StackPanel>
        </DataTemplate>
    </ContentControl.Resources>
    <ContentControl.Content>
        <!--Отображаемое содержимое формируется с
        помощью конвертера-->
        <MultiBinding Converter="{StaticResource StatisticsBuilder}">
            <!--В конвертер передается выбранный в
            списке сотрудник-->
            <Binding Path="SelectedObject" ElementName="Persons"/>
            <!--и коллекция квантов занятостей-->
            <Binding Path="AllEmploymentsAccessors" />
        </MultiBinding>
    </ContentControl.Content>
</ContentControl>

```

Для упрощения создания конвертера используется способ, описанный в (см. ссылку 2):

```

/// <summary>
/// интерфейс доступа к коллекции
/// со слабоссыльным событием
/// изменения состава коллекции
/// </summary>
public interface ICollectionAccessor<T>
{
    IEnumerable<T> Collection {get;}
    IWeakEvent collectionChangedEvent {get;}
}

/// <summary>
/// конвертер для конструирования экземпляра
PersonEmploymentStat
/// </summary>
public class StatisticsBuilder : UniConverter,
IMultiValueConverter
{
    /// <summary>
    /// метод конвертера для генерации объектов
    /// статистики
    /// </summary>
    public PersonEmploymentStat PersonEmployment-
    StatFactory(Person pers,
    ICollectionAccessor<EmploymentAtom>
    employments )
    {
        return new PersonEmploymentStat(pers, emp-
        lishments.Collection,
        employments.collectionChangedEvent);
    }
}

```

## Заключение

Данный пример сильно упрощен, по сравнению с практическим применением, но полностью показывает суть метода. При необходимости отображения в пользовательском интерфейсе большого количества значений, получение которых требует выполнения вычислений, этот подход позволяет существенно сократить время обновления отображаемой информации в ответ на действия пользователя. Это достигается за счет того, что вычисления начинают выполняться только после завершения операции. К недостаткам метода можно отнести тот факт, что значения могут появляться на экране с небольшой задержкой.

Описанный метод применим, когда незначительные задержки в появлении данных являются некритичными, а вычисление каждого отдельно взятого значения не занимает много времени. В этом случае вычисление каждого из значений помещается в очередь диспетчера независимо, и отложенные вычисления не блокируют поток пользовательского интерфейса.

Безусловно, если это допускается исходными данными, можно перенести вычисление свойств в отдельные потоки, но это потребует дополнительного внимания к целостности данных во время выполнения пользовательских операций при наличии незавершенных отложенных вычислений. Использование очереди диспетчера гарантирует сохранение последовательности выполнения операций и значительно упрощает код. ■

## Список литературы

1. Б. Л. Романов. «Реализация паттерна Наблюдатель с использованием слабых ссылок. Журнал MSDeveloper.RU». Октябрь 2011 г.
2. Б. Л. Романов. «Упрощаем множественные привязки к данным (MultiBinding)». Создание программного продукта - проблемы и решения. [В Интернете] Сентябрь 2011 г.

# Постоянныe подписчики



## Автор

Андрей Гордиенко  
my@violet-tape.net  
<http://softblog.violet-tape.ru/>  
г. Москва



При разработке сервисов для небольшой программы, скорее всего, не думаете о многопоточности, о том, что какой-либо метод сервиса будут запускать сразу несколько экземпляров класса и вообще много еще о чем обычно не думаете. Так же как и я. В небольших программах, которые наверно будут исполняться в один поток, не надо развитой системы событий. Можно просто передать делегат, который будет вызван после окончания работы сервиса.

Может быть обстоятельства, а может быть что-то уложилось в голове, но описанный выше подход больше не работает в программах, которые я пишу. Слишком много действий одновременно могут делать пользователь и начинка программы, чтобы оставаться в рамках модели «один сервис – один потребитель». Однако с новым подходом меняется сложность и ответственность кода, и на этом пути поджидают многие опасности.

Новый подход, как вы наверно догадались, будет состоять в построении системы на событиях (event). При первом взгляде не видно никаких проблем при использовании событий, но это только на первый взгляд. Не уверен, что можноправильно проранжировать опасности, но, пожалуй, перечислю их в той последовательности, в которой с ними сталкиваешься.

Самые частые ошибки, которые вводят людей в ступор, это когда они обнаруживают:

- ▀ кратные прибавки в вычислениях,
- ▀ двойные записи в базе данных,
- ▀ прочие казусы вычислений и отображений данных.

Неявной проблемой, с которой сталкиваются постепенно (при отсутствии проблем с вычислениями), является скорость работы приложения. При двойной, тройной и так далее подписках, программа начинает делать одни и те же действия без всякой необходимости, что может серьезно снизить производительность.

Бывает так, что повторной подписки не происходит, но и отписаться от события забыли. В этом случае могут возникнуть утечки памяти, так как сервис держит класс и не дает сборщику мусора пометить объект на удаление.

К слову, не всегда бывает удобно отписываться и хочется, чтобы подписка была «слабой», как WeakReference. Тогда если на объект нет внешних ссылок, кроме слабых подписок, то сборщик мусора пометит его на удаление, подписка освободится, память освободится и будет небольшое программистское счастье. Собственно, об этом и пойдет речь, как реализовать и использовать в своем приложении слабую подписку на события. Попутно я планирую раскрыть еще несколько рабочих моментов при использовании событий и сервисов.

## Эволюция применения

Прежде чем перейти к сути, я бы хотел немного остановиться на том, как может идти развитие проекта и какие проблемы могут вставать на пути. Начнем с самого простого. Допустим, есть какой-то сервис, который делает сложный расчет в течение нескольких секунд. Для того чтобы не блокировать выполнение других задач, например, обновление UI, будем запускать расчет в отдельном потоке, передавая параметры для расчета и метод для обратного вызова.

Самым простым примером может послужить сервис следующего вида:

```
public class SomeStraightCalcService {
    public void LongRunningMethodAsync(string val,
        Action callback) {
        ThreadPool.QueueUserWorkItem(LongRunning-
            Method, new {val, callback});
    }

    private void LongRunningMethod(dynamic obj) {
        Thread.Sleep(2000);
        obj.callback();
    }
}
```

Создаем новый поток с помощью пула потоков ThreadPool, вызвав метод QueueUserWorkItem. Это наверно самый простой способ для создания нового потока, при этом забота о создании, утилизации и работе потока перекладывается на систему. С помощью этого метода мы сводим практически к нулю вероятность натворить безобразий с потоками. Далее идет эмуляция долгих расчетов и вызов обратной функции, которая подскажет вызывающему классу, что работа готова.

Использование сервиса для демонстрационных целей сделаем таким:

```
internal class NonBlockingResource {
    public Stopwatch Watch1 = new Stopwatch();
    public Stopwatch Watch2 = new Stopwatch();

    public NonBlockingResource() {
        var service = new SomeStraightCalcService();
        Watch1.Start();
        service.LongRunningMethodAsync("“, Ends1);

        Watch2.Start();
        service.LongRunningMethodAsync("“, Ends2);
    }

    private void Ends1() {
        Watch1.Stop();
    }
} ▶
```

## Разработка

```
private void Ends2() {
    Watch2.Stop();
}
}
```

Создаем таймеры, запускаем долгоиграющие методы, затем смотрим результат и сравниваем. Здесь вообще ничего сложного нет. Средой для проверки будут тестовые методы

```
[TestMethod]
public void SimpleLongRunning() {
    var someTestClass = new NonBlockingResource();

    Thread.Sleep(5000);
    var elapsedMs1 = someTestClass.Watch1.Elapsed-
        Milliseconds;
    var elapsedMs2 = someTestClass.Watch2.Elapsed-
        Milliseconds;
    Assert.Fail("1st call = {0}ms, 2nd call = {1}-
        ms", elapsedMs1, elapsedMs2);
}
```

Для вывода информации запорем тест, и в сообщении выведем нужную нам информацию. У меня получилось примерно так: *1st call = 2005ms, 2nd call = 2004ms.*



Далее обычно идет осознание того факта, что вычисления портятся, либо данный подход на-тыкается на использование ресурса, к которому желательно обращаться монопольно.

Теперь вспоминаются начальные знания по многопоточности и на сцену выходит оператор lock.

Пусть монопольным ресурсом будет такой класс:

```
public class SingleAccess {
    private readonly object locker = new object();

    public void Enter() {
        lock (locker) {
            Thread.Sleep(2000);
        }
    }
}
```

Тогда сервис можно переписать таким образом:

```
public class SomeStraightCalcServiceWithBlockingRe-
source {
    private readonly SingleAccess singleAccess = new
        SingleAccess();

    public void LongRunningMethodAsync(string val,
        Action callback) {
        ThreadPool.QueueUserWorkItem(LongRunning-
            Method, new {val, callback});
    }

    private void LongRunningMethod(dynamic obj) {
        singleAccess.Enter();
        obj.callback();
    }
}
```

По сути, не сильно отличается от первоначального примера. Класс, использующий сервис, по своей структуре не поменяется вообще, только будет создавать другой экземпляр сервиса. Тестовый метод:

```
[TestMethod]
public void LongRunningWithBlockingResource() {
    var someTestClass = new WithBlockingResource();

    Thread.Sleep(5000);
    var elapsedMs1 = someTestClass.Watch1.Elapsed-
        Milliseconds;
    var elapsedMs2 = someTestClass.Watch2.Elapsed-
```

```
Milliseconds;
    Assert.Fail("1st call = {0}ms, 2nd call = {1}-
        ms", elapsedMs1, elapsedMs2);
}
```

И результаты вполне ожидаемые: *1st call = 4006ms, 2nd call = 2022ms.*



Дальнейшее развитие ситуации может происходить таким образом, что запрос к блокирующему ресурсу можно выполнить один раз и оповестить об окончании вычислений всех интересующихся.

Т. е. идет первый вызов к сервису, начинает выполняться долгоиграющий код. Затем, пока первый запрос все еще выполняется, какая-то часть программы тоже захочет получить эти же данные и делает второй запрос. В системе, построенной на обратных вызовах, придется в общем случае делать еще один такой же долгий запрос, как только первый пройдет. Либо делать хитрое кеширование и еще что-нибудь.

Более логично будет сказать системе, что как только ты завершил обрабатывать запрос, выдай результаты обоим потребителям, а не делай все то же самое еще раз. Т. е. подписаться на событие. В самом простом варианте это выглядит так (в рабочем коде не забываем про double check lock):

```
public class CalcServiceWithBlockingResource {
    private readonly SingleAccess singleAccess = new
        SingleAccess();
    private bool inAction;

    public event Action Ends;

    public void LongRunningMethodAsync(string val) {
        if(inAction) return;

        inAction = true;
        ThreadPool.QueueUserWorkItem(LongRunning-
            Method, val);
    }

    private void LongRunningMethod(object obj) {
        singleAccess.Enter();
        if(Ends != null)
            Ends();
        inAction = false;
    }
}
```

В качестве простой проверки используется булева переменная *inAction* для того, чтобы не запускать два раза расчет, если он уже идет.

Использование данного сервиса будет немного отличаться от предыдущих примеров, но не кардинальным образом, на мой взгляд.

```
internal class EventWithBlockingResource {
    public Stopwatch Watch1 = new Stopwatch();
    public Stopwatch Watch2 = new Stopwatch();
    private readonly CalcServiceWithBlockingRe-
        source service;

    public EventWithBlockingResource() {
        service = new CalcServiceWithBlocking-
            Resource();
        service.Ends += Ends1;
        service.Ends += Ends2;

        Watch1.Start();
        service.LongRunningMethodAsync("");
    }

    public void Ends1() {
        Watch2.Start();
        service.LongRunningMethodAsync("");
    }
}
```

```

private void Ends1() {
    Watch1.Stop();
    service.Ends -= Ends1;
}

private void Ends2() {
    Watch2.Stop();
    service.Ends -= Ends2;
}
}

```

Вместо передачи метода для обратного вызова, подписываемся на событие об окончании работы метода.

```

[TestMethod]
public void EventBasedWithSameBlockingResource() {
    var someTestClass = new EventWithBlockingRe-
    source();

    Thread.Sleep(5000);
    var elapsedMs1 = someTestClass.Watch1.Elapsed-
    Milliseconds;
    var elapsedMs2 = someTestClass.Watch2.Elapsed-
    Milliseconds;
    Assert.Fail("1st call = {0}ms, 2nd call = {1}-
    ms", elapsedMs1, elapsedMs2);
}

```

Результат: 1st call = 2000ms, 2nd call = 2000ms.

## «Слабые» делегаты

На этом вводную часть можно закончить и перейти к «слабым» делегатам, и показу их работы на примере приложения с несколькими экранами, которые участвуют в подписке на события. В данном примере будет наглядно показано, как утекает память, если не отписываться от событий.



Приложение будет насколько это возможно простым для демонстрации утечек в реальной жизни.

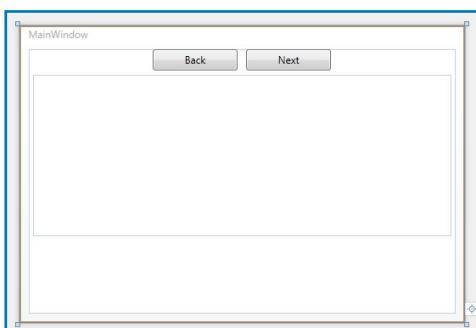
Допустим, у нас есть модели представлений, которые подписываются на некий постоянный сервис, в роли которого выступит основная форма окна. При переходе между UserControls будем замерять занятую память, предварительно запустив сборщик мусора для прохождения по всем поколениям объектов.

```

private void CheckMemory() {
    GC.Collect();
    Message.Content = GC.GetTotalMemory(true);
}

```

Основное окно будет состоять из кнопок для создания UserControls, надписи для вывода объема занятой памяти и панели для показа компонентов.



Вот такое простое будет у нас окно. Начинка у него состоит из:

- ▼ события,

- ▼ вызова события,
- ▼ обработчика нажатия кнопки один,
- ▼ обработчика нажатия кнопки два.

Все просто!

Следуя классике жанра, событие и его вызов будут объявлены так:

```

public event EventHandler LocalEvent;

protected virtual void OnClickEventArgs e) {
    var event = LocalEvent;
    if (event != null)
        event (this, e);
}

```

Обработчик кнопки будет не намного сложнее:

```

private void PrevScreenClick(object sender, RoutedEventArgs e) {
    var controlOne = new ControlOne();
    var oneModel = ((ModelOne) controlOne.DataContext);
    LocalEvent += oneModel.LinkingMethod;
    SetContent(e, controlOne);
}

```

Создаем пользовательский компонент. Получаем его контекст, подписываемся на событие из вышемодели ModelOne. В методе SetContent будет отображение компонента, вызов обработчика события и проверка занятой памяти:

```

private void SetContent(RoutedEventArgs e, UserControl control) {
    content.Children.Clear();
    content.Children.Add(control);

    OnClick(e);

    CheckMemory();
}

```

В свою очередь модели представлений так же очень просты и не содержат никакой логики, так как это не требуется для работы демонстрационного примера.

```

public class ModelOne {
    public void LinkingMethod(object sender, EventArgs eventArgs) {}
}

public class ModelTwo {
    public void LinkingMethod(object sender, EventArgs eventArgs) {}
}

```

Создание моделей и назначение контексту пользовательского компонента будет происходить при создании самого пользовательского компонента:

```

public partial class ControlOne : UserControl {
    public ControlOne() {
        InitializeComponent();
        DataContext = new ModelOne();
    }
}

```

Думаю, что дальнейших разъяснений не требуется и можно запускать приложение и смотреть, что же получается в результате работы. При запуске показывается, что занято 634420 байт. При нажатии на любую из кнопок создается userControl и происходит подпись на событие, после чего размер занимаемой памяти вырастает до 764684 байт. При повторных нажатиях начинается непрерывный рост:

768204

768256

768420

777648 ►

Таким образом, я добился того, что стало показывать 790234 байт. Можете быстро щелкать по кнопкам или ждать в надежде, что подписка отсохнет магическим образом или как-то еще освободиться память, но этого не будет. Память, в конечном счете, будет только потребляться.



*Все потому, что мы забыли отписаться от событий, созданные компоненты привязаны к основной форме через события, и сборщик мусора думает, что они требуются программе, не помечая их на удаление.*

От возникшей проблемы можно избавиться, если реализовать делегаты на «слабых» ссылках. WeakReference «невидимы» для сборщика мусора и позволяют ему помечать объекты на удаление, если есть только такие «слабые» ссылки на объект.

Дабы не томить, сразу приведу полный текст для «слабого» делегата.

```
internal class WeakDelegate<TDelegate> where
    TDelegate : class {
        private readonly List<WeakReference> targets =
            new List<WeakReference>();

        public WeakDelegate() {
            if (!typeof(TDelegate).IsSubclassOf(typeof(Delegate)))
                throw new InvalidOperationException
                    ("TDelegate должен быть настоящим
                     делегатом");
        }

        public void Combine(TDelegate target) {
            if (target == null) return;
            foreach (var d in (target as Delegate).Get-
                InvocationList())
                targets.Add(new WeakReference(d));
        }

        public void Remove(TDelegate target) {
            if (target == null) return;
            foreach (var d in (target as Delegate).Get-
                InvocationList()) {
                var weak = targets.Find(w => d.Equals
                    (w.Target));
                if (weak != null) targets.Remove(weak);
            }
        }

        public TDelegate Target {
            get {
                var deadRefs = new List<WeakReference>();
                Delegate combinedTarget = null;
                foreach (var weak in targets) {
                    var target = (Delegate) weak.Target;
                    if (target != null)
                        combinedTarget = Delegate.Combi-
                            ne(combinedTarget, target);
                    else
                        deadRefs.Add(weak);
                }
                foreach (var weak in deadRefs) // удаление
                    targets.Remove(weak);
                return combinedTarget as TDelegate;
            }
            set {
                targets.Clear();
                Combine(value);
            }
        }
    }
}
```

Теперь можно подробнее разобрать этот код. В конструкто-

ре проверяется, чтобы переданный параметр действительно был делегатом, так как в C# нет ограничителя на шаблонный параметр вида delegate. Нельзя написать

```
internal class WeakDelegate<TDelegate> where TDel-
egate : delegate { ... }
```

Переменной класса является список слабых ссылок, для того чтобы можно было осуществлять множественную подписку.

Методы Combine и Remove будут отвечать соответственно за добавление подписчиков и удаление их. Это потребуется при создании события в итоговом классе. В этих методах у каждого делегата берется список подписчиков, так как они могут быть так же с множественной подпиской. При добавлении, каждый полученный из списка подписчик оборачивается в «слабую» ссылку. При удалении, предварительно проверяем, существует ли до сих пор подписчик.

При получении самого списка подписок, мы отфильтровываем мертвые подписки и возвращаем подписки только для существующих подписчиков.

В целом все не так сложно, если внимательно проговорить про себя код и может даже нарисовать примерный ход выполнения на бумаге. После этого можно использовать «слабые» события в нашем, кхе-кхе, приложении и посмотреть, как будет меняться динамика потребления памяти. Для этого нам потребуется кое-что заменить.

Удаляем старое событие и заменяем его на ручную реализацию:

```
private readonly WeakDelegate<EventHandler> click =
    new WeakDelegate<EventHandler>();

public event EventHandler LocalEvent {
    add { click.Combine(value); }
    remove { click.Remove(value); }
}
```

Метод OnClick также подвергнется небольшим изменениям:

```
protected virtual void OnClick(EventArgs e) {
    var target = click.Target;
    if (target != null) target(this, e);
}
```

Не могу назвать их кардинальными, логика осталась та же самая. Только поправка на структуру делегата.

Собственно все, можно запускать приложение, нажимать на кнопки и смотреть, как меняется потребляемая память. При запуске показывается, что занято 634416 байт. После первого нажатия 764808, потом:

```
777560
777744
```

И далее будет все крутиться около 778000 байт с небольшими отклонениями.

Постоянного роста памяти не будет происходить, чего мы и добивались. Надеюсь, данная техника позволит вам сохранить нервные клетки, память и голосовые связки. При этом все равно не стоит забывать отписываться от событий, даже если они слабо связаны.

\*\*\*

В реальных проектах вместо ThreadPool лучше использовать Task, так как они дают больший контроль над процессом.

Системы на событиях гораздо сложнее, чем можно себе представить, руководствуясь только тем, что было освещено в данной статье.

Исходный код можно взять [здесь](#). Статья написана по мотивам книги J.Albahari «C# in the Nutshell 4.0» ■



# MFCMAPI: траблшутим почтовые ящики MS Exchange Server



Автор

Александр Рагель

Enterprise Messaging Administrator 2007, 2010 (Charter Member), MCSA, Microsoft Operations Framework

Руководитель по развитию услуг для корпоративных клиентов компании Infobox.ru  
[www.infobox.ru](http://www.infobox.ru)

г. Санкт-Петербург

**Б**ольшинство людей, так или иначе занятых администрированием MS Exchange Server (включая не только server-side проблемы, но и вопросы клиентского доступа и связанные с ними) сталкивались и использовали эту утилиту. MFCMAPI является своего рода наследником распространенной в прошлом утилиты MDBVU, которая стала достаточно популярной в конце 1990-х – начале 2000 годов.

Автор MFCMAPI – Stephen Griffin, сейчас уже работает в Microsoft. Раньше были простые «упражнения» с MAPI вызовами, сейчас это общепризнанно лучший инструмент для трафаретинга проблем на уровне почтового ящика. MFCMAPI – это швейцарский нож в руках умелого специалиста и позволяет творить чудеса. Ниже я рассмотрю некоторые из возможных сценариев использования MFCMAPI для решения разнообразных задач, которые могут встать перед Exchange Server администратором.

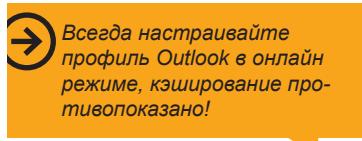
 Не забываем проверить новую версию тут

Универсальность MFCMAPI великолепно показывает себя при устранении проблем с неправильно функционирующим junk email правилом, правке скрытых папок, удалению «застрявших» делегатов из почтового ящика пользователя, устранению проблем при корпоративной активации BlackBerry(!), исправлению неработающего Out-of-Office, нахождении причины перемещения сообщения в папку Junk и многих-многих других. ►

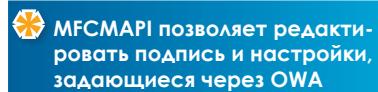
Рис. 1

Рис. 2

# Microsoft Exchange Server



Рассмотрим некоторые интересные из возможностей применения этого инструмента.



Для чего это может быть полезно? Например, когда подпись создается с использованием массы тегов и необычного форматирования, это может вызвать то, что Outlook будет аварийно закрываться при каждой попытке его открыть, а возможности отредактировать эту настройку через OWA не будет. Что делать? Некоторые горячие головы из числа админов предлагают просто пересоздать ящик. Радикально и технически не вполне верно, ибо у нас есть MFCMAPI.

Запускаем, логинимся в нужный профиль: **Logon and display store table** → **Root Container** → **Open Associated Content Table** и в колонке **Message Class** находим требуемое значение **IPM.Configuration.OWA.UserOptions** (рис. 1), в котором смотрим на свойство **PR\_ROAMING\_DICTIONARY**.

Выбираем из контекстного меню **Edit Property** и видим все (рис. 2).



На смену nk2-файлам пришли значения, сохраняющиеся в свойствах почтового ящика. Мы можем получить доступ к этим значениям и отредактировать их в случае необходимости.

Местонахождение значения похоже на предыдущее: **Logon and display store**

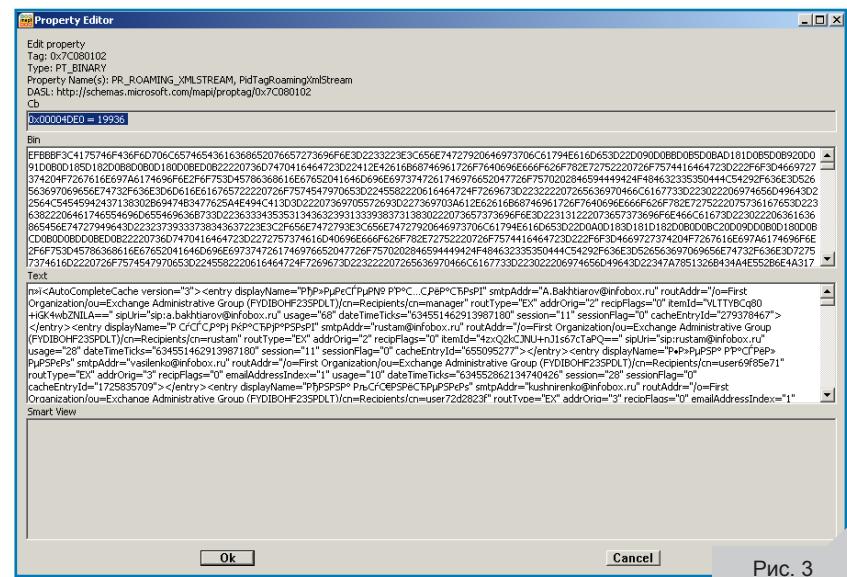


Рис. 3

Property Name(s)	Other Names	Tag	Type	Value
PR_FOO0040		0:3F600040	PT_SYSTIME	01:22:46 24.01.
PR_IC5_CHANGE_KEY		0:66550004	PT_ERROR	Err:0x000100F...
PR_IC5_CHANGE_KEY		0:66550004	PT_STRINGB	HUB2010-01.ex...
PR_ACCESS_PidTagAccess...		0:0PF40003	PT_LONG	63
PR_ACCESS_LEVEL_PidTagAccess...		0:0PF70003	PT_LONG	1
PR_ASSOC_CONTENT_COUNT_Pl...		0:36170003	PT_LONG	1
PR_ATTR_HIDDEN_PidTagAttrib...		0:10F40008	PT_BOOLEAN	False
PR_ATTR_READONLY_PidTagAttrib...		0:10F60008	PT_BOOLEAN	False
PR_ATTR_SYSTEM_PidTagAttrib...		0:10F50008	PT_BOOLEAN	False
PR_CHANGE_KEY_PidTagChange...		0:65E20102	PT_BINARY	cb: 20 lpb: 8D18...
PR_COMMENT_PR_COMMENT_A...	PR_COMMENT_W, PidTagComment	0:3004001E	PT_STRINGB	01:22:46 24.01...
PR_CONTAINER_CLASS_PR_CON...	PR_CONTAINER_CLASS_W, PidTag...	0:3613001E	PT_STRINGB	IPF.Contact
PR_DELETED_ASSOC_MESSAGE_...		0:66900014	PT_I8	cb: 00000000:0:0
PR_DELETED_COUNT_PR_MSG_COUN...		0:66940003	PT_LONG	0
PR_DELETED_COUNT_TOTAL		0:67080003	PT_LONG	20
PR_DELETED_SUB_FOLDER_COUN...		0:66410003	PT_LONG	0
PR_DELETED_MESSAGE_SIZE_EX...		0:66980014	PT_I8	0x00000000:0:0
PR_DELETED_MSG_COUNT		0:66400003	PT_LONG	0
PR_DELETED_NORMAL_MESSAGE...		0:669C0014	PT_I8	0x00000000:0:0
PR_DELETED_NORMAL_MESSAGE...		0:3FEE40008	PT_BOOLEAN	False
PR_DISPLAY_NAME_PR_DISPLAY...	PR_DISPLAY_NAME_W, PidTagDisp...	0:3001001E	PT_STRINGB	LinkedIn
PR_ENTRYID_PR_MEMBER_ENTRYR...		0:39000003	PT_LONG	1677216
PR_EXTENDED_FOLDER_FLAGS...		0:0FF0102	PT_BINARY	cb: 46 lpb: 0000...
PR_FOLDER_CHILD_COUNT		0:66380003	PT_LONG	0
PR_FOLDER_PATHNAME		0:6685001E	PT_STRINGB	?Linkedin
PR_FOLDER_TYPE_PidTagFolderT...		0:36010003	PT_LONG	1
PR_HAS_RULES_PidTagHasRules...	PR_PROFILE_AB_PTHREAD_TIME...	0:66340008	PT_BOOLEAN	False
PR_INTERNET_ARTICLE_NUMBER...		0:0E230003	PT_I8	0x00000000:0:0
PR_LAST_MODIFICATION_TIME...		0:30080040	PT...	

Рис. 4

table → Root Container → Open Associated Content Table и в колонке Message Class смотрим на IPM.Configuration.OWA.AutoCompleteCache.

В нижнем окне находим **PR\_ROAMING\_XMLSTREAM**, выбираем **Edit Property** и все перед глазами (рис. 3). Манипулировать записями в кэше автозаполнения отсюда совершенно легко.

В некоторых особо тяжелых случаях повреждения почтового ящика, свойство **PR\_ROAMING\_XMLSTREAM** отсутствует, и кэширование в таком случае работает совершенно непредсказуемо. Что делать и как быть?

Легко! Выбираем IPM.Configuration.OWA.AutoCompleteCache и применя-

Property Name(s)	Other Names	Tag	Type	Value
PR_ACCESS_LEVEL_PidTagAccess...		0:0FF70003	PT_LONG	1
PR_ADDITIONAL_REN_ENTRYIDS...		0:36D81102	PT_MV_BINARY	6: cb: 46 lpb: 00000000:02D261B7F41...
PR_ADDITIONAL_REN_ENTRYIDS...		0:36D90102	PT_BINARY	cb: 328 lpb: 04B0320001002E00000...
PR_ASSOC_CONTENT_COUNT_Pl...		0:36170003	PT_LONG	35
PR_ATTR_HIDDEN_PidTagAttrib...		0:10F40008	PT_BOOLEAN	False
PR_ATTR_READONLY_PidTagAttrib...		0:10F60008	PT_BOOLEAN	False
PR_ATTR_SYSTEM_PidTagAttrib...		0:10F50008	PT_BOOLEAN	False
PR_CHANGE_KEY_PidTagChange...		0:65E20102	PT_BINARY	cb: 22 lpb: E352E08F2CA22D4EA91...
PR_COMMENT_PR_COMMENT_A...	PR_COMMENT_W, PidTagComment	0:3004001E	PT_STRINGB	01:22:46 24.01...
PR_CONTAINER_CLASS_PR_CON...	PR_CONTAINER_CLASS_W, PidTag...	0:3613001E	PT_STRINGB	IPF.Note
PR_CONTENT_COUNT_PidTagCo...		0:36030003	PT_LONG	7591
PR_CONTENT_UNREAD_PidTagC...		0:30070040	PT_SYSTIME	08:30:09 13.12.2010
PR_CREATION_TIME_PidTagCrea...		0:0E580102	PT_BINARY	cb: 28 lpb: 01500000000000051500...
PR_DELETED_ASSOC_MESSAGE_...		0:669D0014	PT_I8	0x00000000:0:00000000
PR_DELETED_COUNT_TOTAL		0:67080003	PT_LONG	3441
PR_DELETED_FOLDER_COUNT		0:66410003	PT_LONG	0
PR_DELETED_MESSAGE_SIZE_EX...		0:66980014	PT_I8	0x00000000:0:00000000
PR_DELETED_MSG_COUNT		0:66400003	PT_LONG	0
PR_DELETED_NORMAL_MESSAGE...		0:669C0014	PT_I8	0x00000000:0:00000000

Рис. 5

# Microsoft Exchange Server

LinkedIn: IPM.Microsoft.OSC.ContactSync								
File	Actions	Folder	New	Property Pane	Other	Table		
Instance Key	Att?	From	To	Subject	Conversation ID	Row Type	Content Count	Content Unread
db: 20 lp: ... False		Err:0x8004010...		IPM.Microsoft....	Err:0x8004010...	1 (TBL_LEAVE_R...	Err:0x8004010...	Err:0x8004010...
<hr/>								
Property Name(s)	Other Names	Tag	Type	Value	Value - Alternate view	Smart View		
PR_ICCS_CHANGE_KEY		0x6550102	PT_BINARY	cb: 44 lp: 290CF29E54C0444CB5C...	,rTADLpKo3R,б,4=FrRaU,ÿN...			
PR_IMPORTANCE, PidTagImporta...		0x00170003	PT_LONG	1	0x1	Flags: IMPORTANC...		
PR_INTERNET_CID, PidTagInter...		0x02E30003	PT_LONG	105	0x69			
PR_INTERNET_MESSAGE_ID, PR_...	PR_INTERNET_MESSAGE_ID_W, Pid...	0x03FDE0003	PT_LONG	20127	0x4EF9			
PR_LAST_MODIFICATION_TIME, ...		0x1035001E	PT_STRING8	<E352E08F2CA2D4EA9172411695...	cb: 69 lp: 3C45335324530384632...			
PR_LAST_MODIFIER_ENTRYID, Pl...		0x00080040	PT_SYSTEMTIME	01:22:29 24.01.2011	Low: 0xC026CCE0 High: 0x01CBBBC9			
PR_LAST_MODIFIER_NAME, Pl...	PR_LAST_MODIFIER_NAME_W, PidT...	0x3FFB0102	PT_BINARY	cb: 131 lp: 00000000DC740C8C0...	...b\$@MAB,rlNv,+F,.../0=F1...	Exchange Address		
PR_LAST_MODIFIER_NAME_W, Pid...		0x3FFA001E	PT_STRING8	aragel@robobox.ru	cb: 17 lp: 16726167656C40696E66...			
PR_LOCAL_COMMIT_TIME, PidTa...		0x67090040	PT_SYSTEMTIME	01:23:30 24.01.2011	Low: 0xE4458F1E High: 0x01CBBBC9			
PR_LOCALE_ID, PidTagLocaleId...		0x66A10003	PT_LONG	1033	0x409			
PR_MAPPING_SIGNATURE, PidTa...		0x0FF80102	PT_BINARY	cb: 16 lp: E352E08F2CA2D4EA91...	rRaU,ÿN=O,\$.i<+			
PR_MDB_PROVIDER, PidTagStor...		0x34140102	PT_BINARY	cb: 16 lp: 549441C0297F101BA58...	T'ÝA)l,l'!,+*%.			
PR_MESSAGE_CLASS, PR_MESSA...	PR_MESSAGE_CLASS_W, PidTagMes...	0x001A001E	PT_STRING8	IPM.Microsoft.OSC.ContactSync	cb: 29 lp: 49504D2E4D6963726F73...			
PR_MESSAGE_CODEPAGE, PidTag...		0x3FFD0003	PT_LONG	1252	0x4E4			
PR_MESSAGE_FLAGS, PidTagMes...		0x0E070003	PT_LONG	1097	0x449			
PR_MESSAGE_SIZE, PidTagMessa...	PR_MESSAGE_SIZE_EXTENDED, Pid...	0x3FF10003	PT_LONG	1033	0x409			
		0x0E080003	PT_LONG	250	0xFA			

Рис. 6

ем к нему **hard delete (unrecoverable)**. Закрываем MFCMAPI, перезапускаем Outlook, закрываем Outlook, открываем в MFCMAPI и смотрим на воссозданное значение и отмечаем правильную работу кэширования.

 **Производить восстановление сообщений из дампстера часто удобнее, чем через Outlook**

Ниже я покажу, чем конкретно удобнее. Переходим в **Top of Information Store** → подсвечиваем нужную папку → в контекстном меню выбираем → **Open Deleted Contents Table** (рис. 4).

Перед нами содержание дампстера для выбранной папки. Можно выбрать нужные сообщения, в контекстном меню выбрать **Copy** и вставить их в любую нужную папку через опцию **Paste**. Outlook такой гибкости не предоставляет.

 **Восстановление сообщений и копирование из папки в папку также удобно производить с помощью MFCMAPI**

Эта операция совершенно не сложна. Необходимо только помнить о двух ве- щах: выбирая опцию **Open Contents table**, мы получаем вывод актуальных сообщений, содержащихся в папке, а когда выбираем **Open Deleted Content Table** – содержимое дампстера. В любом из двух сценариев сообщения отлично копируются и вставляются в любую папку по вашему выбору.

Количество сообщений в ящике, или количество удаленных сообщений в ящике – все это удобно смотреть через MFCMAPI (да, через Powershell это делается быстрее и красивее, но иногда мы хотим знать не только цифры) (рис. 5).

Свойства каждого сообщения содержат отметку о том, кем последний раз производилось изменение. Крайне полезная вещь для того чтобы установить, например, кем же в ящике с общим доступом было удалено важное

Top								
To	Subject	Conversation ID	Row Type	Content Count	Content Unread	Depth	Received	Submitted
Келина Ирина Казими...	Err:0x004010...	1 (TBL_LEAVE_R...	PT_BINARY	cb: 44 lp: 290CF29E54C0444CB5C...	,rTADLpKo3R,б,4=FrRaU,ÿN...		07:25:22 15.04...	IPM.Appointment 6341
Корнилова Юлия Кондро...	FW: GB	Err:0x004010...	PT_BINARY	cb: 16 lp: E352E08F2CA2D4EA91...	rRaU,ÿN=O,\$.i<+		09:03:16 14.04...	IPM.Appointment 8205
Яхти... Алексей Бахти...	Предложение ГБ	Err:0x004010...	PT_BINARY	cb: 16 lp: 549441C0297F101BA58...	T'ÝA)l,l'!,+*%.		09:43:58 11.04...	IPM.Appointment 7324

Property Name(s)								
To	Subject	Conversation ID	Row Type	Content Count	Content Unread	Depth	Received	Submitted
Келина Ирина Казими...	Err:0x004010...	1 (TBL_LEAVE_R...	PT_BINARY	cb: 2	0	0		
Корнилова Юлия Кондро...	FW: GB	Err:0x004010...	PT_BINARY	3	0	0		
Яхти... Алексей Бахти...	Предложение ГБ	Err:0x004010...	PT_BINARY	0	0	0		
<hr/>								
Property Name(s)	Other Names	Tag	Type	Value	Drag the cursor around the area you want to capture.	Smart View		
PR_ICCS_CHANGE_KEY		0x00400003	PT_LONG	2	0:3			
PR_IMPORTANCE, PidTagImporta...		0x00400003	PT_LONG	3	0:4			
PR_INTERNET_CID, PidTagInter...		0x0051001E	PT_STRING8	(UTC+03:00) Moscow, St. Petersburg, Volgograd	cb: 45 lp: 285544328033A3030...			
PR_LAST_MODIFICATION_TIME, ...		0x00520003	PT_STRING8	0	0:0			
PR_LAST_MODIFIER_ENTRYID, Pl...		0x00530102	PT_BINARY	cb: 17 lp: 763243616C656E646174C4F676769E67	vCalendarLogging			
PR_LAST_MODIFIER_NAME, Pl...		0x00540101E	PT_STRING8	SoftDelete	cb: 10 lp: 536f667444656C657465			
PR_LAST_MODIFIER_NAME_W, Pid...		0x00550102	PT_STRING8	0:000000000026187F41302C4FA044B05E74AD...	...@=,O=D^t-,rRaU,ÿN=...			
PR_MESSAGE_CODEPAGE, PidTag...		0x00570040	PT_STRING8	10:40:48 08.04.2011	cb: 6:8:256lp: 549441C0297F101BA58...			
PR_MESSAGE_FLAGS, PidTagMes...		0x00570040	PT_STRING8	06:28:08 17.10.2011	Low: 0xC0B8C41 High: 0x01C9C935			
PR_MESSAGE_SIZE, PidTagMessa...		0x00580001E	PT_STRING8	/First Organization=Exchange Administrative Group...	cb: 122 lp: 2FAFD46697273749...			
		0x00590001E	PT_STRING8	Client=BEIChangeRPC	cb: 20 lp: 49504D2E4D6963726F73...			
		0x005A0001E	PT_STRING8	0	0:0			
		0x005C0001E	PT_STRING8	0:0.0.0.0	cb: 7 lp: 302E02E0302E30			
		0x005D0001E	PT_STRING8	Microsoft.Exchange.PipClientAccess.Service	cb: 42 lp: 4069372F736F66742...			
		0x005E0001E	PT_STRING8	HUB2010-01	cb: 10 lp: 465542323013020301			
		0x005F0001E	PT_STRING8	14:00:00 2010-02-01:00:00:00	cb: 28 lp: 549441C0297F101BA58...			
		0x00600003	PT_STRING8	Version 14.1 (Build 210.0)	cb: 26 lp: 566f7272694F4E20314...			
		0x00610001E	PT_STRING8	0	cb: 139 lp: 2FAFD46697273749...			
		0x00630003	PT_STRING8	0	0:0			
		0x00640003	PT_STRING8	0	0:0			
		0x00650102	PT_BINARY	cb: 70 lp: 00000000026187F41302C4FA044B05E74AD...	...@=,O=D^t-,rRaU,ÿN=...			
		0x00660002	PT_J2	0	0:0			
		0x00660002	PT_J2	0	0:0			
		0x00660002	PT_J2	0	0:0			
		0x00660002	PT_J2	0	0:0			

Рис. 7

сообщение. **Top of Information Store** → подсвечиваем нужную папку → открываем (в зависимости от ситуации) либо **Contents Table** либо **Deleted Contents Table**.

Подсвечиваем искомое сообщение и смотрим в значение **PR\_LAST\_MODIFIER\_NAME** – опять все перед глазами (рис. 6).

 **Элементы календаря логируются специальным механизмом Exchange Server 2010**

Посмотреть детали удаленных элементов можно с помощью MFCMAPI – удивительно как много интересного содержит в себе эта информация. Увидеть данную информацию можно так: открываем папку, содержащую нужный элемент методами, описанными выше. Находим нужное сообщение и в нижнем окне ищем свойства, показанные на рис. 7. Как четко видно, информация содержит и протокол, и метод удаления, и сервера, участвовавшие в операции.

Этого более чем достаточно, когда требуется, например, установить что какой-то мобильный девайс почему-то что-то ошибочно удалил. Когда-то устройства

на iOS этим страдали.

Подытоживая – показанные примеры иллюстрируют широкий спектр применения MFCMAPI для траблшутинга клиентских проблем Exchange Server. Лично я, будучи страстным фанатом этого инструмента, уверен, что нет ничего такого, что он не умеет. Его создатель, Стивен Гриффин, даже пообещал мне включить в следующий билд coffee brewing функцию, и с MFCMAPI почему-то это не выглядит несбыточным. ■





# ВЫБОР РЕДАКЦИИ



# Интервью с Дмитрием Сошниковым

**координатором  
академических программ  
департамента  
стратегических технологий  
Microsoft Россия**

Беседовал Станислав Горнаков



**Станислав Горнаков:** Здравствуйте, Дмитрий! Спасибо, что согласились дать интервью нашему журналу.  
**Дмитрий,** вы пять лет назад пришли в Microsoft, будучи доцентом МАИ. Расскажите, пожалуйста, от кого поступило предложение работать в Microsoft, и почему решились на этот шаг?

**Дмитрий Сошников:** В Microsoft позвали мои друзья, которые в то время занимались работой с вузами – Андрей Терехов и Александр Гаврилов. Освобождалась позиция, а я был с ними к тому времени уже хорошо знаком – руководил командами МАИ, которые добивались успеха на Imagine Cup, разрабатывал курс «Многоагентные системы» для репозитория учебных курсов Microsoft, организовывал на базе МАИ открытые лекции и т. д. Вполн

не логичным казалось продолжить аналогичную деятельность уже изнутри Microsoft.



**Кстати, а как правильно называется ваша позиция в Microsoft?**

По-английски – Academic Developer Evangelist. По-русски слово «евангелист» не всегда адекватно воспринимается, поэтому в зависимости от контекста я обычно его перевожу как «координатор академических программ» или как «эксперт по разработке ПО».



**Сомнения при переходе из МАИ были или особо не раздумывали, поскольку звали работать в Microsoft?**

До этого моя жизнь состояла как бы из двух частей: бизнес-составляющая, связанная с разработкой ПО, и академическая часть – преподавание, наука. Мне показалось, что в Microsoft получится объединить эти две части и заниматься работой с вузами и студентами «в полную силу». Сомнения были, поскольку пришлось в результате забросить деятельность, связанную с разработкой, но на тот момент принял решение о переходе, и до сих пор не жалею.



**Будучи доцентом, вы естественно преподавали студентам тот или иной курс, в принципе и сейчас занимаетесь тем же, но только со сцены различных мероприятий. А все же, сложно было перестроиться и насколько сильно отличается то, чем занимались в МАИ, и то чем занимаетесь сейчас? ▶**

## Наше досье

В 1991 году поступил в Московский авиационный институт, после чего сразу уехал на два года учиться в Италию, в United World College of the Adriatic, который окончил в 1993 г., получив диплом международного бакалавра.

С 1993 по 1999 г. продолжал учиться в МАИ, в 2002 г. окончил аспирантуру и защитил диссертацию на степень к.ф.-м.н. по теме «Методы и средства распределенного представления знаний на основе распределенной фреймовой иерархии».

Параллельно работал в области разработки ПО в российском представительстве компании Partners in Computing Services International, LLC, возглавлял российское отделение разработки.

В 2006 году был приглашен на работу в Microsoft, где стал заниматься программами взаимодействия с вузами, в особенности программой студентов-партнеров и конкурсом Imagine Cup. Помимо студенческих программ, специализируется в области функционального программирования и языка F#, а также разработки игр на платформе XNA.

Является доцентом МАИ и Высшей школы экономики, автором книг «Функциональное программирование на F#» и «Парадигма логического программирования» и более 60 научных и публицистических статей.

Основное отличие – это масштаб. Работая в Майкрософт можно познакомиться со многими вузами России, поговорить с разными студентами, научить их – пусть и за короткий срок – чему-то важному и полезному. Помимо очевидных рассказов про торжество технологий Майкрософт я стараюсь в своих выступлениях говорить про другие важные вещи: тенденции развития информационных технологий, почему и как ускоряется прогресс. Даже иногда рассказываю про эволюционную кибернетику Турчина и принцип метасистемного перехода. Считаю, что студентам полезно задумываться над такими вещами.

**Вы по-прежнему читаете какие-то курсы МАИ или сейчас полностью сконцентрировались на работе в Microsoft?**

Я не бросаю работу в МАИ, и до сих пор совместно со своими учениками читаю там курс «Логическое программирование и искусственный интеллект», а также недавно поставил и читаю курсы функционального программирования в НИУ ВШЭ и на Физтехе.

**Что входит в круг ваших непосредственных обязанностей и какое конкретно направление вы курируете в Microsoft?**

С одной стороны, я курирую студенческие программы, направленные на технических студентов – потенциальных разработчиков. Это конкурс Imagine Cup, программа студентов-партнеров и т. д. С другой, как евангелист, я специализируюсь на языке F#, на разработке игр и платформе XNA, отчасти на программировании сенсора Kinect и других вещах, которые интересны студентам и исследователям.

**Знаю, что кроме вас еще так же занимается академическим направлением евангелист Константин Кичинский. Расскажите, пожалуйста, сколько человек в академической команде Microsoft кроме вас и кто чем занимается?**

Как такое понятие «академической команды» сейчас очень размыто – многие евангелисты занимаются работой со студентами: это и Костя Кичинский, и Стас Павлов. За работу с младшим поколением (школьниками) отвечает Екатерина Лажинцева (это легендарная женщина, которая придумала перевод названия кнопки «Пуск» в Windows и кодировку Win-1251), за взаимодействие с преподавателями – Саша Гаврилов. Много коллег отвечают за маркетинговую поддержку всех студенческих начинаний (Анна Марынина, Маша Аникина), за работу с сообществами (Оксана Костина-Паничева)... всех сложно перечислить! Руководит всем направлением работы с техническими аудиториями (включая студентов) Владимир Габриель.

**В октябре месяце, 14 числа, прошел очередной «День студента Microsoft 2011». Фоторепортаж с этого мероприятия все читатели найдут сразу же после интервью с Дмитрием. Это мероприятие было организовано академической командой Microsoft и все было сделано на высшем уровне, но скорее всего есть и какие-то интересные, а может и курьезные моменты, о которых мы не знаем. Расскажите, пожалуйста, что-нибудь интересное из-за кулис организации дня студента.**

Для нас организация дня студента бы-



ла грандиозной задачей, и несмотря на это – многие вещи делались в последний момент. Возможно, кто-то помнит момент, когда в ходе выступления наша студенка-партнер Татьяна Бойцова утопила телефон Windows Phone 7 в воде. Это решение было принято буквально накануне выступления, и пришлось на ходу придумывать сценарий, как это красиво сделать. К слову, телефон продолжал работать после двух часов, проведенных в воде!

Из других скандальных подробностей – надеюсь, это было незаметно, но у меня в день мероприятия была температура 38. Но «show must go on»!

**Microsoft достаточно серьезно курирует студенческое направление, понимая, что именно в этом возрасте и на этой стадии нужно «взращивать» будущее поколение, которое со временем придет в компанию. В итоге очень много MSP «перебираются» работать в Microsoft. Расскажите, пожалуйста, о студенческих программах Microsoft.**

Наша основная цель – не набрать рабочую силу для Майкрософт (особенно с учетом того, что как таковых разработ-



чиков в России работает очень мало, по сравнению с другими компаниями, ведь почти вся разработка ведется в штаб-квартире в США), а стимулировать рост экосистемы разработки ПО в России. Нам важно, чтобы студенты выходили из университетов со знаниями технологий Майкрософт, и шли работать к нашим партнерам, или создавали свои высокотехнологичные стартапы. Поэтому мы разделяем наши программы по работе со студентами на следующие этапы:

**1. Получение доступа к ПО.** Мы стараемся предоставить всем студентам бесплатный доступ почти ко всем программным продуктам Майкрософт (за исключением, пожалуй, Microsoft Office, для которого есть очень недорогая студенческая редакция). Это такие программы, как Dreamspark (на сайте DreamSpark.ru можно получить доступ ко всем средствам разработки, дизайна и администрирования по предъявлению скана студенческого билета) и MSDN AA (подписка, которая работает через вуз).

**2. Обучение технологиям** – тут есть масса мероприятий в вузах, встречи студенческих User Groups, а также онлайн-ресурсы. Помимо известных MSDN/TechNet и TechDays.ru мы также силами сообщества MSP поддерживаем студенческий сайт – на нем студенты могут прочитать, как те или иные технологии Майкрософт могут помочь им в конкретных учебных курсах, и с каких ресурсов лучше начать изучение.

3. Мы считаем, что очень важно для студента **проявить себя** и начать что-то делать – для этого мы предлагаем, прежде всего, конкурс Imagine Cup, в котором надо придумать идею, поработать в команде, сделать прототип проекта и выгодно представить его жюри. Это очень «жизненный» конкурс, но позволяет студентам получить те навыки, которые ему трудно получить в вузе (многие студенты впервые учатся представлять свою работу на дипломе, да и жанр групповой командной работы не очень развит в вузах), и которые при этом очень пригодятся в жизни.



4. В результате всего этого студент становится более успешным в жизни – как при попытке найти работу в компании на рынке труда, так и в реализации своих высокотехнологичных идей. В обоих случаях мы стараемся помочь наиболее талантливым и активным студентам и победителям Imagine Cup развивать свои проекты и навыки.

И, конечно, есть программа студентов-партнеров Майкрософт (MSP) – это армия наших друзей, мини-евангелистов в вузах, которые помогают нам общаться с миллионной аудиторией технических студентов со всех концов страны.

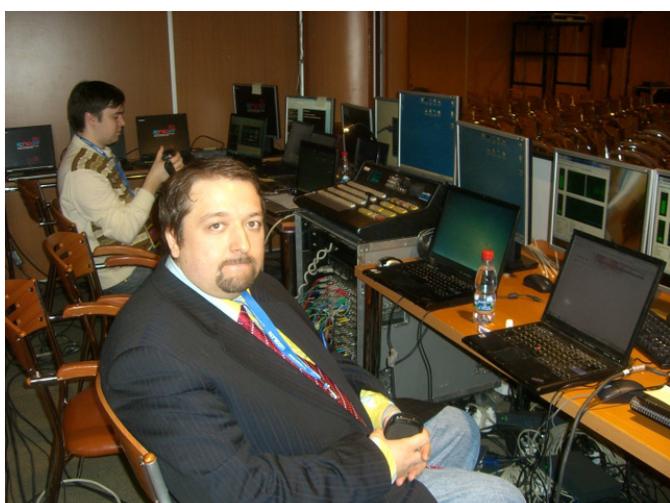
придумывая и проводя с нашей помощью какие-то интересные активности в вузе. Каждый из 100 MSP планирует свою деятельность и получает от нас ресурсы на воплощение своих планов в жизнь. Иногда эти ресурсы могут быть очень неплохими: например, телефон Windows Phone 7 или ноутбук SONY Vaio. В этом году мы также расширили программу на студентов гуманитарных факультетов и специальностей – им ведь тоже надо понимать, что Майкрософт может сделать их жизнь счастливее!

Что касается прошедшего MSP Summit – мне кажется, мы достигли цели! Нам удалось собрать почти все 100 человек в Москве, в результате все ребята уехали из Москвы с желанием помочь российскому образованию и своим коллегам по учебе в понимании того, как важно знать современные передовые информационные технологии. Вместе с ними мы надеемся поднять уровень счастья студентов России на принципиально новый уровень!

**Недавно был осенний набор MSP и MSP Summit 2011, как он прошел, и сколько претендентов в итоге было отобрано?**

Мы сильно расширили программу MSP в этом году, набрав около 100 MSP и еще около 50 MSP-волонтеров по всей России. В результате более 80 вузов по России охвачено программой, и в каждом вузе и на крупных факультетах есть кто-то из активных студентов, кто не дает своим однокурсникам «скучать»,

**Скажите, а что вообще должен сделать студент, чтобы подать заявку на MSP? ▶**





Подать заявку просто – процедура описана на <http://microsoft.ru/msp>. Заявка – это пара страниц ответов на вопросы и ссылка на трехминутное видео, которое каждый кандидат должен снять. Это позволяет нам оценить презентационные навыки и общий уровень «драйвовости» и энергетики человека. Хотя самое главное тут не столько подать заявку, сколько понять для себя, зачем это нужно, и быть готовым делать что-то хорошее с нашей поддержкой. Нам очень важен этот энтузиазм и желание сделать мир лучше, именно таких людей «с горящими глазами» мы стараемся найти!

Я с удовольствием общаюсь с потенциальными MSP в твиттере, поэтому задавайте вопросы, разбирайтесь, нужно ли вам это – и подавайте заявку. Это можно сделать в любое время в течение года!

**К слову в этом номере журнала есть интервью с Яной Курановой, которая стала новым MSP. К сожалению, у нас девушки в IT-секторе большая редкость, почему?**

Не все выдерживают... На самом деле, я думаю, есть особенность человеческой психики, и мужчины больше любят властвовать, доминировать – а это ведь часто проявляется в профессии программиста, который имеет полную власть над создаваемой им программной системой.

Для женщин это менее специфично,

поэтому меньше женщин стремятся стать программистами. Но я не хочу сказать, что женщина не может быть программистом – к сожалению, такое мнение бытует, и с ним надо бороться.

**А в целом, как вы считаете, в индустрии ощущается нехватка специалистов в IT-сфере?**

Это подтверждают многочисленные исследования рынка труда в IT-сфере. На активности студентов, кстати, это очень плохо сказывается – поскольку и хороших и плохих студентов-разработчиков все равно скорее всего возьмут на хорошую высокооплачиваемую работу, поэтому теряется стимул быть лучшим, делать что-то дополнительно к стандартному образованию для саморазвития. А ведь, как говорит пословица: «научить нельзя, можно только научиться» – и это особенно верно в IT-сфере, где система образования не успевает (да и не должна успевать) за всеми новомодными изменениями в технологиях, при этом от хорошего специалиста ожидают высокий уровень технологической эрудиции.

**Давайте поговорим о конкурсе Imagine Cup. Расскажите, пожалуйста, подробней о самом конкурсе Imagine Cup, что это за конкурс, зачем он нужен и почему студентам в нем стоит участвовать.**

Подробнее о конкурсе можно прочитать на [imaginecup.ru](http://imaginecup.ru), вряд ли стоит здесь пересказывать все детали. Остановлюсь на основных особенностях.

В отличие от олимпиад по программированию, где тестируются от части искусственные «спортивные» навыки, в Imagine Cup развиваются и проверяются навыки реальные – умение работать в команде, умение хорошо рассказать о сделанном проекте, умение завершать проекты в срок. Эти навыки пригодятся как при работе в компании, так и для создания своего стартапа.

На самом деле, есть два подхода к тому, зачем участвовать в Imagine Cup. Можно воспринимать это как очень развивающий спорт, и создавать проекты специально для конкурса. Некоторые команды несколько лет подряд участвуют, развивают свои навыки, пока наконец не добиваются победы. Шансы победить у российских студентов, кстати, весьма высоки – все проекты очень на уровне, и почти каждый год в той или иной категории российские студенты получают призовые места.

Второй подход – это если вы уже занимаетесь каким-то проектом, то его надо обязательно подавать на Imagine Cup. Опыт показывает, что почти любой осмысленный проект может быть подан на конкурс, и иногда такие проекты также выигрывают (например, команда Onicrafter).

А даже если и не выигрывают – получают много полезного опыта, PR, и созданный прототип продукта. Конкурс очень мотивирует что-то делать! ▶



В свое время в 2005 году вы с командой завоевали 2-ое место в мире на Imagine Cup 2005 в категории Office Designer. Очень хотелось бы узнать, что называется из первых рук, как тогда все было и как получилось взять второе место.

В нашем случае сработал как раз «второй вариант» – мой аспирант разрабатывал концепцию структурного моделирования интеллектуальных бизнес-процессов, и проект по сути дела стал демонстрационным прототипом концепции для его диссертации. При этом конкурс помог привлечь к проекту студентов, которых не очень интересовало занятие «чистой наукой», но зато участие в конкурсе их мотивировало. Ну и почему-то удалось победить, тут уж отчасти повезло, а отчасти у нас действительно уже был опыт участия в конкурсе в 2004 году.



Текущий Imagine Cup взял свой старт, что посоветует командам с высоты своего опыта?



Недавно у вас вышла новая книга по языку программирования F#. Расскажите о самой книге, а также о языке F#. Вы считаете у него есть будущее на фоне того же C# или C++?



Знаю, что кроме F# вы еще серьезно увлекаетесь XNA и Kodu. С XNA все понятно, технология нашла свое применение на платформе Windows Phone 7. Что скажите по Kodu? Какова перспектива этой технологии?

Вне зависимости от того, удалось ли вам придумать гениальную идею и собрать супер-команду – участвуйте! Вы получите массу интересных впечатлений! А если удастся дойти до финала – проведите часть летних каникул в Австралии за счет Майкрософта. Разве это не отличный способ провести лето?

F# – это мультипарадигмальный язык, который стимулирует программирование в функциональном стиле. Использование функционального программирования в индустрии дискутируется давно, но сейчас все острее возникает в этом необходимость из-за важности параллельного программирования для многоядерных процессоров и асинхронного веб-программирования. F# очень хорошо справляется с такими задачами. В целом можно сказать, что F# позволяет больше думать и меньше писать кода. Если вам нравится работать головой, а не пальцами – вы оцените красоту этого языка!

Сейчас на русском языке есть две книги по F# – моя, и переводная книга Криса Смита. Мне не хотелось дублировать существующие книги, которые достаточно полны и исчерпывающи; скорее мне хотелось создать небольшую книгу, которая позволит быстро понять основные принципы и начать использовать F# в реальных задачах. Поэтому книга получилась действительно компактная, и как результат – недорогая и легкая в прочтении. Почти половину книги составляют реальные примеры использования F# для практических задач – разбор XML-файла, построение графиков, работа с Excel-таблицами и SQL, программирование для облака и телефона Windows Phone 7. Используя эти примеры кода как заготовки, вы сможете очень быстро собрать из них решение своих задач!



У меня растет дочь, которой сейчас 7 лет (кстати, ее зовут Вики – в честь Википедии, и вы можете почитать ее твиттер @vickievalerie), и одна из основных задач, которую я перед собой ставлю – это показать ей, что компьютер – это не ящик для игр, а инструмент для творчества. Kodu прекрасно справляется с этой задачей, позволяя создавать программируемые миры, причем как на компьютере, так и на Xbox. Это первый шаг в понимании места компьютера в нашей жизни, после которого можно переходить к Small Basic, C# и другим языкам программирования.

Кстати, мы с Викторией начали было выпускать детский подкаст, в котором рассказываем о том, как использовать компьютер для развития детского потенциала. Можете нас послушать на <http://childcast.rpod.ru/> и написать отзывы – если дело стоящее, мы будем продолжать его делать.



Дмитрий, спасибо за интересное интервью, приятно было пообщаться, надеюсь, вы нас читаете.

Спасибо за отличную инициативу по созданию такого журнала! Верю, что его читают не только я, но и миллионы русскоязычных разработчиков со всех концов планеты! ■

# Шпаргалка по сертификации приложений на Windows Phone 7

**Автор**

Филипп Панфилов  
 ponf@inbox.ru  
<http://mne.p0x.ru>  
 MSP в МГТУ им. Н.Э.Баумана  
 г. Москва

Представлю вашему вниманию небольшую шпаргалку с основными требованиями к сертификации приложений в Marketplace. Данная шпаргалка основана на статье Peter Kuhn «[Windows Phone 7 Application Certification Cheat Sheet](#)», которую я немного дополнил и внес некоторые изменения, связанные с нововведениями в Marketplace. Разумеется, в рамках одной статьи не раскрыть все пункты, поэтому я постарался выбрать те требования, на которые стоит обратить особое внимание перед публикацией приложения.

**Приложение в целом**

Перечислены некоторые основные требования к приложением:

- ▼ В приложении для пользователя должны быть доступны: название, номер версии и контакты техподдержки ([5.6](#));
- ▼ Приложение должно быть полностью работоспособным после покупки в Marketplace (не может требовать какой-то дополнительной оплаты для активации, разблокировки и т. п.) ([2.1](#));
- ▼ Приложение не должно продавать или рекламировать тарифные планы мобильных операторов ([2.2](#));
- ▼ Если приложению для работы требуется скачать большой объем данных, оно должно предупреждать об этом в описании ([2.5](#));
- ▼ Вся реклама в приложении должна соответствовать [Microsoft Advertising Creative Acceptance Policy Guide](#) ([2.4](#));
- ▼ Если в приложении предусмотрены возможности IM, то приложение должно удостовериться в том, что пользователю больше 13 лет ([2.6](#));
- ▼ Если в приложении планируется использовать личную информацию о пользователе, то необходимо: ознакомить пользователя с вашей политикой конфиденциальности, спросить у пользователя разрешение на использование его данных и предоставить механизм, благодаря которому пользователь сможет в любой момент запретить использование его личной информации ([2.8](#));

▼ Приложение должно быть чем-то большим, чем просто запуск веб-страницы ([2.10](#));

▼ Разработчик должен убедиться, что все элементы управления видны и легко читаемые со всеми темами телефона (темной/светлой) ([5.5.2](#)).

**XAP пакет (4.1.1)**

Следующие требования проверяются во время загрузки приложения в AppHub. Процесс включает в себя проверку метаданных и проверку XAP пакета, который вы загружаете.

▼ Максимальный размер: 225 МБ;

▼ Требуемое содержимое:

- Файл WMAppManifest.xml;
- Файл AppManifest.xml;
- Файлы для сборки, указанные в AppManifest.xml;
- Большая (173x173) и маленькая (62x62) иконки приложения.

▼ Требования к WMAppManifest.xml:

- Поле названия не должно быть пустым;
- Название должно совпадать с названием, указанным при загрузке приложения в Marketplace.

Пожалуйста, учтите, что эти требования иногда меняются. Так, ранее размер пакета не должен был превышать 400 МБ.

**Проверка кода**

Чтобы успешно пройти проверку кода, приложение должно соответствовать следующим требованиям:

▼ При разработке приложения необходимо использо- ►

вать только документированный API ([4.2.1](#));

- ▼ Приложение не должно вызывать нативный код с помощью P/Invoke или COM-совместимостей ([4.2.2](#));
- ▼ Приложение должно быть собрано в режиме Release; не должна содержаться отладочная информация ([4.2.3](#));
- ▼ Приложение не должно использовать XNA Game или Graphics одновременно с System.Windows.Controls ([4.2.5](#)).



## Возможности телефона ([4.3](#))

Список возможностей, которые использует приложение, содержится в файле WMAppManifest.xml и автоматически определяются во время отправки приложения в Marketplace. При этом не обнаруживаются функции, вызывающиеся с помощью .NET рефлексии. Таким образом, в манифесте приложения не будут перечислены необходимые возможности, и приложение не пройдет сертификацию.

- ▼ Приложение должно работать с использованием автоматически распознаваемых возможностей. Майкрософт рекомендует вам использовать [Capability Detection Tool](#) для проверки работоспособности с возможностями, распознанными во время отправки приложения ([4.3](#)).



## Языки

Основное требование к приложению:

- ▼ Каждое приложение должно поддерживать хотя бы один язык, доступный в Marketplace ([4.4](#)).

Подробнее со списком поддерживаемых языков можно ознакомиться по ссылке: [Culture and Language Support for Windows Phone](#).

Так же есть дополнительные требования:

- ▼ Должен быть установлен нейтральный язык ([4.4](#));
- ▼ Для каждого поддерживаемого языка должны быть переведены описание и пользовательский интерфейс ([5.5.1](#));
- ▼ Все сообщения в программе должны быть на одном языке, либо должно присутствовать предупреждение, что текст данного сообщения представлен только на определенном языке (например, вывод сообщения о возникновении исключения).



## Иконографика/изображения ([4.5](#), [4.6](#), [4.7](#))

Обсуждение иконографики и интерфейса Metro UI заслуживает даже не одной отдельной статьи, а здесь же я приведу краткую информацию по требованиям к изображениям в Windows Phone 7 (см. таблицу).

Из необозначенного в таблице, стоит обратить внимание на то, что тайлы и иконки должны соответствовать приложению ([4.7](#)), а в скриншоты приложения не должны попадать части

Изображение	Обязательно	Формат	Прозрачный фон	Размер
Маленькая иконка приложения	Да	PNG	Да	62x62
Большая иконка приложения	Да	PNG	Да	173x173
Маленькая иконка для Marketplace	Да	PNG	Нет	99x99
Большая иконка для Marketplace	Да	PNG	Нет	173x173
Большая иконка для Marketplace для РС	Да	PNG	Нет	200x200
Задний фон	Нет	PNG	Нет	1000x800
Скриншот приложения	Мин. 1, макс. 8	PNG	Нет	480x800
Иконка для Application Bar	Нет	PNG	Да	48x48 (Содержимое 26x26)

эмулятора (например, рамка) ([4.6](#)).



## Временные рамки

Присутствуют следующие специальные временные ограничения:

- ▼ Приложение должно появляться на экране в течение пяти секунд (экрана заставки достаточно) ([5.2.1 - 5.2.3](#));
- ▼ Приложение должно быть доступно для пользовательского ввода не позднее, чем через 20 секунд после запуска ([5.2.1 - 5.2.3](#));
- ▼ При блокировке интерфейса более чем на 3 секунды необходимо показывать **Progress Bar** или **Busy Indicator** ([5.1.3](#)).



## Кнопка «Назад»

При использовании кнопки **Назад** необходимо учитывать следующие требования:

- ▼ При нажатии на кнопку **Назад** приложение должно переходить на предыдущую страницу ([5.2.4](#), [5.2.4.1](#));
- ▼ Нажатие должно закрывать диалоговое окно/контекстное меню и приложение должно оставаться на текущей странице ([5.2.4.3](#));
- ▼ Если нажатие происходит на главной странице приложения, то оно должно закрыться ([5.2.4.2](#));
- ▼ Для игр: во время игры нажатие на кнопку **Назад** может вызывать меню паузы ([5.2.4.4](#)).

Кнопку **Назад** нельзя использовать для каких-либо функций, не обозначенных выше. ►



## Надежность, безопасность и прочее

Следующие требования связаны с безопасностью, надежностью и встроенными в телефон функциями:

- Приложение должно работать на любом устройстве с Windows Phone 7 ([5.1.1](#));
- Приложение не должно неожиданно закрываться из-за необработанных исключений или по другим причинам ([5.1.2](#), [5.3.3](#));
- При обработке ошибок приложение должно предоставлять понятные пользователю сообщения ([5.1.2](#));
- Приложение не должно препятствовать использованию функций телефона (звонки, SMS/MMS) ([5.3.1](#));
- Приложение должно продолжать работу после использования функций телефона (звонки, SMS/MMS) ([5.3.2](#));
- Приложение не должно содержать вирусы, вредоносный код или вредоносные функции ([5.4.1](#));
- Приложение не должно запускать небезопасный код или пытаться обойти механизм песочницы ([5.4.2](#), [5.4.3](#));
- Приложение не должно ставить под угрозу безопасность телефона или Marketplace ([2.3](#));
- Приложение не должно использовать более 90 МБ оперативной памяти в телефонах с 256 (или менее) МБ ([5.2.5](#));
- Приложение не должно использовать Trial API в цикле (например, игровом). То есть API должен вызываться редко, а возвращенное им значение – кэшироваться ([5.2.6](#)).



## Приложения, использующие местоположение

К приложениям, использующим местоположение устройства пользователя, предъявляются следующие требования:

- Приложение должно использовать Microsoft Location Service API ([2.7.1](#));
- Информацию о местоположении можно использовать только для функций самого приложения (нельзя использовать для показа таргетированной рекламы) ([2.7.7](#));
- Приложение должно предупреждать пользователя об использовании данных о его местоположении ([2.7.2](#));
- Приложение должно предоставлять возможность отключения использования данных о местоположении ([2.7.3](#));
- Приложение должно обеспечивать возможность отказа от использования данных о местоположении, если эти данные станут доступны третьим лицам ([2.7.4](#));
- Приложение должно защищать полученные данные о местоположении от несанкционированного доступа ([2.7.8](#));
- Если сервис определения местоположения недоступен, приложение должно оставаться отзывчивым (продолжать реагировать на пользовательский ввод) ([6.1](#));



- Приложение не должно вмешиваться в системные функции, связанные с Location Service API ([2.7.5](#));
- Приложение не должно нарушать работу устройства, делая Location Service недоступным ([2.7.6](#), [6.1](#)).



## Служба Push-уведомлений

Для использования Push-уведомлений ваше приложение должно удовлетворять следующим требованиям:

- У всех используемых Push-уведомлений должно быть описание ([2.9.1](#));
- Приложение должно спросить у пользователя разрешение на использование Push-уведомлений и предоставить механизм, благодаря которому пользователь сможет в любой момент отказаться от них ([2.9.1](#));
- Запрещается злоупотреблять Push-уведомлениями ([2.9.2](#));
- Не разрешается использовать Push-уведомления в отношении вопросов жизни и смерти или других критически важных данных, изменяющихся в реальном времени ([2.9.3](#));
- Приложение должно предоставить возможность пользователю самостоятельно отключить всплывающие уведомления ([6.2.1](#));
- Приложение должно спрашивать разрешение для использования всплывающих уведомлений ([6.2.2](#)).

На этом основные требования подходят к концу. С полным перечнем требований вы можете ознакомиться по ссылке: [Application Certification Requirements for Windows Phone](#).

С развитием платформы меняются и требования к прохождению сертификации. Для того чтобы всегда оставаться в курсе изменений в требованиях, рекомендую периодически посещать страницу [Change History](#). ■

# Обзор библиотеки Kendo UI



## Автор

Валерий Воробьев  
Разработчик  
г. Москва



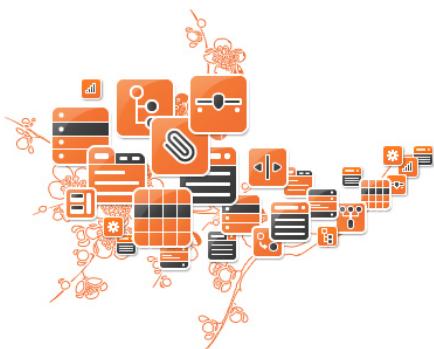
Kendo UI – это новая библиотека для разработки web приложений. Она включает в себя различные элементы управления и функции. Kendo UI основан на jQuery, а так же использует новые стандарты HTML5 и CSS3. Более подробную информацию о возможностях Kendo UI вы можете получить на сайте проекта <http://www.kendoui.com/>

Впервые столкнувшись с Kendo UI, я был приятно удивлен предоставляемыми ею возможностями. Вот перечень некоторых из них:

- Много удобных, быстрых элементов управления, таких как Grid, Autocomplete, Calendar, Chart, ComboBox, Menu и так далее;
- Обеспечивает работу с привязкой данных, создание шаблонов и анимации, позволяет реализовать Drag & Drop;
- Поддержка современных стандартов HTML5 и CSS3;
- В качестве основы используется jQuery;
- Поддержка браузеров: IE 7+, Firefox 3+, Chrome, Safari 4+, а так же планшетных устройств. Элементы управления оптимизированы под планшетный интерфейс;
- Kendo UI не является клоном jQuery UI. Все элементы управления Kendo UI были спроектированы с нуля и гораздо быстрее работают, чем jQuery UI. Это достигается за счет оптимизации Java Script сценариев, оптимизации анимации, которая использует CSS3 и аппаратное ускорение, а так же виртуализации пользовательского интерфейса.



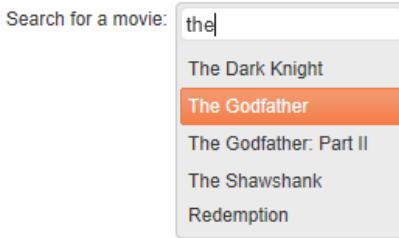
На момент написания этой статьи Kendo UI находится в стадии Beta 2. Вероятно, в финальном выпуске продукта появятся новые возможности и элементы управления.



## Элементы управления: Autocomplete (авто заполняющееся поле)

### AutoComplete

Данный элемент управления используется довольно часто для того, чтобы облегчить пользователю ввод. Работает он очень просто. В момент ввода под текстовым полем появляется список, из которого можно выбрать предлагаемые значения. Как правило, значения фильтруются по первым введенным символам.



### Пример использования

Использовать Kendo UI Autocomplete очень просто. Для начала нужно подготовить данные, затем создать элемент управления, используя стандартный синтаксис jQuery.

```
var data = [
    "12 Angry Men",
    "Il buono, il brutto, il cattivo.",
    "Inception",
    "One Flew Over the Cuckoo's Nest",
    "Pulp Fiction",
    "Schindler's List",
    "The Dark Knight",
    "The Godfather",
    "The Godfather: Part II",
    "The Shawshank Redemption"
];
<input id="input" />
$("#input").kendoAutoComplete({
    dataSource: data,
    separator: ","
});
```

При создании элемента Autocomplete мы заполняем два свойства: dataSource – данные, которые будут участвовать в авто-подборе и separator – разделительный символ. ►

## Элементы управления: Calendar (календарь)

### Calendar

Довольно часто при создании web приложений возникает необходимость реализовать работу с датой. Когда-то разработчикам приходилось самостоятельно реализовывать элемент управления «календарь», чтобы дать пользователю возможность наглядно выбирать интересующую дату. Сейчас довольно много различных вариантов «календарей» можно найти в интернете, так сказать, на любой вкус и цвет.

Kendo UI включает свой элемент управления calendar. Чтобы создать календарь средствами Kendo UI, достаточно определить html элемент с идентификатором, например div. Затем, используя синтаксис jQuery, создать элемент управления.



### Пример использования

Нужно добавить область календаря, например, включить в html разметку тег div:

```
<div id="calendar"></div>
$("#calendar").kendoCalendar();
```

## Элементы управления: Chart (график)

### Chart

Библиотека Kendo UI позволяет легко организовать работу с графиками. Чтобы создать график, вам не нужно генерировать изображения на сервере или использовать сложные библиотеки JavaScript. Для визуализации диаграмм Kendo UI использует технологию SVG (масштабируемую векторную графику), которая позволяет отображать диаграммы без потери качества при масштабировании.

В браузерах, не поддерживающих HTML5 и SVG данные возможности эмулируются, позволяя работать с диаграммами.

В Kendo Chart Beta 2 поддерживаются следующие виды диаграмм:

- ▶ **Bar** – столбиковая диаграмма;
- ▶ **Columns** – столбиковая диаграмма;
- ▶ **Line** – график;
- ▶ **Pie** – круговая диаграмма.

### Пример создания столбиковой диаграммы (гистограммы)

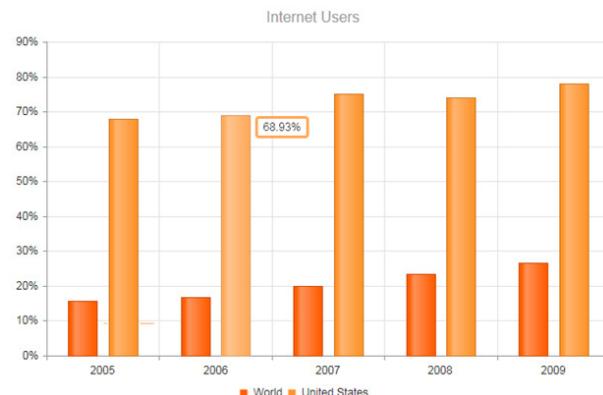
Для создания диаграммы зададим значения следующим свойствам:

- ▶ **Title.text** – заголовок диаграммы;

- ▶ **Legend.position** – место расположение легенды;
- ▶ **SeriesDefault.Type** – тип диаграммы;
- ▶ **Series** – данные для создания серий;
- ▶ **Series.name** – название серии;
- ▶ **Series.data** – числовой ряд для построения диаграммы;
- ▶ **ValueAxis.labels.format** – формат данных;
- ▶ **CategoryAxis.categories** – категории;
- ▶ **Tooltip** – задает всплывающее сообщение.

```
$(«#chart»).kendoChart({
    title: {
        text: «Internet Users»
    },
    legend: {
        position: «bottom»
    },
    seriesDefaults: {
        type: «column»
    },
    series: [
        {
            name: «World»,
            data: [15.7, 16.7, 20, 23.5, 26.6]
        },
        {
            name: «United States»,
            data: [67.96, 68.93, 75, 74, 78]
        }
    ],
    valueAxis: {
        labels: {
            format: «{0}»
        }
    },
    categoryAxis: {
        categories: [2005, 2006, 2007, 2008, 2009]
    },
    tooltip: {
        visible: true,
        format: «{0}»
    }
});
<div id=«chart» ></div>
```

Полученный результат.



### Пример создания круговой диаграммы

Похожим образом можем создать круговую диаграмму. Для этого, как и в случае с гистограммой, нужно инициализировать данные, на основе которых будет создаваться диаграмма, затем задать свойства в конструкторе диаграммы.

Подготовка данных.

```
var data = [
    {
        «source»: «Hydro»,
        «percentage»: 22
    },
    {
        «source»: «Solar»,
```



```

        «percentage»: 2
    },
    {
        «source»: «Nuclear»,
        «percentage»: 49
    },
    {
        «source»: «Wind»,
        «percentage»: 27
    }
];

```

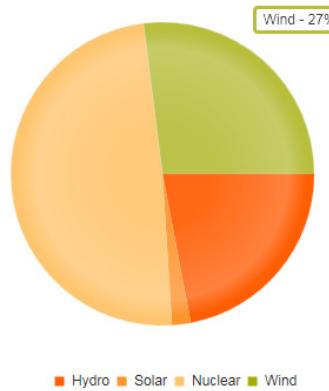
Создание круговой диаграммы.

```

$(“#chart”).kendoChart({
    theme: $(document).data(“kendoSkin”) || “kendo”,
    title: {
        text: “Break-up of Spain Electricity
        Production for 2008”
    },
    legend: {
        position: “bottom”
    },
    dataSource: {
        data: data
    },
    series: [{
        type: “pie”,
        field: “percentage”,
        categoryField: “source”,
        startAngle: 160
    }],
    tooltip: {
        visible: true,
        template: “${ category } - ${ value }%”
    }
});

```

Break-up of Spain Electricity Production for 2008



```

        «value»: 74
    },
    {
        «country»: «United States»,
        «year»: «2009»,
        «value»: 78
    }
];

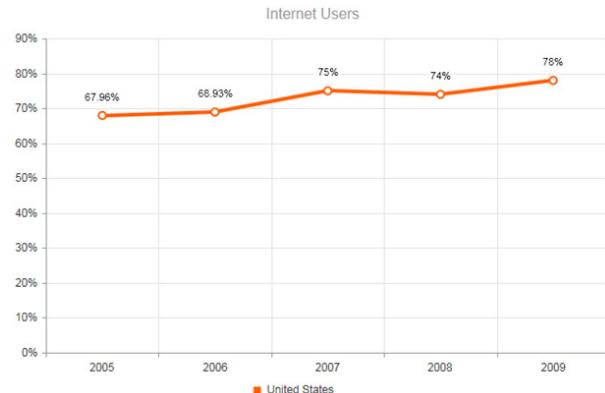
```

Создание графика.

```

$(“#chart”).kendoChart({
    dataSource: {
        data: internetUsers
    },
    title: {
        text: “Internet Users”
    },
    legend: {
        position: “bottom”
    },
    seriesDefaults: {
        type: “line”,
        labels: {
            visible: true,
            format: “{0}%
        }
    },
    series: [{
        field: “value”,
        name: “United States”
    }],
    valueAxis: {
        labels: {
            format: “{0}%
        }
    },
    categoryAxis: {
        field: “year”
    }
});

```



## Пример создания линейного графика

Инициализация данных.

```

var internetUsers = [
    {
        «country»: «United States»,
        «year»: «2005»,
        «value»: 67.96
    },
    {
        «country»: «United States»,
        «year»: «2006»,
        «value»: 68.93
    },
    {
        «country»: «United States»,
        «year»: «2007»,
        «value»: 75
    },
    {
        «country»: «United States»,
        «year»: «2008»,

```

## Элементы управления: ComboBox (поле со списком)

### ComboBox

Элемент управления ComboBox обладает большим количеством возможностей, чем классический html select. Он включает такие возможности, как привязка к локальным и удаленным данным, использование шаблонов и настройка различных параметров для управления поведением выпадающего списка.

Допустим, нам необходимо реализовать выпадающий список, который будет отображать краткую информацию о фильмах. Мы хотим, чтобы в выпадающем списке присутствовало название фильма, изображение и год выпуска. Данные будут приходить от web службы. ►

```

<input id="titles" />
  $("#titles").kendoComboBox({
    dataTextField: "Name",
    dataValueField: "Id",
    // define custom template
    template: '' +
      '<dl>' +
        '<dt>Title:</dt><dd>${ data.Name }</dd>' +
        '<dt>Year:</dt><dd>${ data.ReleaseYear }</dd>' +
      '</dl>',
    dataSource: {
      type: "odata",
      pageSize: 20,
      transport: {
        read: "http://odata.netflix.com/Catalog/Titles"
      }
    }
  });

```

В конструкторе поля со списком были заполнены следующие свойства:

- **dataTextField** – задается текст, который будет отображаться в текстовом поле при выборе элемента из выпадающего списка;
- **dataValueField** – значение, которое будет отправляться серверу;
- **template** – шаблон для отображения каждого элемента;
- **dataSource** – источник данных. Здесь заполняются следующие свойства: **type** – тип данных, **pageSize** – количество элементов, получаемых от сервера;
- **transport.read** – адрес службы <http://odata.netflix.com/Catalog/Titles>, которая возвращает данные.

В результате получим такое поле со списком.



## Элементы управления: DatePiker (выбор даты)

### DatePiker



Для быстрого выбора даты очень удобно использовать элемент управления DatePiker. Этот элемент управления представляет собой текстовое поле, в левой части которого размещена кнопка с изображением календаря.

### Пример использования

Для использования элемента управления DatePiker достаточно лишь вставить в Html разметку тег `input`, задав ему `id`. После чего создать элемент управления, используя стандартный синтаксис jQuery.

```

<input id="datepicker"/>
$(document).ready(function(){
  $("#datepicker").kendoDatePicker();
});

```

## Элементы управления: DropDownList (выпадающий список)

### DropDownList

Элемент управления DropDownList похож на элемент управления ComboBox за исключением того, что выпадающий список DropDownList не позволяет вводить данные.

## Элементы управления: Grid (сетка)

### Grid

Элемент управления Grid предназначен для отображения табличных данных. Kendo UI Grid поддерживает такие операции как: разбиение на страницы, сортировку, группировку и т. д. Он предоставляет разработчикам большой набор параметров для настройки.

Элемент управления Grid поддерживает виртуализацию данных. Что позволяет значительно повысить производительность при работе с большим объемом данных.

First Name	Last Name	Age
Robert	Fuller	53
Janet	Buchanan	74
Anne	King	53
Michael	Davolio	53
Nancy	Davolio	63
1	2	3
4		

### Пример использования

Создать элемент управления Grid можно несколькими способами – создать Grid из существующей Html таблицы или созданием сетки в конструкторе. Рассмотрим оба варианта.

Создание элемента управления Grid из существующей html таблицы:

```

<!-Создаем таблицу, указывая строки, столбцы и данные -->
<table id="grid">
  <thead>
    <tr>
      <th data-field="title">Title</th>
      <th data-field="year">Year</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Star Wars: A New Hope</td>
      <td>1977</td>
    </tr>
  </tbody>
</table>

```

```

</tr>
<tr>
    <td>Star Wars: The Empire Strikes Back</td>
    <td>1980</td>
</tr>
</tbody>
</table>
<!--Создаём элемент управления Grid --&gt;
$(document).ready(function(){
    $("#grid").kendoGrid();
});
</pre>

```

Пример второй: создание сетки в конструкторе.

```

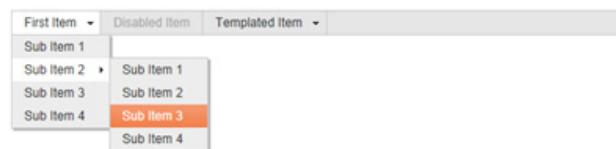
<!--Подготовим область, в которую будет помещена таблица --&gt;
&lt;div id="grid"&gt;
&lt;/div&gt;
<!--Инициализация таблицы --&gt;
$("#grid").kendoGrid({
    columns:[
        {
            field: "FirstName",
            title: "First Name"
        },
        {
            field: "LastName",
            title: "Last Name"
        }
    ],
    dataSource: {
        data: [
            {
                FirstName: "Joe",
                LastName: "Smith"
            },
            {
                FirstName: "Jane",
                LastName: "Smith"
            }
        ]
    }
});
</pre>

```

## Элементы управления: Menu (меню)



Элемент управления Menu позволяет реализовать иерархическую структуру данных. Как правило, он используется для создания горизонтального меню. Использовать его так же просто, как и другие элементы управления Kendo UI. Просто создайте Html список с нужной структурой, а затем инициализируйте элемент управления menu.



### Пример использования

```

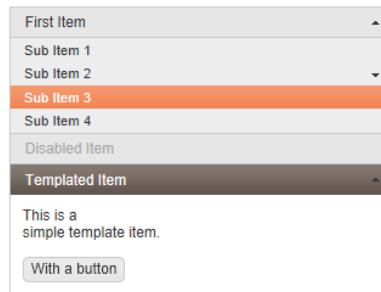
<ul id="menu">
    <li>Item 1
        <ul>
            <li>Item 1.1</li>
            <li>Item 1.2</li>
        </ul>
    </li>
    <li>Item 2</li>
</ul>
$("#menu").kendoMenu();

```

## Элементы управления: PanelBar (многоуровневая расширяемая панель)



Элемент управления PanelBar позволяет реализовать различные сценарии при разработке web приложений. Например, с его помощью можно реализовать вертикальное меню, наподобие того, что используется в программе Microsoft Outlook, сделать колонку новостей и т. д.



### Пример использования

Процесс создания PanelBar аналогичен созданию элемента управления Menu.

```

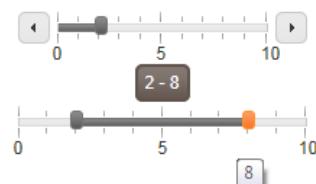
<ul id="panelbar">
    <li>Item 1
        <ul>
            <li>Sub Item 1</li>
            <li>Sub Item 2</li>
        </ul>
    <li>
        <li>Item 2</li>
        <li>Item with Content
            <div>This is some PanelBar Item content</div>
        </li>
    </li>
</ul>
$("#panelbar").kendoPanelBar();

```

## Элементы управления: Slider (слайдер)



Элемент управления Slider предоставляет выбор значения или диапазона значений. Он имеет множество настроек и событий. Использование элемента Slider позволяет более удобно взаимодействовать с вашим приложением. Работа со слайдером эффективнее и интереснее, чем простой ввод чисел в текстовое поле. Сайт становится интерактивным, что в свою очередь нравится пользователям.



Создать слайдер с помощью Kendo UI очень просто. Для этого нужно определить поле input, затем инициализировать slider с помощью jQuery.

### Пример использования

```

<input id="slider" type="range" />
$("#slider").kendoSlider();

```

Слайдер не трудно настроить. Например, чтобы задать максимальное и минимальное значение нужно всего лишь установить значения свойств в конструкторе.

```

$("#slider").kendoSlider({
    min:10,
    max:50,
    ...
})

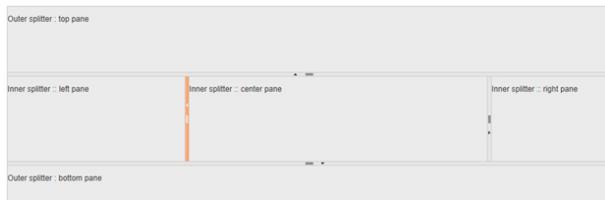
```

});

## Элементы управления: Splitter (разделитель)



С помощью элемента управления Splitter можно быстро создать страницу с областями. Каждая область может становиться шире или длиннее в зависимости от настроек разделителей. Пользователь может менять размер областей, взаимодействуя с разделительными линиями. Например, если потянуть вертикальную линию левой области вправо, то область будет расширяться. При этом содержимое области будет динамически подстраиваться под новый размер, то есть сужаться или расширяться.



Каждую область можно свернуть. Тем самым освободить больше места для других областей. Например, если на сайте есть меню и рабочая область, с которой пользователь взаимодействует большую часть времени, может быть полезно скрыть меню, тем самым дав больше пространства рабочей области, и не отвлекать взгляд на ненужный контент.

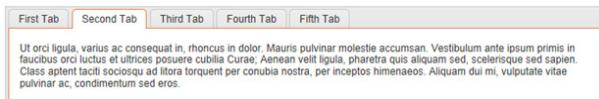
### Пример использования

```
<div id="splitter">
  <div>
    Area 1
  </div>
  <div>
    Area 2
  </div>
</div>
$("#splitter").kendoSplitter();
```

## Элементы управления: TabStrip (вкладки)



Элемент управления TabStrip отображает набор вкладок с содержимым. Вкладки представляют собой простой html список, а содержимое определяется в тегах div. TabStrip можно использовать для создания простой навигации по странице и в других случаях, когда нужно отобразить несколько разных групп данных на одной странице.



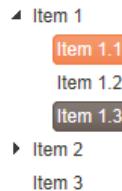
### Пример использования

```
<div id="tabstrip">
  <ul>
    <li>First Tab</li>
    <li>Second Tab</li>
  </ul>
  <div>First Tab Content</div>
  <div>Second Tab Content</div>
</div>
$("#tabstrip").kendoTabStrip();
```

## Элементы управления: TreeView (дерево элементов)



Элемент управления TreeView отображает данные в древовидном виде. Он поддерживает drag & drop для изменения порядка элементов в списке или обмена элементами между разными деревьями.



### Пример использования

```
<ul id="treeview">
  <li>Item 1
    <ul>
      <li>Item 1.1</li>
      <li>Item 1.2</li>
    </ul>
  </li>
  <li>Item 2</li>
</ul>
$("#treeview").kendoTreeView();
```

## Элементы управления: Upload (загрузка файлов)

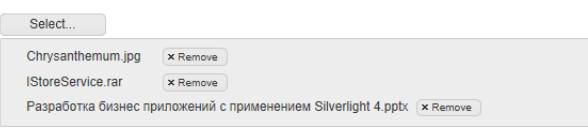


Элемент управления Upload позволяет загружать различный контент (изображение, видео, текстовые документы и т. д.) на сервер. Он предоставляет следующие возможности:

- ▼ Асинхронная и синхронная загрузка файлов;
- ▼ Множественное выделение файлов для загрузки;
- ▼ Удаление загруженных файлов;
- ▼ Отслеживание прогресса выполнения загрузки файлов \*;
- ▼ Перетаскивание файлов (Drag & Drop) \*;
- ▼ Отмена процесса загрузки файлов \*.

\* Эти функции включаются автоматически, если они поддерживаются браузером.

Самое замечательное в Kendo UI UpLoad то, что он реализован с применением стандартов Html5 и CSS3, что не требует использования подключаемых плагинов. При этом он работает в браузерах, которые не поддерживают новые стандарты.



### Пример использования

Существует два основных способа использования элемента ►

управления Upload:

- Синхронная загрузка файлов, используя Html форму и элемент input;
- Асинхронная загрузка файлов, используя Html элемент input.

Пример использования Upload для синхронной передачи файлов.

```
<form method="post" action="/Home/Upload">
    <div>
        <input name="files" id="files" type="file" />
    </div>
</form>
$("#files").kendoUpload();
```

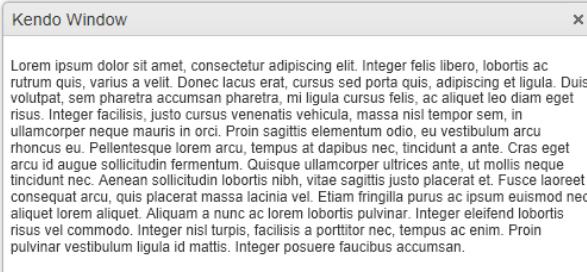
Пример использования Upload для асинхронной передачи файлов.

```
<input name="files" id="files" type="file" />
$("#files").kendoUpload({
    async: {
        saveUrl: "/ Home/Upload",
        removeUrl: "/Home/DeleteFile",
        autoUpload: true
    }
});
```

## Элементы управления: Window (окно)

### Window

Элемент управления Window позволяет отобразить некоторое содержимое в модальном или простом окне. По умолчанию окно можно перемещать по экрану и изменять его размер. Содержимое окна может быть как статической Html разметкой, так и загружено с помощью AJAX.



### Пример использования

```
<p id="window">
    Kendo window content
</p>
$("#window").kendoWindow();
```

## Kendo UI Framework: Templates (шаблоны)

### Templates

Библиотека Kendo UI предоставляет такое замечательное средство как шаблоны. Шаблоны позволяют создавать html фрагменты, которые могут динамически заполняться данными и многократно использоваться. Сам шаблон содержит html разметку и заполнители, то есть метки, которые впоследствии будут замещены данными.

Использовать шаблоны очень просто. Для этого достаточно создать html разметку, которая будет формировать шаблон,

разместить в ней специальные метки, затем, используя селектор jQuery, назначить шаблон элементу html.

### Вот такой шаблон



POSTED ON 16 ОКТЯБРЯ 2011 Г. BY VALERAVOROBIEV

Привет! Мой twitter @valeravorobjev

### Пример использования

```
var inlineTemplate = kendo.template("Hello, #= firstName # #= lastName #");
var inlineData = { firstName: "John", lastName: "Doe" };
$("#inline").html(inlineTemplate(inlineData));
```

Результат:  
Hello, John Doe!

## Kendo UI Framework: Drag & Drop (перетаскивание)

### Drag & Drop

Drag & Drop – это возможность перетаскивать любой html элемент на любой другой элемент DOM.



Now you can drop it.

### Пример использования

Инициализация перетаскиваемого элемента.

```
var draggable = $("#draggable").kendoDraggable();
```

Инициализация элемента, который будет принимать перетаскиваемый элемент.

```
var dropTarget = $("#dropTarget").kendoDropTarget();
```

## Заключение

Подводя итоги, хочется выразить огромную благодарность разработчикам библиотеки Kendo UI. Библиотека действительно получилась очень классная. Не смотря на раннюю версию (Beta 2), в ней уже присутствует большое количество качественных элементов управления. Но Kendo UI не ограничивается одними виджетами. В нее включены такие функции, как шаблоны, перетаскивание и привязка данных.

До конца 2011 года должна выйти библиотека для мобильных устройств Kendo UI Mobile. Эти элементы управления отлично подойдут для использования в смартфонах. Пока Kendo UI Mobile больше нацелена на платформы iOS и Android, но я искренне надеюсь, что и Windows Phone 7 не останется в стороне.

Kendo UI отлично интегрируется с различными JavaScript библиотеками, ее очень просто применять в ASP.NET MVC проектах. Так как по умолчанию шаблон проекта ASP.NET MVC использует jQuery, вам достаточно только добавить ссылку на библиотеку Kendo UI и таблицу стилей. Удачи! ■



Автора,

автора,

автора...

!

Команда журнала MSDeveloper.RU ищет авторов для написания различных статей, обзоров, превью и других материалов по тематике Microsoft. Если вы любите Microsoft, умеете красиво излагать свои мысли на бумаге, являетесь блогером или только, только начинаете пробовать себя в этом жанре, то обязательно свяжитесь с нами по электронной почте: [msd@msdeveloper.ru](mailto:msd@msdeveloper.ru) В качестве темы письма используйте заголовок «Статья для журнала».

#### Какие темы нам интересны:

- Тематические статьи;
- Обзоры новостей;
- Технические статьи;
- Обзоры железа;
- Программирование;
- Использование инструментов и технологий Microsoft;
- Материалы для Интервью;
- Качественные фотопортажи;
- Обзоры мероприятий;
- Эксклюзивы и сенсации;
- И многое другое.

# ASP.NET MVC + jQuery AJAX



## Автор

Валерий Воробьев  
Разработчик  
г. Москва

*Современные web приложения не обходятся без сценариев на стороне клиента. Сайт должен быть быстрым, отзывчивым и интерактивным. Многие пользователи считают, что сайты, требующие полной перезагрузки страницы устарели и ими неудобно пользоваться. Для создания web приложений отвечающих современным требованиям была создана технология Ajax.*

В этой статье я хочу рассказать об использовании AJAX средствами jQuery при разработке сайтов на ASP.NET MVC. Мы создадим проект, в котором реализуем различные сценарии использования AJAX. Я постараюсь максимально подробно описать процесс разработки на ASP.NET MVC, чтобы примеры были понятны тем, кто не знаком с этой технологией.



Для получения более подробной информации о ASP.NET MVC я рекомендую книгу Владимира Юнева и Гайдара Магданурова «ASP.NET MVC Framework».

ASP.NET MVC – это фреймворк, позволяющий разрабатывать web приложения любой сложности. Он спроектирован таким образом, чтобы максимально разделить ответственность между различными частями приложения. Как видно из названия, фреймворк использует шаблон проектирования MVC (Model – View – Controller). Такой подход позволяет четко разделить ответственность между View – представлением, Model – моделью и Controller – контроллером. Представление (View) содержит Html разметку и управляющие конструкции, которые отвечают за формирование представления. Модель (Model) содержит описание объектов, это могут быть ORM или обычные CLR типы. Контроллер (Controller) подготавливает модель и передает ее представлению. В некоторых случаях разработчики пишут бизнес-логику приложения прямо в методах контроллера. Такой подход возможен для небольших проектов или демонстраций. В реальных же проектах бизнес-логику лучше инкапсулировать в отдельные классы и библиотеки.

AJAX – расшифровывается как асинхронный JavaScript и XML. Эта технология позволяет в фоновом режиме, то есть без полного обновления страницы, обмениваться данными между сервером и клиентом.

jQuery – это JavaScript фреймворк, который позволяет легко получать доступ к любому элементу DOM и манипулировать ими.

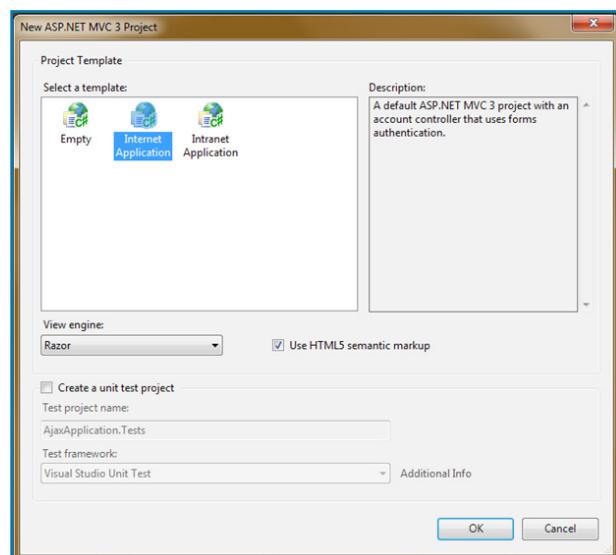
## Создание проекта ASP.NET MVC

Создадим новый проект ASP.NET MVC. Для этого откроем

Visual Studio 2010, выберем пункт в верхнем меню **File → New → Project**. В открывшемся диалоговом окне выберем проект **ASP.NET MVC 3 WebApplication** и назовем его **AjaxApplication**. После нажатия кнопки **OK**, появится новое диалоговое окно, в котором будет предложено выбрать один из трех шаблонов проекта. Первый шаблон **Empty**, как и следует из названия, является «пустым». В нем отсутствуют генерированные по умолчанию контроллеры и представления, а так же отсутствует аутентификация. Он содержит лишь основной каркас приложения MVC и скрипты. Второй и третий шаблоны **Internet Application** представляют собой базовые web-приложения с аутентификацией и приятной темой оформления. Отличие заключается лишь в типе аутентификации. В первом используется **users forms** аутентификация, во втором **windows forms** аутентификация.

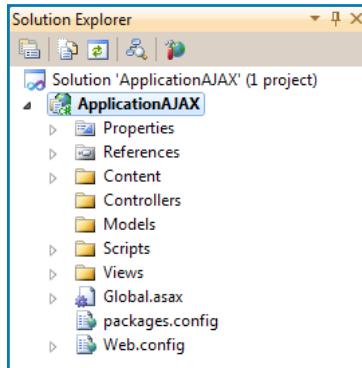
Выберем второй шаблон **InternetApplication**. Отметим галочкой пункт **Use HTML5 semantic markup**. Он включает использование семантики HTML 5 в авто-генерированных представлениях, а так же подключает библиотеку **modernize.js**, которая позволяет сайту нормально функционировать в браузерах, не поддерживающих HTML5, например в браузере IE6.

Снимем галочку с пункта **Create Unit Test Project** (Создание тестового проекта).



## Структура проекта ASP.NET MVC

После создания нового проекта мы получим следующую структуру.



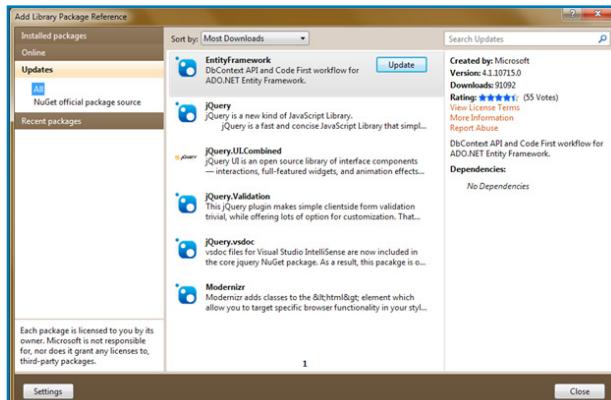
Описание папок проекта:

- **Content** – содержит изображения, иконки, css стили и другие объекты, использующиеся в проекте;
- **Controllers** – содержит специальные классы-контроллеры. Каждый контроллер унаследован от класса `System.Web.Mvc.Controllers` и имеет в названии окончание `Controller`;
- **Models** – содержит модель;
- **Scripts** – содержит различные библиотеки jQuery, MicrosoftAJAX, Modernize;
- **Views** – содержит представления, то есть то, что видит пользователь в своем браузере. ASP.NET MVC позволяет использовать различные движки представлений. По умолчанию доступны два: asp и razor. В нашем проекте мы будем использовать razor;
- **Global.asax** – содержит обработчики событий уровня приложения и уровня сеанса;
- **Web.config** – конфигурационный файл проекта;
- **Packages.config** – содержит информацию об установленных пакетах.

Удалим из проекта все файлы из папки **Controllers** и папки **Home** и **About** в папке **Views**.

## Обновление пакетов

Перед тем, как мы начнем работу над проектом, неплохо было бы проверить обновления установленных пакетов. Для этого щелкнем правой кнопкой мыши по папке **Reference** и выберем пункт **Add Library Packages Reference**. Появится



новое диалоговое окно. Перейдем в раздел **Updates**. Начнется поиск обновлений для установленных пакетов. Если для ваших пакетов были найдены обновления, то вы увидите список доступных обновлений. Выберите пакет, который хотите обновить и нажмите кнопку **Update**.

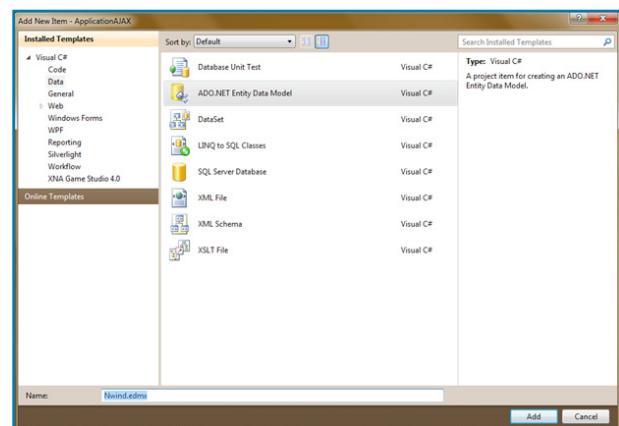
После обновления библиотеки jQuery, необходимо изменить все ссылки, указывающие на предыдущие версии файлов. Откроем файл `_Layout.cshtml`, который находится в папке **Views → Shared**, и изменим названия java script файлов в соответствии с текущей версией. Например, так:

```
<script src="@Url.Content("~/Scripts/jquery-1.6.4.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/modernizr-2.0.6-development-only.js")" type="text/javascript"></script>
```

## Создание модели данных

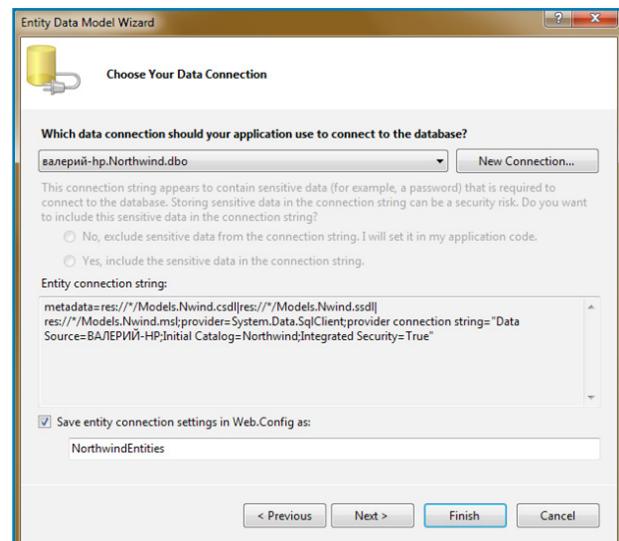
В проекте мы будем использовать тестовую базу данных Northwind, которую можно загрузить [отсюда](#).

После того как база данных создана в SQL Server, добавим в наш проект модель. Для этого нажмем правой кнопкой мыши на папку **Models** и выберем пункт **Add → New Item**. Затем выберем элемент **ADO.NET Entity Data Model** и назовем его `Nwind.edmx`.



Нажмем кнопку **Add**. Появится следующее диалоговое окно, в котором мастер создания модели данных предложит выбрать модель содержимого, то есть, из какого источника будут получены данные. Выберем **Generate from database** и нажмем **Next**.

Следующим шагом нужно создать строку подключения к базе ►



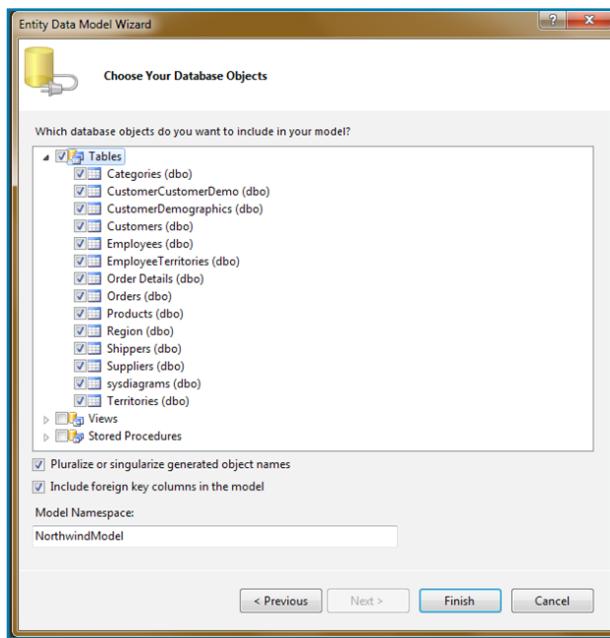
данных. Если ранее уже было создано соединение с базой данных, то можно выбрать существующую строку подключения из выпадающего списка. В противном случае необходимо создать новое соединение. Для этого щелкнем по кнопке **New Connection**. Откроется диалоговое окно, в котором заполним поле **Server Name** – в этом поле укажем имя вашего сервера базы данных или выберем из списка найденных серверов баз данных на вашем компьютере.

Выберем способ аутентификации **Log on to the server**. Для нашего приложения подойдет Windows аутентификация – **Use Windows Authentication**. Выберем базу данных, к которой будет выполнено подключение. В нашем случае нужно выбрать базу данных Northwind. Нажмем кнопку **OK**. Если все было сделано правильно, то работа окна завершится и будет сформирована строка подключения к базе данных.

Отметим галочкой **Save entity connection string in Web.Config as** и зададим название **NorthwindEntities**.

Нажмем **Next**. На следующем шаге нам предложено выбрать элементы, для которых будет сформирована модель. Поставим галочку на элементе **Tables**, и все сущности базы данных будут автоматически выбраны. Зададим имя пространства имен Model Namespace – **NorthwindModel**.

Отметим галочкой пункт **Pluralize or singularize generated object names** и нажмем **Finish**.



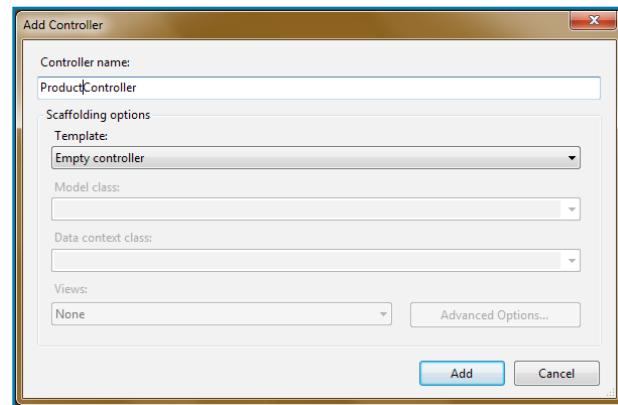
После завершения работы мастера нам будет доступна объектно-реляционная модель, которая содержит все сущности базы данных Northwind. Скомпилируем проект.

## Создание контроллера

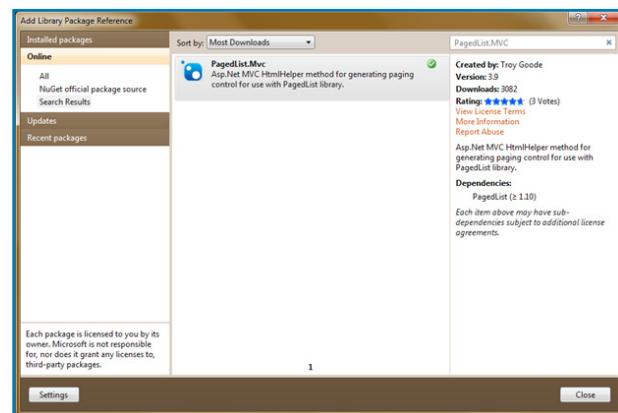
Давайте добавим в наш проект контроллер, который будет отвечать за работу с продуктами. Для этого нажмем правой кнопкой мыши на папку **Controllers** и выберем пункт **Add → Controller**. Задайте имя контроллера **ProductController** и нажмите **Add**.

В папке **Controllers** появится новый класс с именем **ProductController**. Перед тем, как мы приступим к написанию кода, добавим пакет **PagedList**, который поможет организовать постраничную навигацию для списка продуктов.

Нажмем правой кнопкой мыши на папку **Reference** и выберем пункт **Add Library Packages Reference**. Появится диалоговое окно добавления пакетов. Перейдем в раздел **Online** и в поле поиска введем **PagedList.MVC**. Добавим найденный пакет в



проект, нажав **Install**. В папку **Reference** будет добавлено несколько библиотек **PagedList** и **PagedList.MVC**.



Откроем файл **ProductController.cs** в папке **Controllers** и отредактируем его следующим образом.

```
using System.Linq;
using System.Web.Mvc;
using AjaxApplication.Models;
using PagedList;

namespace AjaxApplication.Controllers
{
    public class ProductController : Controller
    {
        // Контекст данных
        private NorthwindEntities _db;

        // Конструктор контроллера
        public ProductController()
        {
            // Инициализация контекста данных
            _db = new NorthwindEntities();
        }

        // Действие возвращает список продуктов
        public ActionResult Products(int? page)
        {
            // Если не выбрана страница...
            if (page == null)
            {
                page = 1;
            }

            // Выбираем все продукты
            IQueryables<Product> products = _db.Products.OrderBy(p => p.ProductName);
            // Выполняем запрос с разбиением на страницы. Каждая страница включает в себя 10 продуктов
            IPagedList<Product> pageList = products.ToPagedList((int)page, 10); ▶
        }
    }
}
```

```

        return View(pageList);
    }

}

```

ProductController содержит действие Products, которое возвращает список продуктов.

## Создание представления

Теперь добавим представление для действия **Products**. Нажмем правой кнопкой мыши на название действия **Products** в контроллере **ProductController**. В выпадающем меню выберем пункт **Add View**. В открывшемся диалоговом окне введем имя нашего представления и нажмем **Add**.

В окне добавления нового представления мы можем выбрать пункт **Create a strongly typed view** (Создать строго-типовизированное представление) после чего нам будет доступен список доступных типов. Если мы выберем модель представления, например **Product**, то станет доступным пункт **Scaffold Template** (Шаблон построения страницы), который позволяет выбрать один из следующих шаблонов:

- ▀ **Empty** – пустой шаблон,
- ▀ **Create** – шаблон создания (например, создание нового продукта),
- ▀ **Edit** – редактирование,
- ▀ **Details** – детальная информация,
- ▀ **List** – таблица,
- ▀ **Delete** – удаление.

Будет создано новое представление. Отредактируем файл Products.cshtml следующим образом:

```

@*Модель представления*@
@model IEnumerable<AjaxApplication.Models.Product>

@*Пространства имён, содержащие типы и методы для
разбиения на страницы*@
@using PagedList;
@using PagedList.Mvc;

 @{
    //Заголовок
    ViewBag.Title = "Продукты";
    //Ссылка на мастер-страницу
    Layout = "~/Views/Shared/_Layout.cshtml";
}



## Список продуктов



@*HTML помощник, формирующий ссылку на действие
    Create контроллера Product*
    @Html.ActionLink("Добавить новый продукт",
    "Create")



@*Создаем таблицу и определяем столбцы*
    <table>
        <tr>
            <th> Название продукта
            </th>
            <th> Поставщик
            </th>
            <th> Категория
            </th>
            <th> Количество
            </th>
            <th> Цена
            </th>
            <th> Количество
            </th>
            <th> Заказано
            </th>
            <th> Скидка
            </th>
        </tr>


```

Название продукта	Поставщик	Категория	Количество	Цена	Количество	Заказано	Скидка

@\*Цикл, создающий тело таблицы.\*@
 @foreach (var item in Model) {
 //Создаем новую строку и заполняем ячейки
 //таблицы из модели
 <tr>
 <td> @item.ProductName </td>
 <td> @item.Supplier.CompanyName </td>
 <td> @item.Category.CategoryName </td>
 <td> @item.QuantityPerUnit </td>
 <td>
 @if (item.UnitPrice != null)
 {
 @item.UnitPrice.Value.ToString("C")
 }
 </td>
 <td> @item.UnitsInStock </td>
 <td> @item.UnitsOnOrder </td>
 <td> @item.Discontinued </td>
 <td>
 @\*HTML помощники для создания ссылок на
 действия Edit и Delete, контроллера Product\*
 @Html.ActionLink("Edit", "Edit", new { id=item.ProductID }) |
 @Html.ActionLink("Delete", "Delete", new { id=item.ProductID })
 </td>
 </tr>
 }
 </table>
 @\*Метод помогающий сформировать ссылки для навигации
 по страницам\*
 @\*В качестве параметров метод получает список
 продуктов и ссылку на действие Products.\*@
 @Html.PagedListPager((IPagedList)Model, page=> Url.
 Action("Products", new {page}))

Для корректного отображения ссылок на страницы добавим ссылку на стили PagedList.css. Этот файл находится в папке Content.

```

<link href="../../Content/PagedList.css" rel="stylesheet" type="text/css" />

```

## Подготовка мастер страницы

Перед тем, как запустить приложение, убедимся, что наша мастер-страница содержит правильные ссылки на библиотеки и действия. За исключением версии jQuery, страница должна содержать приблизительно такую разметку:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>@ViewBag.Title</title>
    <link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
    <script src="@Url.Content("~/Scripts/jquery-1.6.4.min.js")" type="text/javascript"></script>
    <script src="@Url.Content("~/Scripts/modernizr-2.0.6-development-only.js")" type="text/javascript"></script>
</head>
<body>
    <div class="page">
        <header>
            <div id="title">
                <h1>ASP.NET MVC + jQuery</h1>
            </div>
            <div id="logindisplay">
                @Html.Partial("_LogOnPartial")
            </div>
            <nav>
                <ul id="menu">
                    <li>@Html.ActionLink("Продукты", "Products", "Product")</li>
                </ul>
            </nav>
        </header>
        <section id="main">
            @RenderBody()
        </section>
        <footer>
        </footer>
    </div>
</body>
</html>
```

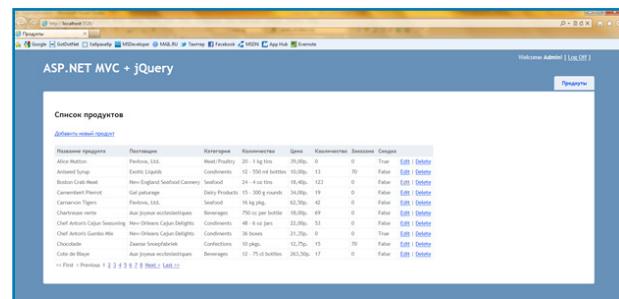
## Настройка маршрутизации

Поскольку мы удалили контроллер **Home** и все его представления, необходимо изменить параметры маршрутизации. Для этого отредактируем метод RegisterRoutes в файле Global.asax

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        "Default", // Route name
        "{controller}/{action}/{id}", // URL with parameters
        new { controller = "Product", action = "Products", id = UrlParameter.Optional } // Parameter defaults
    );
}
```

Запустим приложение. Если все сделано правильно, то мы увидим список продуктов.



## jQuery AJAX (получение информации о поставщике)

Самое время добавить в наш проект Ajax функциональность. Пусть при выборе строки выводится детальная информация о поставщике. Для того чтобы реализовать подобный сценарий, нам потребуется создать новое действие в контроллере Products, которое будет возвращать данные поставщика по номеру продукта. Затем мы напишем сценарий с использованием jQuery, который вызовет действие с использованием Ajax и вернет данные в формате json.

Добавим в контроллер ProductController действие GetSupplierInfo.

```
public JsonResult GetSupplierInfo(string id)
{
    // Преобразуем строку в число
    int productId = int.Parse(id);

    // LINQ запрос. Послучаем данные о поставщике для найденного продукта.
    var result = from p in _db.Products
                where p.ProductID == productId
                select new {
                    SupplierName = p.Supplier.CompanyName,
                    SupplierContact = p.Supplier.ContactName,
                    SupplierCountry = p.Supplier.Country,
                    SupplierFax = p.Supplier.Fax,
                    SupplierPhone = p.Supplier.Phone };

    // Создаем класс, который будем использовать для передачи результата
    JsonResult jsonResult = new JsonResult();

    // Записываем в свойство Data полученные данные в виде массива.
    jsonResult.Data = result.ToArray();
    // Разрешаем ответ на Get запрос
    jsonResult.JsonRequestBehavior = JsonRequestBehavior.AllowGet;

    return jsonResult;
}
```

Действие GetSupplierInfo имеет тип возвращаемого значения JsonResult, так как мы будем возвращать данные в формате json. После того, как действие будет вызвано и ему будет передан идентификатор продукта, выполняется Linq запрос. В запросе используется синтаксис Linq to SQL, который похож на стандартный SQL синтаксис. В этом запросе мы выбираем продукт по идентификатору, после чего оператором select new создается новый анонимный тип.

Стоит отметить, что на данном шаге запрос к базе данных не выполнен. Это происходит потому, что по умолчанию Linq запросы выполняются по требованию, то есть только в тот момент, когда данные будут явно запрошены.

Дальше создаем экземпляр типа JsonResult и записываем в его свойство Data результат выполнения запроса, прео->

бразованный в массив. Именно в этот момент выполняется SQL запрос к базе данных.

Перейдем к представлению Products. Сделаем так, чтобы при щелчке мышью по какой либо строке, эта строка выделялась цветом и запускалось действие, возвращающее информацию о поставщике.

Добавим в файл стилей Site.css новый класс .selected-row, который будет устанавливать свойство background в оранжевый цвет.

```
.selected-row {
    background: orange;
}
```

Под заголовком таблицы **<h2>Список продуктов</h2>** разместим область, в которой будем отображать полученные данные о поставщике.

```
<div id="InfoPanel">
    <b>Поставщик: </b><span id="CompanyName"></span><br/>
    <b>Контактное лицо: </b><span id="ContactName"></span><br/>
    <b>Страна: </b><span id="Country"></span><br/>
    <b>Телефон: </b><span id="Phone"></span><br/>
    <b>Факс: </b><span id="Fax"></span><br/>
</div>
```

Элемент div с идентификатором InfoPanel содержит строки, в которые будут помещены данные о поставщике. Полученные данные будут вставлены в тело элементов span, каждый из которых имеет свой идентификатор, указывающий, какие именно данные он будет содержать.

Наконец, напишем сценарий на jQuery. Добавите в представление Products следующий JavaScript код.

```
<script type="text/javascript">
$(document).ready(function () {
    $("#InfoPanel").hide();
    $("tr").click(function () {
        $("tr").each(function () {
            $(this).removeClass("selected-row");
        });
        $(this).addClass("selected-row");

        var id = $(this).find("td:first").text();

        $.getJSON('@Url.Action("GetSupplierInfo")', { id: id }, callback);

        function callback (result) {
            $("#InfoPanel").show("slow");
            $("#CompanyName").html(result[0].SupplierName);
            $("#ContactName").html(result[0].SupplierContact);
            $("#Country").html(result[0].SupplierCountry);
            $("#Phone").html(result[0].SupplierPhone);
            $("#Fax").html(result[0].SupplierFax);
        }
    });
});
</script>
```

Разберем работу сценария по порядку. Основной операцией в jQuery является выборка части документа. Для этого служит конструкция `$( )`. Как правило, она принимает строковый параметр, который может содержать любое выражение селектора CSS.



Чтобы наш код выполнился после того, как все элементы дерева DOM будут загружены без ожидания загрузки изображений, применяется такая конструкция:

```
<script type="text/javascript">
$(document).ready(function () {
    });
</script>
```

Где фабричная функция `$( )` принимает в качестве параметра объект document (текущая HTML страница) и подписывается на событие ready() – специальное событие jQuery наступающее после загрузки дерева DOM. В методе ready() создается функция без имени. Именно в эту функцию мы поместим весь код.

В первом операторе `$("#InfoPanel").hide();` мы скрываем область, где должна выводиться информация о поставщиках. То же самое можно было сделать без использования jQuery, например, с помощью стилей CSS.

Затем мы подписываемся на событие, возникающее при щелчке мышью по строке. Как и в случае с методом ready(), в методе click() создается функция. В тело этой функции мы добавим код, который должен выполняться при щелчке мышью по строке.

```
$(tr).click(function () {
});
```

Так как мы хотим, чтобы при нажатии мышью на строке эта строка получала фокус, выделяясь оранжевым цветом, а предыдущая строка теряла, то есть принимала цвет, заданный по умолчанию, необходимо выполнить сброс фокуса для предыдущей выбранной строки и установить новый цвет для текущей.

В качестве фокуса мы будем использовать заранее подготовленный класс, который устанавливает свойство background строки в оранжевый цвет. Для того чтобы снять фокус со строки мы будем удалять этот класс.

Итак, первым делом мы «очищаем» все строки таблицы, удаляя класс selected-row. Для этого используется метод `each()`. Этот метод является циклом, перечисляющим все строки в таблице и для каждой найденной строки выполняющим операцию `$(this).removeClass("selected-row")`. Где `$(this)` указывает на текущий элемент, в данном случае на строку, а метод `removeClass()` удаляет класс по имени.

```
$(tr).each(function () {
    $(this).removeClass("selected-row"); ▶
```

```
});  
$(this).addClass("selected-row");
```

Методом `addClass()` мы добавляем класс, выделяющий текущую строку оранжевым цветом.

Для того чтобы выполнить действие возвращающее информацию о поставщике, ему необходимо передать в качестве параметра идентификатор продукта. Извлечь номер продукта из строки очень просто. Для этого мы воспользуемся методом `find()` которому в качестве параметра передадим селектор, отбирающий первую ячейку в строке. Так как именно в первой ячейке строки у нас находится номер продукта, этот метод вернет его.

```
var id = $(this).find("td:first").text();
```

Найденный номер продукта сохраняется в переменной с именем `id`.

Теперь все готово, чтобы выполнить Ajax запрос.

```
$.getJSON('@Url.Action("GetSupplierInfo")', { id: id },  
callback);
```

Данная функция имеет следующий вид:

```
$.getJSON({  
    url: url,  
    data: data,  
    success: callback  
});
```

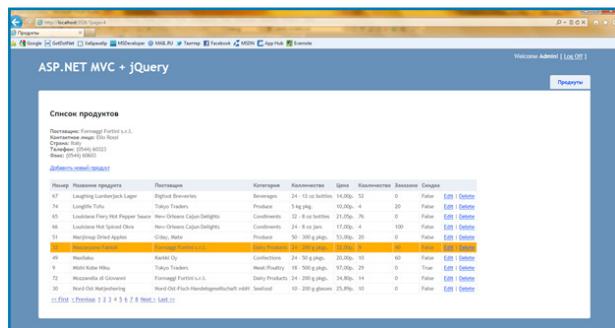
Где:

- `url` – это строка содержащая адрес, по которому отправляется запрос,
- `data` – данные, которые передаются серверу с запросом,
- `success` – функция обратного вызова, которая выполняется в случае, если запрос выполнен успешно.

С помощью вспомогательного метода `Url.Action()` мы формируем ссылку на действие `GetSupplierInfo`. Затем передаем переменную `id`, в которой содержится идентификатор продукта, и функцию обратного вызова. Функция обратного вызова имеет следующий вид:

```
function callback (result) {  
    $('#InfoPanel').show('slow');  
    $('#CompanyName').html(result[0].SupplierName);  
    $('#ContactName').html(result[0].SupplierContact);  
    $('#Country').html(result[0].SupplierCountry);  
    $('#Phone').html(result[0].SupplierPhone);  
    $('#Fax').html(result[0].SupplierFax);  
}
```

Функция `callback` в качестве параметра принимает результат, полученный от сервера. Затем выбирает нужные html элементы, в которые помещаются результаты. Результат будет иметь следующий вид.



Теперь при выборе продукта, в таблице, мы получаем дан-

ные о поставщике без обновления всей страницы.

## jQuery AJAX (создание нового продукта)

Допустим, что мы хотим создать новый продукт без полного обновления страницы при помощи Ajax. Такое поведение не трудно реализовать. Для этого мы воспользуемся возможностями, предоставляемыми jQuery по аналогии с предыдущим примером. Как вы помните, для получения информации о поставщике мы использовали функцию `getJSON`, которой передавали Url действия, номер продукта и функцию обратного вызова. Для создания нового продукта мы воспользуемся функцией `$.post`, которой передадим Url действия и данные с формы.

Создадим в контроллере `Product` новое действие. Назовем его `AddProduct`. Так как действие будет отвечать за создание продукта, пометим его атрибутом `HttpPost`, чтобы оно смогло отвечать только на запросы типа POST. Отредактируем метод `AddProduct` следующим образом.

```
[HttpPost]  
public void AddProduct(Product product)  
{  
    // Добавляем новый продукт  
    _db.AddToProducts(product);  
    // Отправляем изменение в базу данных  
    _db.SaveChanges();  
}
```

Метод `AddProduct` не возвращает никакой информации, поэтому имеет тип `void`. В качестве параметра он получает данные с формы. Модель связывания данных в ASP.NET MVC достаточно умна, чтобы самостоятельно заполнить объект типа `Product`, путем сопоставления имен полей. В большинстве случаев, стандартных средств связывания данных хватает, но если необходимо, мы можем создавать и свои привязки данных.



**Привязка модели** – это механизм ASP.NET для отображения данных HTTP запроса непосредственно на параметры метода действия и специальные объекты .NET (включая коллекции).

Создание нового продукта мы сделаем через модальное окно, которое будет открываться при нажатии на кнопку **Добавить продукт**. Добавим в Html код представления `Products` форму.

```
<div id="ProductWindow">  
<form id="ProductForm" method="POST">  
    <fieldset>  
        <input type="hidden" name="ProductID" id="ProductID"/>  
        <legend>Продукт</legend>  
        <div class="editor-label">  
            <label for="ProductName">Название продукта</label>  
        </div>  
        <div class="editor-field">  
            <input type="text" name="ProductName" id="ProductName" class="required" />  
        </div>  
        <div class="editor-label">  
            <label for="SupplierID">Номер поставщика</label>  
        </div>  
        <div class="editor-field">  
            <input type="text" name="SupplierID" id="SupplierID" />  
        </div>  
        <div class="editor-label">  
            <label for="CategoryID">Номер категории</label>  
        </div>
```

```

</div>
<div class="editor-field">
    <input type="text" name="CategoryID" id="CategoryID"/>
</div>

<div class="editor-label">
    <label for="QuantityPerUnit">Количество</label>
</div>
<div class="editor-field">
    <input type="text" name="QuantityPerUnit" id="QuantityPerUnit" minlength="2" />
</div>

<div class="editor-label">
    <label for="UnitPrice">Цена</label>
</div>
<div class="editor-field">
    <input type="text" name="UnitPrice" id="UnitPrice"/>
</div>

<div class="editor-label">
    <label for="UnitsInStock">Количество</label>
</div>
<div class="editor-field">
    <input type="text" name="UnitsInStock" id="UnitsInStock"/>
</div>

<div class="editor-label">
    <label for="UnitsOnOrder">Заказано</label>
</div>
<div class="editor-field">
    <input type="text" name="UnitsOnOrder" id="UnitsOnOrder"/>
</div>

<div class="editor-label">
    <label for="ReorderLevel">Уровень</label>
</div>
<div class="editor-field">
    <input type="text" name="ReorderLevel" id="ReorderLevel"/>
</div>

<div class="editor-label">
    <label for="Discontinued">Скидка</label>
</div>
<div class="editor-field">
    <input type="checkbox" name="Discontinued" id="Discontinued" value="" />
</div>
</fieldset>
</form>
</div>

```

Форма обернута в Html элемент div, использующий в качестве контейнера. Это и есть будущее модальное окно, которое мы создадим чуть позже, используя jQuery UI.

Давайте добавим кнопку, при нажатии на которую будет отображаться диалоговое окно. Так как у нас уже есть ссылка на несуществующее действие Create, заменим его на Html Button, задав в качестве идентификатора имя OpenWindow.

```
<button id="OpenWindow">Добавить новый продукт</button>
```

Добавим сценарий создания модального окна.

```
$("#ProductWindow").dialog({
    autoOpen: false,
    height: 500,
    width: 400,
```

```

    modal: true,
    buttons: {
        «Сохранить»: function () {
            if ($("#ProductForm").valid()) {
                var data = $("#ProductForm").serialize();
                $.post(action, data,
                    function (e) {
                        $("#ProductWindow").dialog("close");
                    });
            }
        },
        «Отмена»: function () {
            $(this).dialog("close");
        }
    });
}
```

Для создания окна используем элемент управления jQuery UI Window. Используя селектор jQuery, мы выбираем элемент с идентификатором ProductWindow. Затем вызываем метод создания окна .dialog(). В конструкторе окна заполняем следующие свойства:

- ▼ **AutoOpen** – разрешить или запретить автоматическое открытие окна;
- ▼ **Height** – высота;
- ▼ **Width** – ширина;
- ▼ **Modal** – сделать окно модальным;
- ▼ **Buttons** – кнопки.

Окно имеет две кнопки, кнопку Сохранить и Отмена. Кнопка Сохранить содержит функцию, отвечающую за асинхронную отправку данных с формы на сервер.

Сначала мы проверяем форму на валидность.

```
if ($("#ProductForm").valid() {...})
```

Если форма заполнена правильно, то выполняется сериализация в переменную data.

```
var data = $("#ProductForm").serialize();
```

Сериализация помогает сформировать строку в виде пар ключ\значение для отправки на сервер.

И наконец, выполняется Ajax запрос.

```

$.post(action, data,
    function (e) {
        $("#ProductWindow").dialog("close");
    });
}
```

Так как данный сценарий мы будем использовать не только для создания новой записи, но и для редактирования существующей, необходимо предусмотреть различные варианты Url адреса на методы действия. Например, для создания нового продукта мы использовали действие AddProduct, а для редактирования существующего продукта будем использовать EditProduct.

Для того чтобы реализовать подобный сценарий, добавим глобальную переменную action, в которую будем помещать адрес на действие.

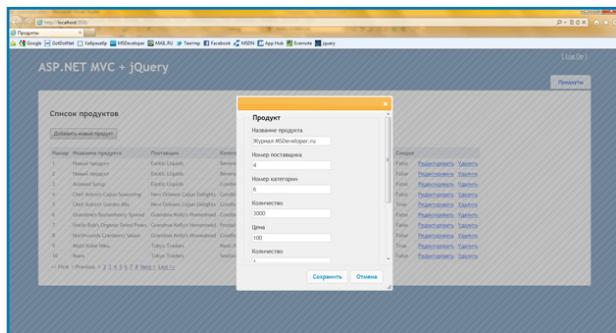
```
var action = null;
```

Добавим обработчик события для кнопки создания нового продукта.

```

$("#OpenWindow").click(function () {
    action = '@Url.Action("AddProduct", "Product")';
    $("#ProductWindow").dialog("open");
    return false;
}); ▶
```

В этом обработчике мы формируем ссылку на метод действия AddProduct с помощью Url Helper и помещаем ее в переменную action. Затем открываем окно. Теперь при нажатии на кнопку **Добавить новый продукт**, откроется модальное окно, которое в качестве метода действия будет использовать /Product/AddProduct.

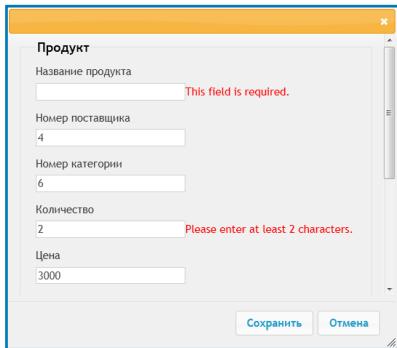


Обратите внимание, так как мы использовали валидацию, форма не будет отправлена на сервер, пока не отвечает правилам проверки. Использовать правила проверки очень просто. Посмотрите на html разметку формы. У поля **Название продукта** имеется атрибут class с установленным значением required (обязательное поле).

```
<input type="text" name="ProductName" id="ProductName" class="required" />
```

А у поля **Количество** добавлен атрибут minlength, устанавливающий минимальную длину поля.

```
<input type="text" name="QuantityPerUnit" id="QuantityPerUnit" minlength="2" />
```



Не забывайте, что валидация пользовательского ввода должна быть не только на клиенте, но и на сервере. Поэтому просто необходимо проверить данные на сервере перед их использованием. В данном примере этот шаг опущен, для упрощения кода.

Чтобы получить более подробную информацию о валидации посетите [сайт](#) библиотеки jQuery. К тому же там вы найдете большое количество плагинов для реализации проверки пользовательского ввода.

Итак, теперь у нас почти все готово для создания нового продукта. Но если вы попробуете создать продукт сейчас, то заметите, что даже при установленном флагке у поля **Discounted** (скидка) в базу всегда попадает значение false, так как не передается обязательное поле **Discounted** (скидка). Это происходит потому, что после сериализации формы, параметр Discounted будет иметь значение On, то есть на сервер придет строка Discounted=On. Стандартная модель привязки ASP.NET не сможет сконвертировать это значение в true, поэтому просто проигнорирует его.

Чтобы решить эту проблему, добавим небольшой сценарий,

который будет менять значение флагка при щелчке мышью.

```
$(«#Discontinued»).click(function () {
    if ($(«#Discontinued»).is(«:checked»))
        $(«#Discontinued»).val(«true»);
    }
    else {
        $(«#Discontinued»).val(«false»);
    }
});
```

Мы подписали флагок на событие щелчка мышью, в котором осуществляется проверка. Если флагок установлен, то значение устанавливается в true.

## jQuery AJAX (редактирование продукта)

Добавим в ProductController новое действие, отвечающее за редактирование продукта.

```
public JsonResult EditProduct(Product product)
{
    object jproduct = new object();

    // Получаем продукт из базы данных, который необходимо изменить
    Product updateProduct = _db.Products.Where(p => p.ProductID == product.ProductID).FirstOrDefault();
    // Если такой продукт найден, изменяем его
    if (updateProduct != null)
    {
        updateProduct.ProductName = product.ProductName;
        updateProduct.SupplierID = product.SupplierID;
        updateProduct.CategoryID = product.CategoryID;
        updateProduct.QuantityPerUnit = product.QuantityPerUnit;
        updateProduct.UnitPrice = product.UnitPrice;
        updateProduct.UnitsInStock = product.UnitsInStock;
        updateProduct.UnitsOnOrder = product.UnitsOnOrder;
        updateProduct.ReorderLevel = product.ReorderLevel;
        updateProduct.Discontinued = product.Discontinued;

        _db.SaveChanges();

        // Создаем новый объект, который будет возвращен
        jproduct = new
        {
            ProductName = updateProduct.ProductName,
            Supplier = updateProduct.Supplier.CompanyName,
            Category = updateProduct.Category.CategoryName,
            QuantityPerUnit = updateProduct.QuantityPerUnit,
            UnitPrice = updateProduct.UnitPrice,
            UnitsInStock = updateProduct.UnitsInStock,
            UnitsOnOrder = updateProduct.UnitsOnOrder,
            ReorderLevel = updateProduct.ReorderLevel,
            Discontinued = updateProduct.Discontinued
        };
    }

    return Json(jproduct);
}
```

Действие EditProduct имеет тип JsonResult, который подразумевает возвращение результата в формате json. Для того чтобы вернуть данные в формате json, выполняется метод Json(jproduct), который в качестве входного параметра получает объект.

Так как нам нужно, чтобы вместо номера категории и номера поставщика отображались их имена, создается новый объект, где вместо SupplierID и CategoryID передаются CompanyName и CategoryName. ►

Для редактирования выбранной записи мы будем использовать то же окно, что и для создания. Для этого нам будет нужно передать ссылку на другой метод действия, а так же модифицировать функцию обратного вызова для внесения изменений в таблицу.

Подпишемся на событие click() ячейки таблицы.

```
$(<td>).click(function () {
    var t = $.trim($(this).text());
    if (t == «Редактировать») {

        $('#ProductID').val($(this).parent().find(<td>).
            eq(0).text().trim());
        $('#ProductName').val($(this).parent().find(<td>).
            eq(1).text().trim());
        $('#SupplierID').val($(this).parent().find(<td>).
            eq(2).attr('Tag').trim());
        $('#CategoryID').val($(this).parent().find(<td>).
            eq(3).attr('Tag').trim());
        $('#QuantityPerUnit').val($(this).parent().
            find(<td>).eq(4).text().trim());
        $('#UnitPrice').val($(this).parent().find(<td>).
            eq(5).text().trim());
        $('#UnitsInStock').val($(this).parent().
            find(<td>).eq(6).text().trim());
        $('#UnitsOnOrder').val($(this).parent().
            find(<td>).eq(7).text().trim());

        var b = $(this).parent().find(<td>).eq(8).text().
            trim().toLowerCase() == 'true';
        $('#Discontinued').attr('checked', b);

        action = '@Url.Action("EditProduct", "Product")';
        $('#ProductWindow').dialog('open');
    }
});
```

Сначала мы получаем текст ячейки, на которую пользователь щелкнул. Если текст содержит значение Редактировать, то поочередно выбираются все значения ячеек из строки.

```
$('#ProductID').val($(this).parent().find(<td>).
    eq(0).text().trim());
```

Сначала мы переходим на родительский элемент ячейки, то есть на строку, затем в строке ищутся все ячейки и выбирается самая первая. После чего извлекается ее текст, и обрезаются пробелы.

```
$('#ProductID').val(...)
```

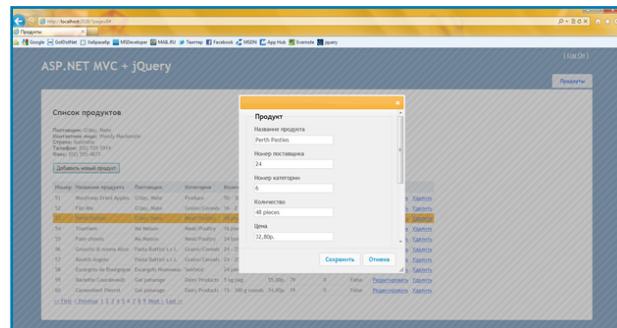
Полученное значение присваивается соответствующему полю.

После заполнения всех полей формы, присваиваем значение переменной action и открываем окно.

Для того чтобы, после редактирования записи в таблице произошли изменения, необходимо модифицировать функцию обратного вызова метода .post() в конструкторе окна следующим образом.

```
if (action == '@Url.Action("EditProduct", "Product")'
{
    var cell = $('#table1').
        find('tr[class=selected-row]').find(<td>);
    cell.eq(1).text(e.ProductName);
    cell.eq(2).text(e.Supplier);
    cell.eq(3).text(e.Category);
    cell.eq(4).text(e.QuantityPerUnit);
    cell.eq(5).text(e.UnitPrice);
    cell.eq(6).text(e.UnitsInStock);
    cell.eq(8).text(e.Discontinued);
});
```

Сперва проверяем условие, если запись редактировалась, то находим все ячейки в текущей строке. Затем устанавливаем новые значения для каждой ячейки. В результате, после редактирования продукта данные в таблице изменятся без перезагрузки страницы.



## jQuery AJAX (удаление продукта)

Добавим в ProductController новое действие DeleteProduct.

```
[HttpPost]
public void DeleteProduct(int productId)
{
    Product product = _db.Products.Where(p => p.ProductID == productId).FirstOrDefault();
    if (product != null)
    {
        _db.Products.DeleteObject(product);
        _db.SaveChanges();
    }
}
```

На вход действие получает id продукта, затем ищет такой продукт и если продукт был найден, удаляет его.

Добавим сценарий в представление Products. Для этого отредактируем событие щелчка мыши по ячейке, добавив новое условие после условия редактирования записи.

```
if (t == «Удалить») {
    action = '@Url.Action("DeleteProduct", "Product")';
    $.post(action, { productId: $(this).parent().
        find(<td>).eq(0).text().trim() },
        function ()
        {
            $('#table1').find('tr[class=selected-row]').
                remove();
            alert('Продукт удален');
        });
}
```

Здесь аналогично редактированию выполняется Ajax запрос с помощью метода .post(). В качестве параметра метод получает id продукта, ссылку на метод действия и функцию обратного вызова. Последняя, в свою очередь, удаляет текущую строку без полной перезагрузки страницы и выводит сообщение.

## jQuery AJAX (весь сценарий)

Ниже представлен весь сценарий представления Products.

```
<script type="text/javascript">
$(document).ready(function () {

    var action = null;
    $('#InfoPanel').hide();

    $('#ProductWindow').dialog({ ▶
```

```

autoOpen: false,
height: 500,
width: 400,
modal: true,
buttons: {
    "Сохранить": function () {
        if ($("#ProductForm").valid()) {
            var data = $("#ProductForm").serialize();
            $.post(action, data,
                function (e) {
                    $("#ProductWindow").dialog("close");
                }
            );
            if (action == '@Url.Action("EditProduct", "Product")')
            {
                var cell = $("#table1").find("tr [class=selected-row]").find("td");
                cell.eq(1).text(e.ProductName);
                cell.eq(2).text(e.Supplier);
                cell.eq(3).text(e.Category);
                cell.eq(4).text(e.QuantityPerUnit);
                cell.eq(5).text(e.UnitPrice);
                cell.eq(6).text(e.UnitsInStock);
                cell.eq(7).text(e.UnitsOnOrder);
                cell.eq(8).text(e.Discontinued);
            }
        });
    },
    "Отмена": function () {
        $(this).dialog("close");
    }
});
});

$("#OpenWindow").click(function () {
    action = '@Url.Action("AddProduct", "Product")';
    $("#ProductWindow").dialog("open");
    return false;
});

$("#Discontinued").click(function () {
    if ($("#Discontinued").is(":checked"))
        $("#Discontinued").val("true");
    else
        $("#Discontinued").val("false");
});

$("#tr").click(function () {
    $("#tr").each(function () {
        $(this).removeClass("selected-row");
    });
    $(this).addClass("selected-row");

    var id = $(this).find("td:first").text();

    $.getJSON('@Url.Action("GetSupplierInfo")', {
        id: id }, callback);

    function callback(result) {
        $("#InfoPanel").show("slow");
        $("#CompanyName").html(result[0].SupplierName);
        $("#ContactName").html(result[0].SupplierContact);
        $("#Country").html(result[0].SupplierCountry);
        $("#Phone").html(result[0].SupplierPhone);
        $("#Fax").html(result[0].SupplierFax);
    }
});
});

$("#td").click(function () {
    var t = $.trim($(this).text());
    if (t == "Редактировать") {
        $("#ProductID").val($(this).parent().find("td").eq(0).text().trim());
        $("#ProductName").val($(this).parent().find("td").eq(1).text().trim());
        $("#SupplierID").val($(this).parent().find("td").eq(2).attr("Tag").trim());
        $("#CategoryID").val($(this).parent().find("td").eq(3).attr("Tag").trim());
        $("#QuantityPerUnit").val($(this).parent().find("td").eq(4).text().trim());
        $("#UnitPrice").val($(this).parent().find("td").eq(5).text().trim());
        $("#UnitsInStock").val($(this).parent().find("td").eq(6).text().trim());
        $("#UnitsOnOrder").val($(this).parent().find("td").eq(7).text().trim());
        var b = $(this).parent().find("td").eq(8).text().trim().toLowerCase() === 'true';
        $("#Discontinued").attr('checked', b);
    }
    action = '@Url.Action("EditProduct", "Product")';
    $("#ProductWindow").dialog("open");
}
);

if (t == "Удалить") {
    action = '@Url.Action("DeleteProduct", "Product")';
    $.post(action, { productId: $(this).parent().find("td").eq(0).text().trim() },
        function () {
            $("#table").find("tr[class=selected-row]").remove();
            alert("Продукт удален");
        }
    );
}

```

## Заключение

В этой статье я постарался рассказать, насколько просто и удобно работать с jQuery Ajax. Были продемонстрированы функции .getJSON() и .post(). Эти функции являются упрощенными версиями основной функции jQuery .ajax(), которая предоставляет гораздо больше возможностей.

ASP.NET MVC отлично сочетается с jQuery, не с проста же ее включили в состав проекта ASP.NET MVC. И не смотря на то, что проект jQuery не принадлежит компании Microsoft, он активно поддерживается и регулярно выходят новые версии библиотеки. Сообществом разработано множество плагинов, а именно: Ajax forms, Ajax Grid и т. д. Это позволит вам всегда найти нужный функционал без необходимости написания кода самому.

С помощью jQuery можно создавать поистине удивительные вещи без написания низкоуровневого JavaScript кода. При этом весь написанный код будет поддерживаться всеми популярными браузерами.

В заключение хочу сказать, что у разработчиков на платформе Microsoft появился потрясающий инструмент для разработки современных, динамичных web приложений любой сложности. ■



## Управление руководителями проектов



### Авторы

**Борис Романов**  
boris.romanov@projectsprofiler.ru  
Projects Profiler LLC  
<http://projectsprofiler.ru>  
г. Москва

**Андрей Соловьев**  
zen@cs.isa.ru  
ИСА РАН  
г. Москва

**Дмитрий Соловьев**  
sol@cs.isa.ru  
ИСА РАН  
г. Москва

**А** любому профессиональному, работающему в сфере IT, хорошо знаком такая персонаж, как Руководитель проекта или РП. Обязанности РП и типовые алгоритмы его действий зафиксированы в целом ряде методологий. Проблематика проработана настолько глубоко, что в ее рамках уже появились своя «классика» (PMBOK, Oracle AIM или Microsoft MSF) и свои «иконоборцы» (Scrum, XP прочие Agile-методологии). Количество же программ, автоматизирующих деятельность РП – просто не поддается исчислению.

Все эти методологии и программы оставляют за рамками важный класс руководителей, которые также в некотором смысле руководят проектами, но их деятельность радикально отличается от той, которой занимаются РП. Речь идет о руководителях, которые ответственны не за один, а сразу за несколько (может быть десятки или даже сотни) проектов выполняемых их подчиненными. Таких немало в IT-сообществе. Это руководители отделов разработки IT-фирм, руководители небольших компаний по разработке программного обеспечения, руководители IT отделов организаций, сотрудники проектных офисов. Они не руководят проектами напрямую (для этого есть РП), но курируют их отбор, обеспечивают общий успех и решают проблемы, возникающие «на стыке». Далее мы подробно оста-

новимся на задачах, типовых для руководителя, контролирующего много проектов, но не существующих для РП в рамках одного проекта. Заметим, что в плане непосредственного управления, такой руководитель управляет только руководителями проектов, являясь, фактически, «руководителями руководителей проектов», или РРП. Именно эту «рычащую» аббревиатуру мы и будем далее использовать в статье.

Очевидно, что РРП несет на себе больший объем ответственности, чем руководитель отдельного проекта, ибо он прямо или косвенно управляет гораздо большим количеством людей. На нем гораздо больше оказывается знаменитая изменчивость требований в IT-проектах, вызванная ошибочным ощущением заказчиков, что программный код можно легко исправить в любой момент. Тем самым, для него гораздо больше, чем для РП актуальна автоматизация рутинных ежедневных процессов. В связи с этим вызывает удивление, что на рынке практически отсутствует программное обеспечение, «заточенное» специально под нужды РРП.

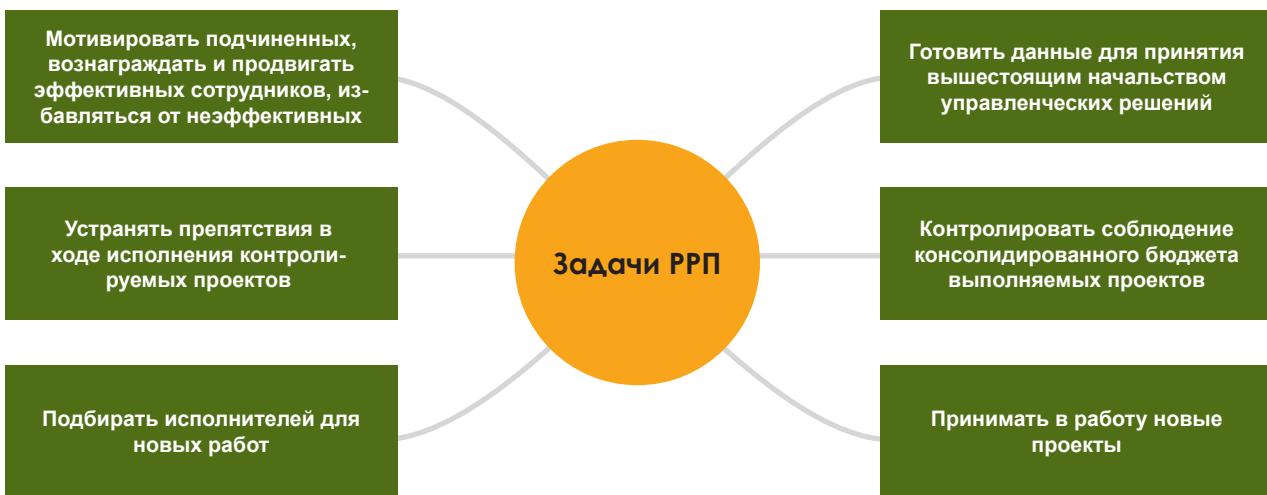
«Как же отсутствует!» – возразит внимательный читатель. Есть же системы управления портфелями проектов, есть MS Projects Server с его замечательными диаграммами загруженности. Даже для трекинговых систем есть

надстройки, обеспечивающие много-проектность.

Но при практическом использовании этих решений сразу становится видно, что этот функционал «прикручен» к системам, изначально нацеленным на один проект. Они не проектировались «с нуля» под решение конкретных проблем РРП (примеры которых мы опишем в следующем разделе). Поэтому и пользоваться такими системами на каждодневной основе неудобно.

Если же мы посмотрим на серьезные системы управления портфелем проектов, например, Oracle Primavera или SAP RPM, то они конечно удовлетворят потребности РРП. Однако их функционал гораздо более широк, системы достаточно громоздки и затратны во внедрении, как по времени, так и по ресурсам. В самом деле, сложно себе представить ситуацию, когда руководитель подразделения, которому нужно быстро и без затрат решить свои конкретные задачи, для своих 20 проектов будет использовать модуль от SAP или Oracle.

Итак, рассмотрим типовые задачи РРП, которые должна помогать решать ориентированная на его нужды автоматизированная система и предложим их техническую реализацию. ►



### Типовые задачи и способы их решения

По роду своей деятельности РРП должен регулярно решать различные задачи. Помощь РРП в решении этих задач со стороны специализированной программы должна заключаться в том, чтобы обеспечить его необходимой информацией по проектам для принятия управляющих решений, а также предупредить о надвигающихся или уже наступивших проблемах. На практике, исходная информация о проектах часто находится в разных системах ведения проектов и даже в трекинговых системах. Тем не менее, РРП нуждается в том, чтобы эта информация была представлена ему консолидировано и наглядно.

Для каждой задачи будет рассмотрен:

- набор ситуаций, вынуждающих РРП ставить перед собой и решать разбираемую типовую задачу;
- возможные варианты разрешения ситуации и информация, которую программа должна предоставить РРП для принятия решений.

### Подобрать исполнителей для новых работ

Опишем примеры ситуаций, которые ведут к возникновению перед РРП этой задачи.

#### Ситуация: проект отстает от графика

##### Решение

Решений два: привлечь ресурс извне (из других подразделений или нанять) или изыскать «внутренние резервы». Прежде всего, должен быть исследован второй вариант. Внутренние резервы – это перераспределение работ (например, отказ от части работ) или усиление проекта сотрудниками за счет других проектов.

#### Информация, необходимая для принятия решения

Для разрешения ситуации РРП необходимо получить следующую информацию:

- ✓ плановая дата проблемной контрольной точки проекта;
- ✓ скорость выполнения проекта;
- ✓ прогнозируемая дата контрольной точки при данной скорости выполнения;
- ✓ перечень проектов, приоритетность или скорость выполнения которых позволяет забрать из них людей;
- ✓ квалификации кандидатов на усиление проекта «А»: смогут ли они справиться с работой. Это

очень важная информация, ибо в зависимости от квалификации, производительность сотрудников может отличаться на порядок;

- ✓ история работы кандидата: работал ли он уже в этом проекте или имел дело с подобными задачами в других проектах.

#### Ситуация: возникла срочная работа

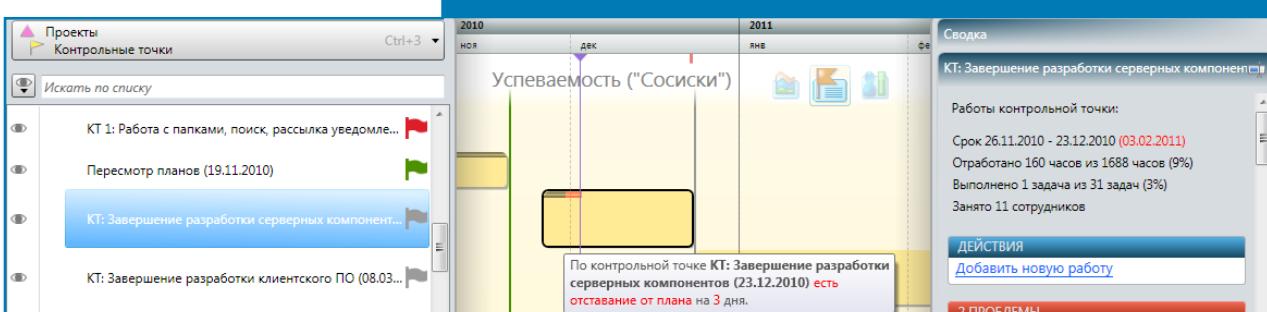
Часто бывает так, что необходимо отвлечь участника какого-то проекта на срочную работу, например, для усиления отстающего проекта, выполнения работ по прессейлу, помочи сотрудникам технической поддержки, выполнения специального задания руководства, для командировок и т. п.

##### Решение

Проанализировать загруженность сотрудников в проектах, понять, откуда можно извлечь сотрудников нужной квалификации с наименьшими негативными последствиями. Возможно, перераспределить сотрудников для минимизации последствий.

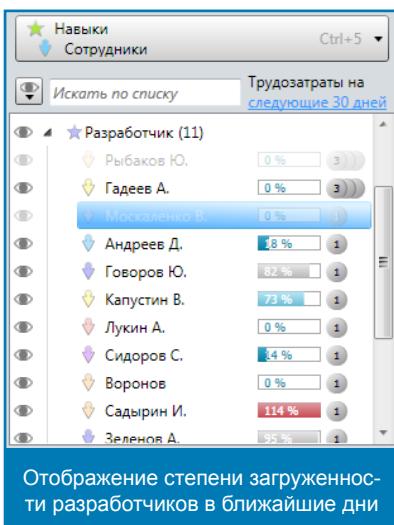
#### Информация, необходимая для принятия решения ▶

Диаграмма успеваемости отражает степень отставания работ от графика и прогнозируемую дату выполнения работ контрольной точки



Для разрешения ситуации РРП необходимо получить следующую информацию:

- ✓ скорость выполнения проектов для определения, из каких можно извлекать ресурсы;
- ✓ информацию о квалификации и опыте работы сотрудников, чтобы выбрать того, кто поможет разрешить ситуацию, а также для определения кандидатов на затыкание «дыр».



## Задача Вознаграждать и приводить эффективных сотрудников, избавляясь от неэффективных

Опишем примеры ситуаций, которые ведут к возникновению перед РРП этой задачи.

### Ситуация: регулярный анализ эффективности сотрудников

Как руководитель, РРП должен прово-

дить определенную кадровую политику в своем подразделении, в частности, регулярно вознаграждать эффективных сотрудников и «взбадривать» отстающих.

## Решение

Чтобы эффективно действовать в этой ситуации РРП необходимо получить следующую информацию:

- ✓ оценки эффективности для своих сотрудников;
- ✓ динамику роста квалификации;
- ✓ факты участия в сложных, дающих важный опыт проектах;
- ✓ загруженность в прошлом и сейчас (не лентяй ли сотрудник).

### Ситуация: удержание эффективных сотрудников

Как руководитель, РРП должен чутко улавливать ситуации, когда какой-то нужный сотрудник собирается покинуть команду, в силу семейных причин, кризиса возраста, недовольства атмосферой в компании и т. п. Для этого необходимо сформировать «ядро» своего подразделения, то есть перечень сотрудников, которых нельзя терять, и следить за тем, чтобы члены «ядра» находились в рабочем состоянии.

## Решение

Для разрешения ситуации РРП необходимо получать и регулярно освежать информацию, состав которой изложен в описании предыдущей ситуации.

### Ситуация: подбор фактов для увольнения

В любом коллективе есть какой-то про-

цент неэффективных членов. Их поголовье не должно быть слишком велико. Опытный руководитель вычисляет таких довольно быстро, но ему не всегда удается найти формальный повод для их увольнения.

## Решение

Для разрешения ситуации РРП необходимо получить следующую информацию:

- ✓ историю участия сотрудника в проектах;
- ✓ историю динамики его загрузки за несколько периодов;
- ✓ историю динамики его квалификации.

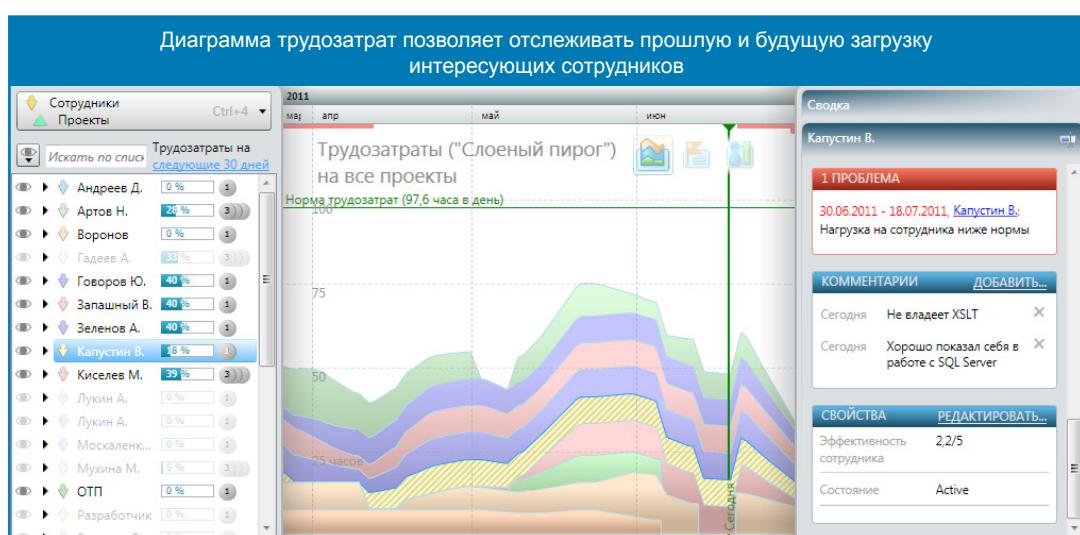
## Задача Устранить препятствия в ходе исполнения контролируемых проектов

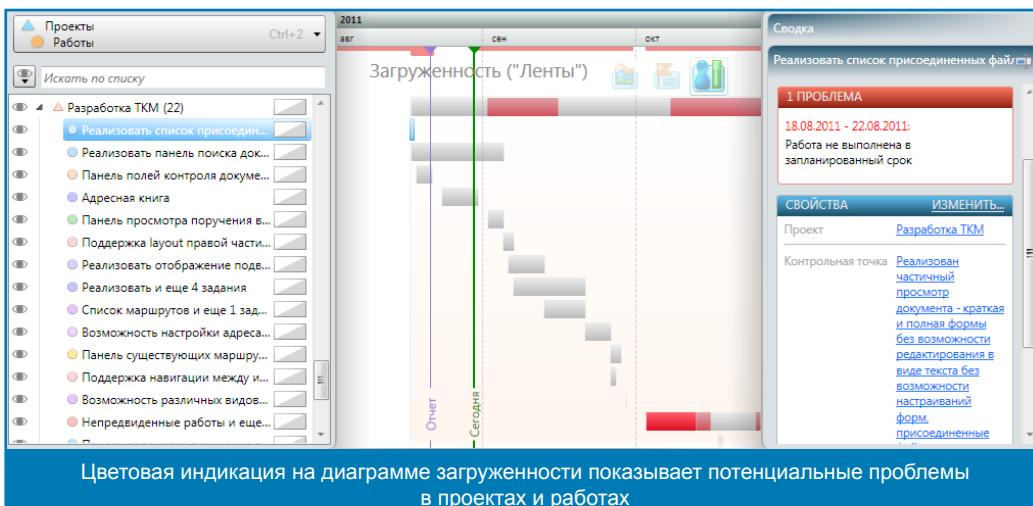
К возникновению этой задачи может привести, например такая ситуация. РРП видит по внешнему проявлению, что состояние работ в подразделении не благополучно. Что это за «внешние проявления»? Например, уволился один из ведущих разработчиков или выявились серьезные ошибки в проектировании или неожиданный и объемный пресейл «вырвал» разработчиков из графика.

## Решение

Для принятия решений по данной ситуации РРП должен получить конкретные параметры состояния проектов в новых условиях:

- ✓ диаграммы и сводки по «пострадавшим» проектам;
- ✓ загруженность сотрудников, которые могли бы прикрыть образовавшиеся «дыры»;





- ✓ моделирование различных ситуаций, чтобы оценить эффект от своих решений;
- ✓ диаграммы и сводки по проектам с учетом корректирующих данных.

## Задача Принять в работу новые проекты

Опишем примеры ситуаций, которые ведут к возникновению перед РРП этой задачи.

**Ситуация:**  
сэйл хочет понять, когда мы можем взять в работу новый проект

«Когда сможем взять в работу этот проект?» – пожалуй, один из самых частых вопросов, который задают РРП. И почти всегда ответить на него нужно срочно, ибо договор или «уже почти заключен», или «мы не можем упустить такую возможность», и тому подобное. Необходимо либо обнадежить или, наоборот, твердо отказать. И в том и в другом случае решение должно быть основано на реальных фактах, чтобы не обмануть ожидания в первом случае и дать аргументированный ответ во втором.

### Решение

Для разрешения ситуации РРП необходимо получить следующую информацию:

- ✓ когда у сотрудников нужной квалификации есть «окна» в их графике;
- ✓ какова загрузка этих сотрудников (может быть, сотрудник какое-то время сможет участвовать в текущем и новом проекте);
- ✓ загруженность эти сотрудников в текущих проектах;
- ✓ состояние проектов, в которых эти сотрудники участвуют (скорость

выполнения, важность) – может быть их можно приостановить без особого ущерба.

**Ситуация:**  
генеральный директор требует срочно запустить проект

Генеральный директор (или другой «большой босс») хочет запустить проект. Подвести его нельзя, поэтому если мы соглашаемся, то должны быть на 101% уверены, что ресурсов в необходимый момент будет достаточно. Или же, если прямо сейчас такой возможности нет – нужно четко и аргументировано перечислить, какие проекты пострадают, если мы все-таки запустим проект прямо сейчас.

### Решение

Чтобы смоделировать последствия срочного запуска новых проектов, РРП необходимо получить следующую информацию:

- ✓ наличие сотрудников нужной квалификации;

состояние идущих проектов, в которых они участвуют;

- ✓ загруженность сотрудников в этих проектах.

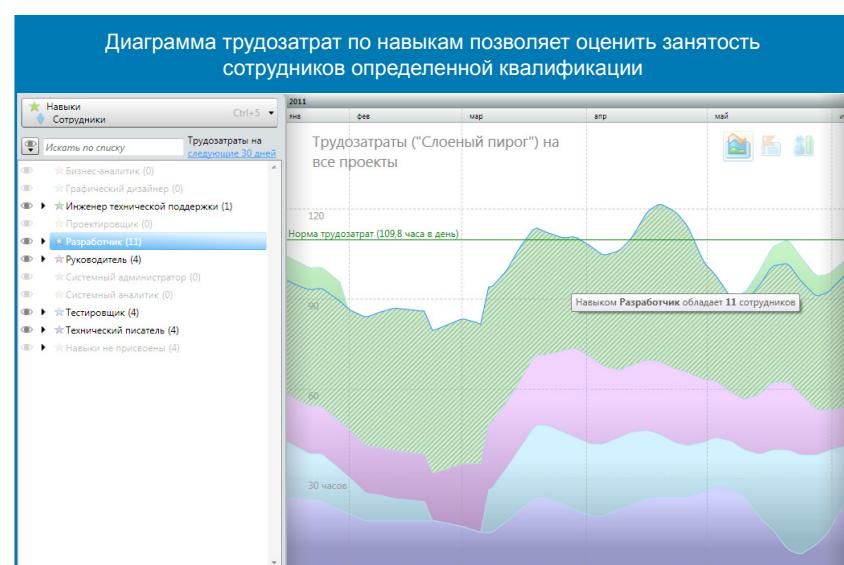
## Задача Готовить данные для принятия вышестоящим начальством управлени-ческих решений

Начальство имеет обыкновение задавать вопросы относительно состояния дел в самый неожиданный момент и отвечать нужно быстро и точно.

### Решение

Для подготовки данных РРП нужно получить следующую информацию:

- ✓ перечень «безпроблемных» проектов (в том числе в виде наглядных диаграмм);
- ✓ перечень «проблемных» проектов (в том числе в виде наглядных диаграмм); ►



для проблемных проектов получить список проблем;

- ✓ загрузка сотрудников, чтобы подготовиться к вопросам о вариантах решения проблем.

## Задача

### Контролировать соблюдение консолидированного бюджета выполняемых проектов

Опишем ситуации в проектах, которые ведут к возникновению перед РРП этой задачи.



**Ситуация:**  
есть подозрение, что превышен бюджет по отдельным проектам или по всему портфелю

#### Решение

Для проверки подозрения, РРП нужно получить фактические трудозатраты по соответствующим проектам (сводки) или по всем проектам подразделения в целом.



**Ситуация:**  
РРП необходимо понять, удастся ли удержаться в рамках бюджета до конца года (квартала/полугодия)

РРП должен уложиться в бюджет периода (например, года, полугодия, квартала), а если это не получается, то заранее быть к этому готовым.

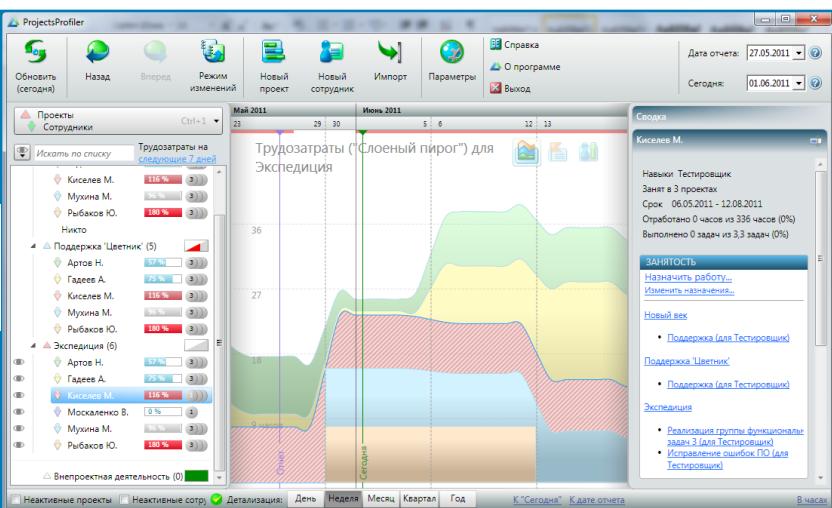
#### Решение

Для разрешения ситуации РРП нужно получить следующую информацию:

- ✓ фактические трудозатраты по всем проектам;
- ✓ прогноз затрат до конца периода.



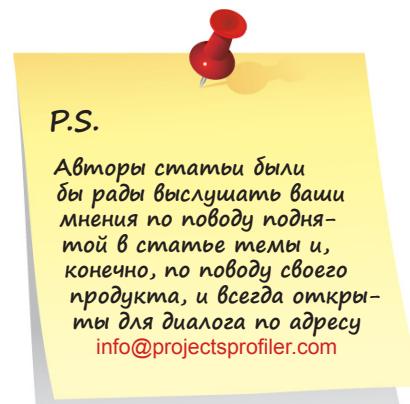
# Projects Profiler



В результате был создан программный продукт ProjectsProfiler, который автоматически загружает информацию из систем управления проектами, обрабатывает ее, и формирует для руководителя обобщенные представления о состоянии дел в проектах и занятости сотрудников. Программа является настольным приложением и использует СУБД Microsoft SQL Server Compact Edition, может использоваться на настольном компьютере или ноутбуке под управлением Windows. Необходимым условием успешного применения программы является наличие в компании определенного порядка ведения проектов с помощью автоматизированных средств управления. В этом случае внедрение ProjectsProfiler дает возможность использовать уже имеющуюся инфраструктуру гораздо более эффективно, сохраняя при этом привычные и отлаженные методы работы.

## Заключение

Нужды людей, занимающихся управлением РП, отличаются от нужд, покрываемых современными программами для управления проектами и портфелями проектов. Поэтому для их информационной поддержки требуется специальный класс программного обеспечения. Сейчас эта рыночная ниша еще слабо заполнена, но мы надеемся, что в ближайшее время это положение вещей изменится. В том числе и благодаря нашему продукту ProjectsProfiler. ■



# Фоторепортаж «День студента Microsoft 2011»



## Автор

Станислав Горнаков  
@Gornakov  
gornakov@hotmail.com  
www.gornakov.com



Компания Microsoft 14 октября провела ежегодное мероприятие под названием «День студента Microsoft». Мероприятие прошло в центре Digital October. По сложившейся традиции Microsoft организовала прямую трансляцию этого события в Интернете. Благодаря чему студенты от Чукотки до Калининграда смогли наблюдать в прямом эфире за всеми выступлениями и даже поучаствовать в online конкурсе.



Открытие мероприятия было поручено «чумовым барабанщикам», которые «свалились на голову» всем собравшимся не понятно откуда, показав зажигательный и фееричный номер.



После выступления барабанщиков на сцене появился Дмитрий Сошников, координатор академических программ департамента стратегических технологий Microsoft. В этот день Дмитрий вообще блистал и представил массу интересных докладов.

3



После короткого вступления, Дмитрий пригласил на сцену Николая Прянишникова, президента Microsoft Россия.

4



Николай, побывав несколько минут на сцене в пиджаке, пожаловался на Александра Ложечкина, который сказал ему за кулисами, что тот похож на преподавателя, поэтому Николай предпочел снять с себя пиджак и остаться в рубашке.

5



Отсутствие пиджака не помешало Николаю произнести интересную речь, поведав студентам об инновациях в нашей с вами повседневной жизни. Все собравшиеся слушали Николая Прянишникова с особым вниманием.

6



В середине своего выступления Николай пригласил на сцену Альберта Ефимова, представителя кластера информационных технологий фонда «Сколково».

7



Альберт простывшим голосом рассказал много чего интересного, а также акцентировал внимание на том, что Microsoft заключала договор с открытым университетом «Сколково» по проведению отбора на конкурс Imagine Cup.

8



9

После выступления Альбера Ефимова на сцене появились любители оригами, участники Imagine Cup от России.



10

Ксения сложила в оригами голубя с помощью Kinect.



11

Затем со своим докладом «Майкрософт для студентов «Что вы услышите сегодня, и что делать потом» вышли Дмитрий Сошников и Евгения Левинская.



12

Дмитрий открыл для всех студентов страшную тайну. Оказывается перевод слова «Студент», звучит не иначе, как усердно работающий, что вызвало в зале легкую иронию.



13

По окончании выступления Дмитрия и Евгении, на мероприятии был объявлен короткий перерыв. После перерыва вышел Глеб Архангельский с докладом «Тайм-менеджмент: как успевать жить и работать». Честно говоря, никогда не улавливал полезности этой темы и всегда удивлялся, кому нужна эта «вода», поэтому пошел смотреть в другой зал Дмитрия Сошникова.



14

А в другом зале было действительно интересно. Там Дмитрий выступил с докладом «Imagine Cup: не бойся мечтать!».



15

Дмитрий выдал массу полезной информации о том, что такое Imagine Cup и почему нужно в нем участвовать, вызвав по ходу выступления еще раз любителей оригами из команды Imagine Cup.



14 октября

16

В главном зале на смену Глебу Архангельскому пришел Аркадий Морейнис из компании «Главстарт».



17

Аркадий Морейнис представил хороший доклад под названием «Своя идея: откуда взять и что с ней делать?». Ключевая фраза доклада: «Прежде чем заниматься Тайм-менеджментом и чего-то записывать в Outlook, нужно что-то придумать!».



18

После Аркадия Морейниса в главном зале состоялся конкурс «Сочини гимн Windows 7». Там даже пели рэп.



19

А тем временем в другом зале Digital October проходил доклад об истории успеха молодых сотрудников Microsoft с участием Константина Кичинского, Натальи Малыкиной и Елены Павловой.



20

Первым взял слово Константин Кичинский, академический евангелист компании Microsoft, который рассказал о своей истории прихода в компанию Microsoft.



21



Затем выступила Наталья Малыкина с историей одного e-mail, который изменил ее жизнь и привел в Microsoft.

22



И в конце Елена Павлова рассказала о программе MSP, о том, почему не нужно бояться преподавателей, и как же красива эта Барселона.

23



Одним из ключевых докладов на дне студента стал доклад «Платформа Microsoft и концепция трех экранов» или «Вместе в облака» от Александра Ложечкина, руководителя департамента стратегических технологий Microsoft.



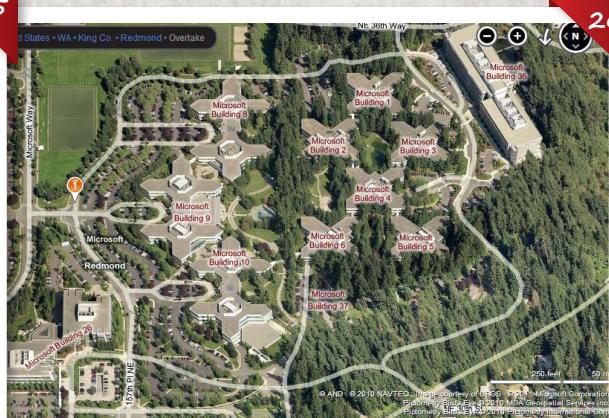
Потрясающе, насколько Евангелисты Microsoft и в частности Александр Ложечкин профессионалы своего дела. Представьте, в это утро мы с ним активно дискутировали по электронной почте, обсуждая рабочие моменты и буквально за 20 минут до начала своего выступления Александр мне все еще отвечал по почте даже со своим Windows Phone 7, стоя за кулисами. Александр представил интересный рассказ о студенте, который жил 15 лет назад (в комнате со стеклопакетами и крутыми батареями), показывая на конкретном примере, как поменялся наш информационный мир.

25



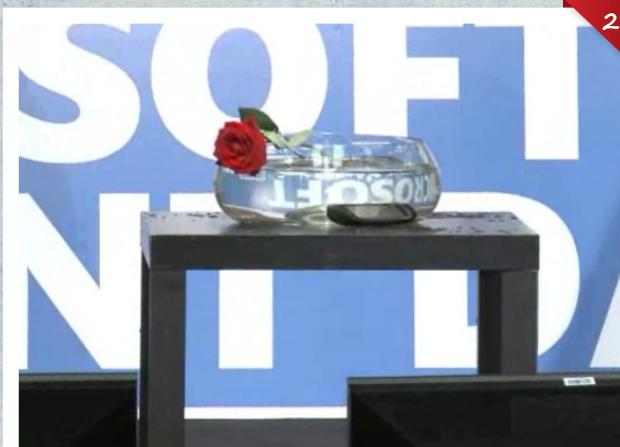
По ходу доклада на сцену была приглашена Татьяна Бойцова, студент-партнер Microsoft (MSP), которая рассказал об облаках и работе над общими документами в сервисе SkyDrive.

26



Татьяна в показательном примере задействовала сервис Bing Maps, показав с высоты птичьего полета кампусы Microsoft в городе Редмунд США. Корпуса

Microsoft построены в виде креста, как заметил Александр Ложечкин, в штаб-квартире Microsoft все хотят иметь офис с окном, а это форма построек как раз дает максимально возможную отдачу в виде офисов с окнами.



27

В середине выступления, Татьяна упустила смартфон Windows Phone 7 в воду.



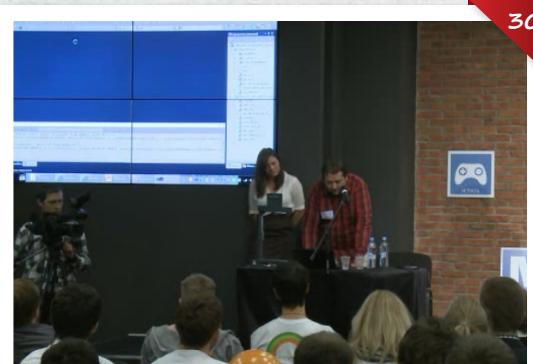
28

Один из студентов сразу же попросил этот утопленный телефон в качестве сувенира, но Windows Phone 7 даже спустя три минуты все еще продолжал работать. Студенту не повезло, и телефон остался реквизитом.



29

Казалось бы, все данные из телефона потеряны навсегда, но благодаря тому, что вся информация с телефона хранилась в облаках, данные благополучно восстановлены в память уже с другого смартфона Windows Phone 7.



30

Татьяне с докладом помогал Дмитрий Сошников.



31

В конце доклада Дмитрий Сошников с Александром Ложечкиным мучили кротов с помощью Kinect. А также предложили студентам посетить Tech End 2011 со скидкой на входной билет.



32

Следующий доклад в главном зале носил название «Windows Phone 7 – возможность реализовать себя». Доклад представил Дмитрий Сошников.



### Прогнозы IDC и Gartner

Операционная система	IDC	Gartner
Android	45%	49%
Symbian	0%	0%
<b>Windows Phone</b>	<b>21%</b>	<b>20%</b>
iOS	15%	17%
BlackBerry OS	14%	11%
Other (Linux, webOS, Maemo/MeeGo)	5%	3%
Grand Total	100%	100%

Дмитрий представил весьма интересную таблицу с прогнозами по популярности Windows Phone 7 в ближайшем будущем.



Последним выступлением в главном зале было выступление Владимира Юнева с докладом «Windows Azure – дорога в облака». Владимир рассказал массу интересного материала о Windows Azure.

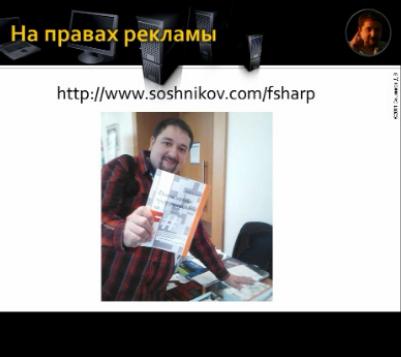
### БЕСПЛАТНАЯ СЕРТИФИКАЦИЯ: WINDOWS AZURE



В конце своего выступления Владимир анонсировал для студентов возможность бесплатной сертификации по Windows Azure.



В это время в другом зале Digital October параллельно Владимиру, с докладом «Эзотерическое программирование» выступал Дмитрий Сошников, быстренько перебравшийся из главного зала.



Между делом, Дмитрий представил свою книгу F#.



Заключительным аккордом мероприятия стало выступление команды КВН и коллективная фотография на память.



# Интервью с Яной Курановой

## Microsoft Student Partner

Беседовал Станислав Горнаков



**Станислав Горнаков:** Яна, добро пожаловать на страницы журнала MSDeveloper.RU. Так получилось, что вы первая девушка, а также первый Microsoft Student Partner (MSP), который будет давать интервью нашему журналу. Яна, 18 лет возраст скажем так достаточно юный для IT сектора, как вы смогли, и как у вас получилось стать MSP?

**Яна Куранова:** Здравствуйте, Станислав. Для меня большая честь общаться с вами на страницах журнала MSDeveloper.ru, читательницей которого я являюсь. Ваши статьи помогают мне найти ответы на многие вопросы, познакомиться с интересными людьми и погрузиться в атмосферу будущего.

### Наше досье

В 2009 году стала призером Всероссийской аэрокосмической олимпиады для школьников по математике.

В 2009 году окончила ОГОУ Астраханский технический лицей с серебряной медалью.

В 2009 году участвовала в Форуме Победителей 2009.

С 2009 года обучается в Астраханском государственном техническом университете.

В 2010 году вывела новую формулу «Условие перехода волны из одной среды в другую» и стала победителем 60-й международной студенческой отраслевой научно-технической конференции.

В 2011 году стала победителем 3-го Каспийского Инновационного форума 2-ой региональной конференции молодых ученых и инноваторов «Инно - Каспий» в секции «Механика, физика, машиностроение».

С 2011 года является Microsoft Student Partner.

Имеет несколько публикаций в научных журналах.

Участница различных конференций, городских эстафет.

Живет и обучается в Астрахани.

Любопытство – не знаю порок это или достоинство? Но это качество всегда толкало меня к поиску нового, неизведанного. И этот случай (участие в программе Microsoft Student Partner) не исключение. Узнав о конкурсе, я подумала: «Вот и еще одна возможность реализовать свои идеи в IT секторе». Заполнила анкету, сняли видео и с нетерпением ждала результат. И вот я MSP!



По большому счету в IT индустрии девушек мало, почему выбрали именно эту сферу?

Если мужчина прекрасно умеет готовить, почему бы ему не стать поваром, ведь он занимается любимым делом. Так и я люблю свою будущую профессию.

В свое время передо мной стоял выбор либо творческой профессии, связанной с гуманитарными науками, либо профессии технического профиля. Профессия программиста разрешила все мои колебания и предоставила отличный шанс для реализации всех творческих идей и задач технического характера, ведь в мире так много нерешенного. Считаю, что выбор был сделан верно – за нами будущее.



Кстати, а как учитесь: отличница, хорошистка или так себе?

Удобнее всего ходить в середнячках, но это не мое. У меня много идей и способов претворения их в жизнь, поэтому с шести лет длится роман с оценкой «5». ▶





Помимо учебы чем-то еще занимаетесь, увлекаетесь, может, есть какое-то интересное хобби?

Да, так хочется в жизни успеть все. Кроме учебы в университете и самообразования, большой акцент я стараюсь делать на самосовершенствовании не только в IT, но и в сфере здорового образа жизни. Увлекаюсь йогой, которая помогает мне расслабиться, ведь в наши дни это так сложно уйти от всех проблем и погрузиться полностью в себя. Также веду свой блог ([kyana-msp.blogspot.com](http://kyana-msp.blogspot.com)), читаю книги, изучаю английский язык (учусь по дополнительной программе «Технический переводчик в сфере профессиональных коммуникаций»). Вы удивляетесь, что в IT-сфере так мало девушек, но еще меньше девушек, которые разделяют мое хобби – рыбалку.



Как вы узнали о программе Microsoft Student Partner и почему решили в ней участвовать?

Я узнала о данной программе от своего преподавателя А. В. Морозова в конце второго курса. За два года учебы в институте я участвовала в конференциях, сдавала лабораторные работы, курсовые, но чего-то в моей жизни на тот момент не хватало, может азарта? Но вот закончилась лекция, я пришла домой, прочла о данной программе в интернете и сказала: «Хочу!».

Почему я решила в ней участвовать?



Данная программа имеет множество плюсов: изучение новых технологий Microsoft, общение в кругу IT-профессионалов и т. д. Также не малое влияние на меня произвела встреча с Дмитрием Андреевым в User Group, спасибо ему за это!



А какова общая цель участия в данной программе?

Моя цель – сделать мир лучше, так же как и у компании Microsoft. Вот еще один плюс для меня, у нас с компанией общие цели – это очень важно. Для себя хочу извлечь и бесценный жизненный опыт общения с людьми, приобрести организаторские способности, изучить новые технологии.



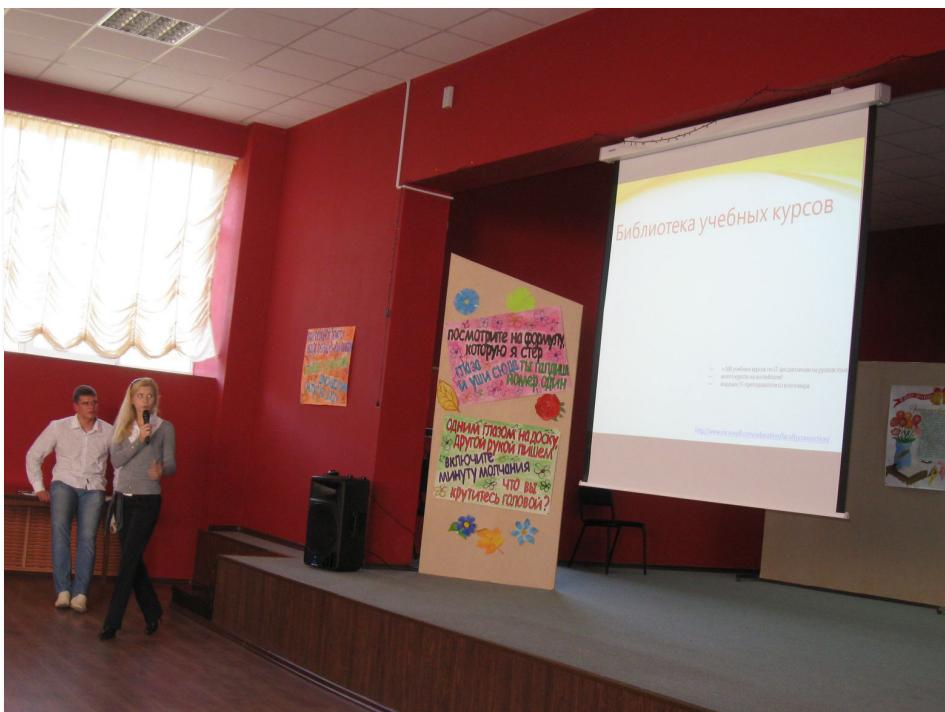
Насколько трудно, а может быть, и интересно было готовить анкету и видеорезюме для жюри?

Составить материал было не трудно, труднее всего было заснять видео, ведь меня снимала моя мама, которая непроизвольно начинала улыбаться во время съемки =)



Я знаю, вы были на MSPSummit, каковы ваши впечатления от этого мероприятия?

Да, Станислав, я была на этом потрясающем мероприятии!



сающем мероприятии. Организаторы постарались на славу! Все пронеслось настолько быстро, что хотелось бы перемотать все назад и еще раз прожить эти мгновения. Все ребята очень интересные, на какой-то момент мне показалось, что мы одна большая семья, хотя были знакомы всего лишь два дня. Выставка «Игромир» очень впечатлила, это был мир будущего. Развлекательная программа в «Самолет» была интересной, позволила сблизиться с некоторыми участниками, ведь нас разделили на две команды и дали задание на ночь. Весь второй день MSP Summit мы провели в офисе Microsoft, и я посетила его впервые. Тренинги и лекции были очень интересными, я узнала много нового и сделала некоторые выводы для себя.



**Как отнеслось ваше окружение к тому, что вы стали MSP?**

Мои родители и близкие друзья гордятся мной и поддерживают во всех начинаниях. Многие знакомые тоже заинтересовались этой программой.

**Каковы планы на будущее, а также ваши текущие задачи?**

Планов очень много, стараюсь расставлять по приоритетам и так, чтобы при их претворении в жизнь получать удовольствие, а не хроническую усталость. В первую очередь как для MSP, для

меня текущими задачами является выполнение плана, который я составила на год. Хотелось бы стать участницей форума за рубежом и познакомиться с опытом иностранных коллег. После посещения MSP Summit у меня возникла идея запустить свой стартап, надеюсь, что все получится. Ну и естественно никто не отменял учебу в университете и ее успешное завершение.

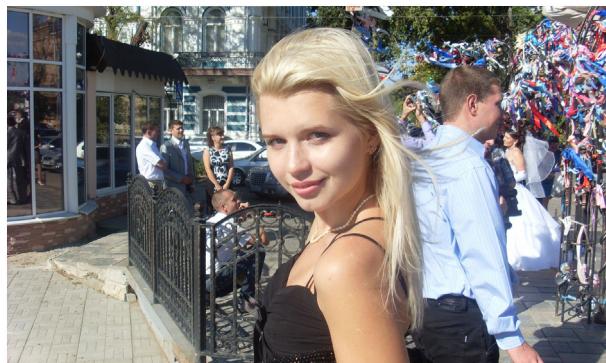
разрабатываю несколько интересных приложений, которые возможно заинтересуют компанию.

**Что же, Яна, желаю вам успехов и удачи, а также надеюсь в вашем лице получить постоянного читателя журнала MSDeveloper.RU.**

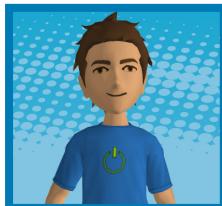
**Яна, есть ли у вас новые идеи для компании Майкрософт?**

Конечно же есть, на данный момент я

Спасибо, Станислав, было очень интересно с вами общаться. Вам тоже удачи в продвижении журнала MSDeveloper.RU и увеличении числа читателей. ■



# Развертывание операционных систем Windows при помощи флэш-накопителя



## Автор

Никита Львов  
MSP  
mgupi-it.ru  
г. Москва



**Н**ачиная с Windows 95, операционные системы от Microsoft выпускаются на CD дисках, а в связи с увеличением объема установочных файлов, операционные системы начиная с Windows Vista выпускаются на DVD-дисках. Оптические диски относительно недолговечны, а также обладают относительно низкой скоростью чтения данных. Поэтому порой более целесообразно выполнять установку операционной системы с флэш-накопителя. Данный метод развертывания незаменим при установке ОС на ультрапортативные ПК (такие как нетбук). Так же при развертывании ОС с использованием файла ответов, его приходится записывать на дискету или флэш-накопитель. Использование же флэш-накопителя при развертывании существенно упростило бы этот процесс, т. к. и дистрибутив системы, и файл ответов будут находиться в одном месте, причем такая установка не будет требовать участия пользователя (при условии, что файл ответов будет иметь имя Autounattend.xml). Все эти преимущества натолкнули меня на тему данной статьи.

## Теоретические сведения о процессе развертывания при помощи установочного флэш-накопителя

Начиная с Windows Vista, при развертывании применяются файловые образы WIM. То есть теперь установочный диск содержит не множество сжатых системных файлов, а образы WIM, содержащие в себе файлы, копируемые на компьютер при установке. Более подробные сведения о формате образа WIM можно прочитать в книге «Развертывание и настройка Windows Server 2008» издательства Microsoft Press.

На любом установочном диске есть два файла:

**Boot.wim** – содержит в себе Windows PE, которая загружает компьютер при загрузке с DVD-диска и создает среду для работы программы установки,

**Install.wim** – содержит в себе файлы, копируемые на компьютер при установке.

Но также диск должен содержать в себе загрузочную запись для того, чтобы компьютер мог с него загрузиться и начать загрузку Windows PE. Скопировать эту загрузочную запись вручную никак нельзя, т. к. она находится в скрытом разделе диска.

Вообще установка с флэш-накопителя мало чем отличается от установки с DVD-диска, основные различия наблюдаются в загрузочной записи.



Поэтому, главной проблемой при развертывании с использованием установочного флэш-накопителя, является создание на нем корректной загрузочной записи, аналогичной той, что используется в Windows при установке на жесткий диск.

Для создания загрузочного ISO-образа с системой Windows PE (которая может быть произвольно отредактирована, вплоть до добавления программы setup.exe и прочих файлов установочного диска), используется программа oscdimg. Но использовать такой метод с флэш-накопителем нельзя по некоторым причинам:

- ▼ редкость программ для записи ISO-образов на флэш-накопитель,
- ▼ различия в загрузочной записи.

В процессе работы я смог решить эти проблемы. Существуют два варианта их решения:

1. Использование специализированных программ.
2. Создание загрузочной записи вручную.

## Использование специализированных программ

В процессе работы я подготавливал загрузочные флэш-накопители при помощи двух программ:

- ▼ Windows 7 USB DVD Download Tool (Microsoft)
- ▼ WinToFlash(Novicorp)

## Работа с программой Windows 7 USB DVD Download Tool

Данную программу можно использовать для развертывания ОС Windows 7.

1. Запустите программу соответствующей командой из меню **Пуск**.
2. В появившемся окне нажмите на кнопку **Browse** и выберите ISO-образ установочного DVD-диска Windows 7.
3. Очистите флэш-накопитель от имеющейся информации.
4. Далее нужно нажать кнопку **USB Device**.
5. Появится окно (рис. 1). В данном окне выберите имя флэш-накопителя, который будет подготовлен для использования в качестве установочного.
6. Нажмите кнопку **Begin copying**.
7. Начнется процесс форматирования и копирования данных, требуемых для установки. ▶

## Администрирование

8. По окончании работы программы будет выведено соответствующее окно (рис. 2).
9. Закройте окно программы. Установочный флэш-накопитель с Windows 7 готов к работе.

### Работа с программой WinToFlash

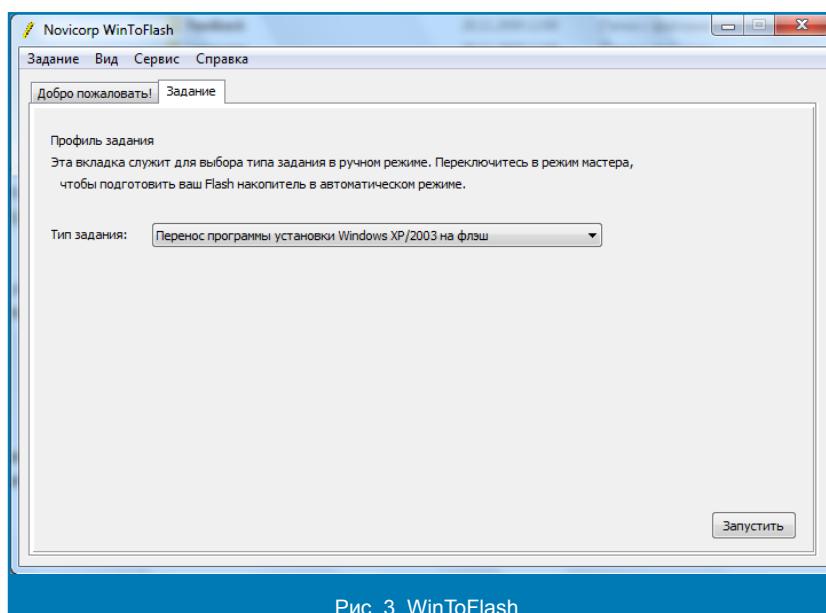
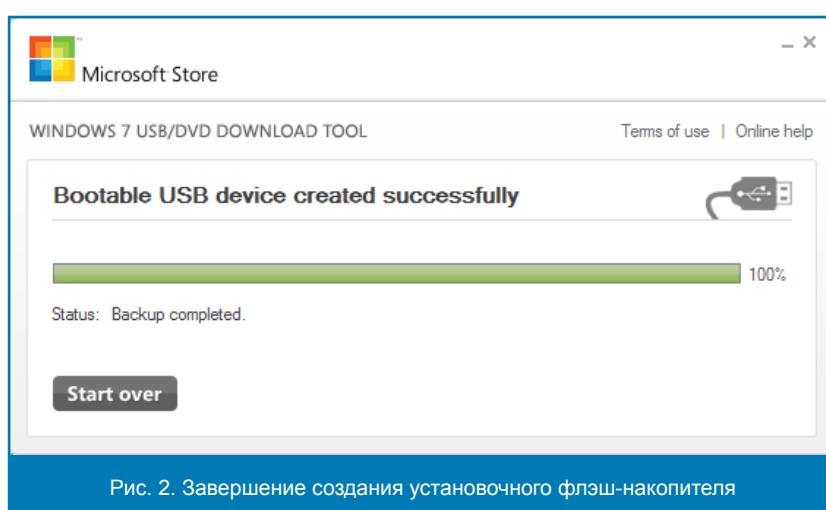
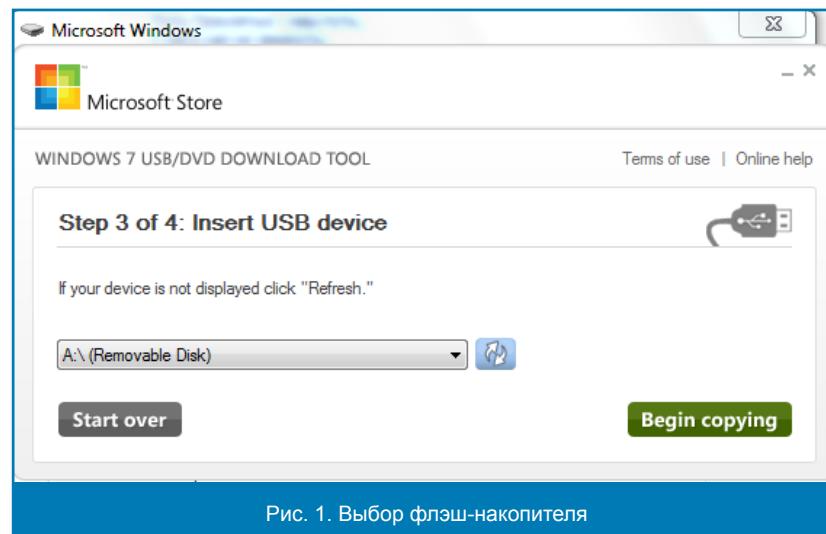
1. Запустите программу WinToFlash.
2. Откроется основное окно программы. Перейдите на вкладку **Задание** (рис. 3).
3. В раскрывающемся списке выберите пункт **Перенос программы установки Vista/2008/7 на флэш**.
4. Нажмите кнопку **Запустить**.
5. Укажите путь к установочным файлам ОС.
6. Укажите путь к флэш-накопителю, который необходимо сделать установочным.
7. Установите флагок **Снять атрибут только для чтения** с файлов (для последующего редактирования установочных файлов ОС).
8. Нажмите кнопку **Запустить**.
9. Примите условия лицензионного соглашения нажатием кнопки продолжить.
10. Нажмите кнопку **OK**. Начнется процесс форматирования, создания загрузочной записи и копирования установочных файлов.
11. По окончании копирования закройте окно программы. Установочный флэш-накопитель с Windows 7 готов к работе.

### Сравнение функциональности программ WinToFlash и Windows 7 USB DVD Download Tool

→ Программа Windows 7 USB DVD Download Tool, как видно из названия, предназначена для создания установочного флэш-накопителя Windows 7.

Однако ее можно использовать и для создания установочного флэш-накопителя Windows Server 2008R2.

→ Программа WinToFlash при использовании параметров по умолчанию меняет файловую систему на FAT32. ▶



А программа Windows 7 USB DVD Download Tool меняет файловую систему на NTFS, что дает больше преимуществ. Применять файловую систему NTFS при использовании программы WinToFlash можно, но предварительно надо отформатировать флэш-накопитель в NTFS, а затем на вкладке **Этапы подготовки** снять флагок **Форматировать диск**.

## Создание установочного флэш-накопителя вручную

Функциональность вышеуказанных программ, как видно из таблицы, оставляет желать лучшего.

 Поэтому более рациональным методом решения проблемы будет создание установочного флэш-накопителя вручную.

Начиная с Windows Vista, как уже говорилось, используется совершенно новый формат установочных файлов. В связи с этим был обновлен и загрузчик. Он поддерживает, помимо всего прочего, и загрузку с флэш-накопителей, но для этого необходимо вызвать специальную команду. Рассмотрим более подробно создание загрузочного флэш-накопителя вручную. Данный метод применим ко всем ОС, начиная с Windows Vista.

1. Во-первых, необходимо отформатировать накопитель в формате NTFS. Для этого можно воспользоваться как графической оболочкой, так и средствами командной строки. Я приведу пример, того, как пользоваться командной строкой, так как весь метод основан на командной строке, это позволяет использовать сценарии.

Код:

```
diskpart
select disk <Номер флэш-накопителя, можно определить командой list disk>
clean
create partition primary
select partition 1
active
format fs=ntfs quick
assign
exit
```

Приведенные здесь команды создают на флэш-накопителе основной раздел, форматируют его и назначают активным.

2. Далее надо скопировать все файлы с установочного компакт диска Windows. Это можно сделать через проводник или же следующей ко-

Возможности программы WinToFlash	Возможности программы Windows 7 USB DVD Download Tool
<b>Поддерживаемые ОС</b>	
Windows XP, 2003, Vista, Server 2008, Server 2008 R2, 7, а также среди PE перечисленных ОС	Windows 7 Windows Vista, Server 2008, Server 2008 R2 (не документировано)
<b>Поддерживаемые файловые системы флэш-накопителей</b>	
FAT 16 CHS, LBA; FAT 32 CHS, LBA; NTFS – только при оставлении файловой системы без изменений	Только NTFS
<b>Возможность автоматизации установки</b>	
Поддерживается при создании установочного флэш-накопителя с Windows XP\SERVER 2003	Только при ручной записи файла ответов

мандой:

```
xcopy <Путь установочному диску Windows> <Путь к флэш-накопителю> /E
```

3. Далее нужно создать загрузочную запись на флэш-накопителе:  

```
<Имя флэш-накопителя>
cd boot
bootsect.exe /nt60 <Имя флэш-накопителя>
```
4. Загрузочная запись создана. Установочный флэш-накопитель готов к работе.

**ВНИМАНИЕ** Всегда используйте загрузчик (файл bootsect.exe) только с того же диска, с которого вы копируете установочные файлы.

Если требуется создать загрузочный флэш-накопитель со средой Windows PE (поддерживается только с 2.0), то порядок действий будет немного иным.

После того, как накопитель отформатирован и на него скопированы все файлы Windows PE, необходимо вставить установочный диск той версии Windows, на основе которой была сделана данная версия PE (для Vista\Server 2008 – 2.0, для 7\Server 2008 R2 – 3.0).

После этого в командной строке перейти в каталог boot и оттуда выполнить указанную в пункте 3 команду.

## Загрузка компьютера с загрузочного флэш-накопителя

 Основным требованием к компьютеру, на который предполагается развернуть ОС при помощи установочного флэш-накопителя, является поддержка загрузки с флэш-накопителя в BIOS.

Отформатированный вышеуказанным методом флэш-накопитель, для BIOS представляет собой USB-жесткий диск. Поэтому необходимо отредактировать в BIOS параметры загрузки, а именно порядок загрузки с жестких дисков (при подключенном флэш-накопителе).

Далее компьютер следует перезагрузить, и загрузка с флэш-накопителя начинается без каких-либо действий пользователя.

После загрузки программы установки, установка осуществляется согласно стандартному сценарию. ■

# Работа с базой данных в WP7 для ленивых



## Автор

Филипп Панфилов  
ponf@inbox.ru  
<http://mne.p0x.ru>  
МСП в МГТУ им. Н.Э.Баумана  
г. Москва

Так уж получилось, что никогда раньше у меня не было необходимости использовать в моих проектах базы данных для хранения информации, так что все мои знания ограничивались парой посещений лекций по СУБД на втором или третьем курсе :[

Но во время написания приложения *HabraReader* под Windows Phone 7, мной и моим коллегой @kin9pin было принято самое логичное решение – хранить посты в БД. В интернете мы нашли довольно много статей и пошаговых инструкций по созданию и работе с БД в Windows Phone, но все это нас по тем или иным причинам не устраивало (например, из-за привязки к MVVM, до которого все никак руки не доходят).

Но выход нашелся! Причем оказался настолько простым и элегантным, что с помощью этого решения любой сможет создать и подключить БД к своему проекту за считанные минуты.

## Создаем структуру базы данных

Для начала нам понадобится база данных в формате .sdf. Как ее создать? Очень просто! Идем в Visual Studio, создаем новый проект (например, Windows Forms Application) (рис.1).

Добавляем новый элемент в наш проект типа Local Database (рис. 2).

Жмем Next и Finish. Далее идем в View и выбираем Server Explorer (Ctrl + W, L). В разделе Data Connections мы видим нашу только что добавленную базу данных. Теперь мы можем создать структуру будущей базы данных. Начнем с того, что добавим одну таблицу (рис. 3).

Для простейшего примера нам будет достаточно таблицы с 5 столбцами: **ID** (идентификатор поста, всегда уникальный, автоматически увеличивается на 1 и служит первичным ключом), **Title** (название поста, текст, не больше 150 символов), **Description** (описание поста, текст, не больше 1000 символов), **Date** (дата и время создания поста, тип datetime) и **Link** (ссылка на полную версию поста, текст, не длиннее 100 символов). В итоге выглядит это примерно как на рис. 4.

Теперь надо перестроить проект, и на этом Visual Studio можно на некоторое время закрыть и расслабиться: самая сложная часть осталась позади :) В папке с вашим проектом появился файл MyDB.sdf. Дальнейшая работа будетходить именно с ним, так что весь солюшн, кроме MyDB.sdf можно удалить. Файл MyDB.sdf я положил в D:\Tests для большего удобства в выполнении следующего шага. ▶

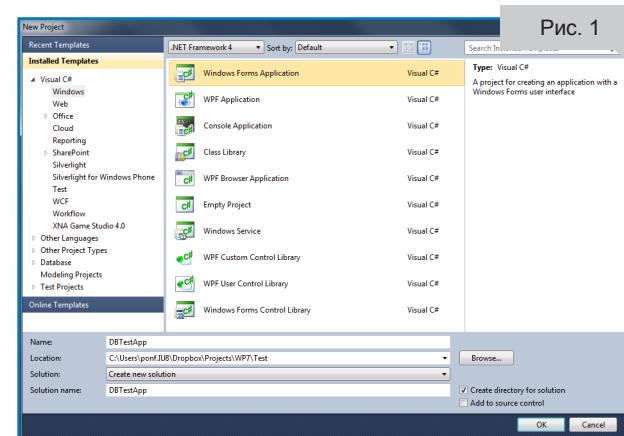


Рис. 1

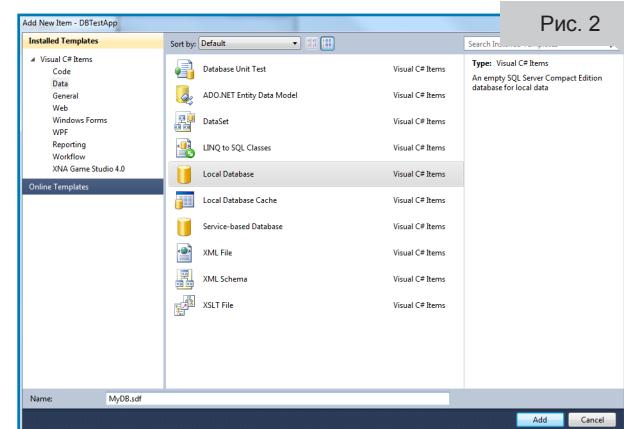


Рис. 2

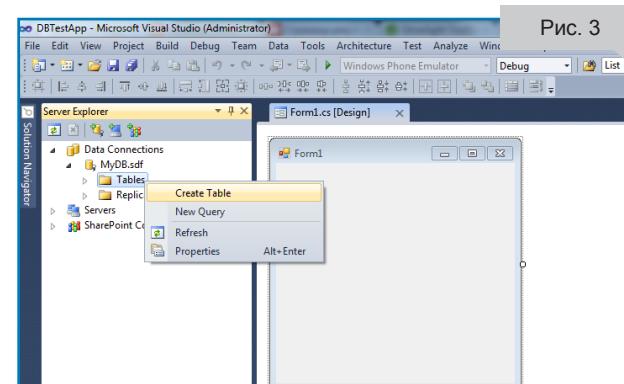


Рис. 3

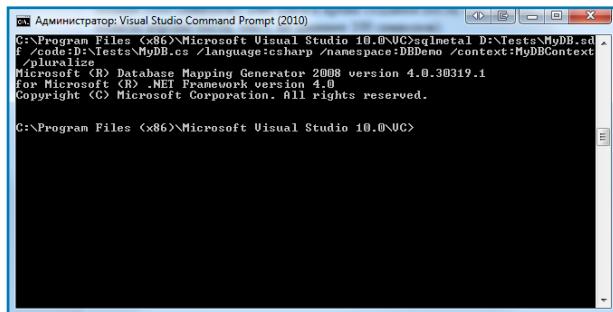
## Создаем DataContext для работы с БД

Что такое `DataContext`? Это некоторый класс, который служит нам для работы с БД и выполнения основных CRUD операций (create, read, update, delete). Для того чтобы его создать нужно открыть командную строку Visual Studio: **Пуск → Все программы → Microsoft Visual Studio 2010 → Visual Studio Tools → Visual Studio Command Prompt**.

В командной строке с помощью утилиты `sqlmetal` мы сгенерируем `DataContext` для нашей базы данных `MyDB.sdf`:

```
sqlmetal D:\Tests\MyDB.sdf /code:D:\Tests\
MyDB.cs /language:csharp /namespace:DBDemo /
context:MyDBContext /pluralize
```

В этой команде мы указали путь к нашей БД и путь, куда поместить наш `DataContext`, выбрали язык C#, указали название пространства имен и имя `DataContext`.



Теперь в нашей папке `Tests` лежит файл `MyDB.cs`, который является `DataContext` нашей БД.

## Пример использования

Давайте создадим новый проект под WP7 (обязательно укажите Windows Phone 7.1, т. к. поддержка баз данных появилась только в WP7 Mango). В проект добавляем наш `MyDB.cs`, в референсы добавляем `System.Data.Linq`. Единственное, что придется изменить в `MyDB.cs` – это namespace на пространство имен нашего проекта и удалить два следующих метода:

```
public MyDBContext(System.Data.IDbConnection connection) :
base(connection, mappingSource)
{
    OnCreated();
}

и

public MyDBContext(System.Data.IDbConnection connection,
System.Data.Linq.Mapping.MappingSource mappingSource) :
base(connection, mappingSource)
{
    OnCreated();
}
```

так как интерфейс `IDbConnection` не поддерживается операционной системой телефона.

Следующим шагом мы создадим промежуточный `Helper`-класс для большего удобства работы с базой данных. При таком подходе в `MyDB.cs` вносить изменения больше не придется (разве что если захотите внести изменения в структуру БД).

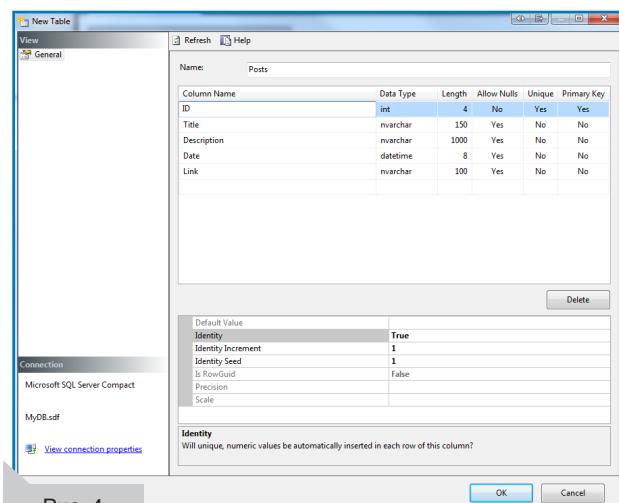


Рис. 4

```
public class DBHelper
{
    private const string ConnectionString = @`isostore:/ReaderDB.sdf`;

    public static void CreateDatabase()
    {
        using (var context = new MyDBContext(ConnectionString))
        {
            if (!context.DatabaseExists())
            {
                context.CreateDatabase();
            }
        }
    }

    public static void DeleteDatabase()
    {
        using (var context = new MyDBContext(ConnectionString))
        {
            if (context.DatabaseExists())
            {
                // delete database if it exist
                context.DeleteDatabase();
            }
        }
    }

    public static void AddPost(Post post)
    {
        using (var context = new MyDBContext(ConnectionString))
        {
            if (context.DatabaseExists())
            {
                context.Posts.InsertOnSubmit(post);
                context.SubmitChanges();
            }
        }
    }

    public static IList<Post> GetPosts()
    { ▶ }
```

```
    IList<Post> posts = new List<Post>();
    using (var context = new MyDbContext(ConnectionString))
    {
        posts = (from emp in context.Posts select emp).ToList();
    }
    return posts;
}
```

Теперь в коде можно просто вызывать статические методы класса DBHelper для различного взаимодействия с БД! Например, будем создавать БД (если она отсутствует) при загрузке страницы.

```
private void LayoutRoot_Loaded(object sender, RoutedEventArgs e)
{
    DBHelper.CreateDatabase();
}
```

## Немного о ConnectionString

В общем случае ConnectionString – некоторая строка, в которой могут быть указаны различные параметры для подключения к базе данных.

### Источник данных

Источник данных – параметр по-умолчанию, то есть вы можете не указывать свойство Data Source. База данных может располагаться как в изолированном хранилище, так и в директории с данными приложения.

```
string ConnectionString = @"isostore:/ReaderDB.sdf";
string ConnectionString = @"appdata:/ReaderDB.sdf";

string ConnectionString = @"DataSource = 'isostore:/ReaderDB.sdf'";
string ConnectionString = @"DataSource = 'appdata:/ReaderDB.sdf'";
```

### Пароль

Пароль используется чтобы зашифровать базу данных при создании и так же его необходимо указывать при подключении к ней. Максимальная длина пароля – 40 символов.

```
string ConnectionString = @"Data Source = 'iso-store:/ReaderDB.sdf'; Password = 'mypassword'";
```

### Максимальный размер БД

Размер указывается в мегабайтах, в пределах от 32 до 512.

```
string ConnectionString = @"Data Store = 'isostore:/ReaderDB.sdf'; Max Database Size = 128";
```

### Режим

Существуют четыре режима открытия БД («Read Write», «Read Only», «Exclusive» и «Shared Read»), которые обозначают вид доступа.

```
// Позволяет нескольким процессам читать и писать в
```



БД одновременно. Это режим по-умолчанию.

```
string ConnectionString = @"Data Store = 'isostore:/ReaderDB.sdf'; File Mode = 'ReadWrite';";
```

```
// Разрешает процессам производить только чтение из БД.
string ConnectionString = @"Data Store = 'isostore:/ReaderDB.sdf'; File Mode = 'Read Only';";
```

```
// Разрешает только одному процессу читать и писать в БД.
string ConnectionString = @"Data Store = 'isostore:/ReaderDB.sdf'; File Mode = 'Exclusive';";
```

```
// Разрешает всем процессам читать из БД, но только один может писать.
string ConnectionString = @"Data Store = 'isostore:/ReaderDB.sdf'; File Mode = 'Shared Read';";
```

### Максимальный размер буфера

Указывается в килобайтах, в пределах от 384 до 5120.

```
string ConnectionString = @"Data Store = 'isostore:/ReaderDB.sdf'; Max Buffer Size = 2048";
```

### Культура

Параметр культуры указывается в виде «язык-Страна», например, «ru-RU» для русского языка в России.

```
string ConnectionString = @"Data Store = 'isostore:/ReaderDB.sdf'; Culture Identifier = ru-RU";
```

### Чувствительность к регистру

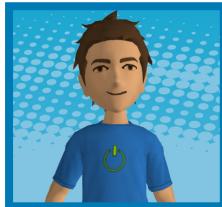
В параметрах можно так же указать, будет ли база данных чувствительна к регистру.

```
string ConnectionString = @"Data Store = 'isostore:/ReaderDB.sdf'; Case Sensitive = true";
```

## Заключение

Надеюсь, с помощью моей статьи вам удалось разобраться в работе с базами данных в Windows Phone 7. Как вы видите, создание и подключение базы – данных достаточно простые операции, и, используя генерированный с помощью sqlmetal класс, можно разобраться в структуре и основных принципах работы. После этого уже ничто не помешает вам редактировать структуру базы данных и создавать новые проекты, использующие БД, с нуля. ■

# Использование VHD в ОС Windows 7 и Windows Server 2008 R2



## Автор

Никита Львов  
MSP  
mgupi-it.ru  
г. Москва

**К**ак известно, одним из нововведений системы Windows 7/Server 2008R2 является поддержка VHD (файлов виртуального жесткого диска). Формат VHD используются в виртуальных машинах для хранения данных, он был выкуплен Microsoft у корпорации Connectix вместе с программой Virtual PC. В 2005 году его спецификация стала доступной трем лицам.

→ VHD – формат файлового образа, содержащий полную структуру физического жесткого диска.

Таким образом, виртуальный жесткий диск может быть разбит на разделы, как и физический жесткий диск. VHD используется в программах виртуализации Virtual PC, Virtual Server 2005, Hyper-V, VirtualBox, Xen, VMWare.

Теперь же операционная система поддерживает функции создания VHD, а также их подключения к системе. Однако самым главным преимуществом является возможность загрузки операционной системы из VHD.

Преимуществами VHD являются:

- ▼ Удобство хранения данных – данные одной виртуальной машины хранятся в одном файле.
- ▼ Удобство ограничения дискового пространства – теперь не нужно настраивать дисковые квоты – стоит просто изменить размер VHD.
- ▼ Экономия физического дискового пространства – можно создать динамически расширяющийся VHD, который увеличивает свой размер соразмерно с увеличением хранящейся на нем информации. То есть только что созданный расширяемый VHD почти не занимает места.
- ▼ Удобство резервного копирования – теперь для резервного копирования данных, хранящихся в VHD, вы можете просто копировать их. Естественно, что поддерживаются сценарии.

## Архитектура VHD

Виртуальные жесткие диски бывают нескольких типов:

### Фиксированного размера

VHD занимает ровно столько места, сколько указано при его создании, вне зависимости от хранящегося там объема информации. Его размер может быть изменен путем утилиты VhdResizer.

### Динамически расширяющиеся

VHD при своем создании почти не занимает места, его размер расширяется при наполнении информацией. Однако удаление информации из VHD не уменьшает его размер.

### Дифференциальные

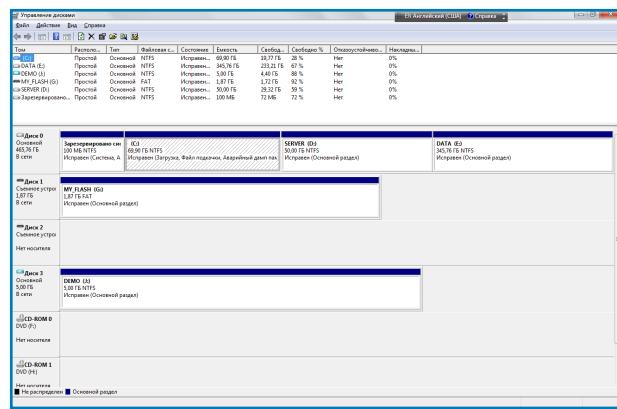
Это диски, которые связаны с уже существующим по связи «ребенок-родитель». Таким образом, дифференциальный диск хранит в себе лишь изменения по сравнению с родительским диском.

## Графические средства работы с VHD

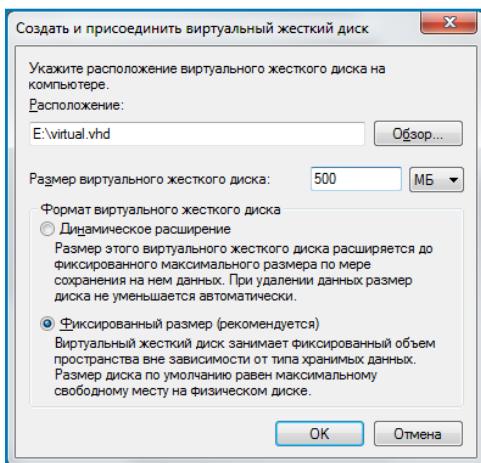
До выхода Windows 7 работа с VHD осуществлялась при помощи ПО для виртуализации: Virtual PC и Virtual Server 2005. Данное ПО позволяло подключить виртуальный диск к виртуальной машине и работать с ним из ВМ. Virtual Server 2005 имеет средства для подключения VHD к операционной системе, однако с выходом Windows 7 средства для работы с VHD значительно расширились.

→ Теперь система имеет возможность создать и подключить VHD из графического интерфейса и командной строки.

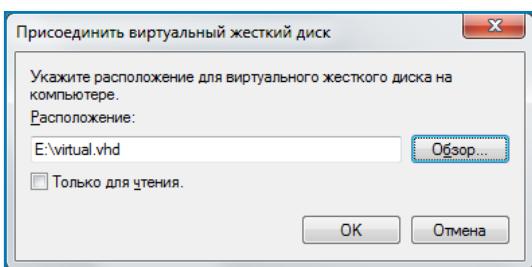
Графические средства реализованы при помощи консоли MMC diskmgmt.msc.



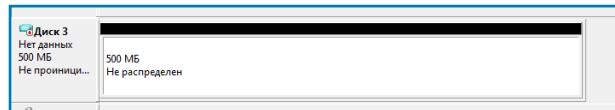
Чтобы создать виртуальный жесткий диск, необходимо вызвать команду **Создать виртуальный жесткий диск** из меню **Действие** консоли MMC. ▶



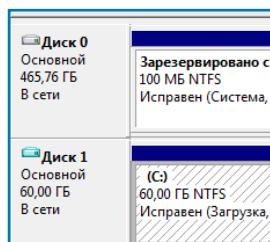
После создания жесткого диска он автоматически подключается. Если же диск был создан ранее, то его необходимо присоединить командой **Присоединить виртуальный жесткий диск**.



После присоединения, только что созданный виртуальный диск должен быть проинициализирован.



В окне diskmgmt.msc виртуальные диски отличаются от физических цветом значка диска.



## Консольные средства работы с VHD

Однако работа с VHD с использованием графических средств, не полностью функциональна. Гораздо более эффективно использовать средство командной строки diskpart.exe.

Для создания VHD из diskpart.exe существует команда

```
Create vdisk file=<путь> maximum=<Размер в МБ>
type=FIXED|EXPANDABLE SD=<строка SDDL> parent=<путь>
source=<путь>
```

- ▶ Параметр **file** указывает путь к создаваемому файлу VHD.
- ▶ Параметр **maximum** указывается для диска фиксированного размера либо для динамически расширяемого, определяя максимальный размер VHD.
- ▶ Параметр **type** указывает тип VHD – фиксированный

либо динамически расширяющийся.

▶ SD указывает строку SDDL, предназначенную для настройки безопасности. Этот параметр необязателен.

▶ Указание параметра **parent** создает дифференциальный VHD. Указывается путь к родительскому VHD. При этом нельзя указывать параметры **type** и **maximum**.

▶ Параметр **source** указывает на создание виртуального жесткого диска на основе другого VHD.

После создания VHD, для возможности работы с ним, необходимо выбрать его в программе diskpart. Для этого существует команда **Select vdisk file=<путь>**.

После того, как файл VHD выбран, с ним можно выполнить следующие действия:

1. Получить сведения о VHD,
2. Подключить VHD,
3. Уменьшить размер динамически расширяющегося VHD,
4. Выполнить слияние VHD с родительским,
5. Отсоединить VHD.

Для получения сведений о VHD, существует команда **detail vhd**. Она выводит информацию о файле виртуального жесткого диска.

Подключить VHD для работы можно путем ввода команды:

```
attach vdisk [readonly] [{sd=<строка SDDL>}|usefilesd]
```

Необязательный параметр **readonly** показывает, что подключенный VHD будет доступен только для чтения. Параметр **sd** указывает на дескриптор безопасности, используемый VHD, либо при использовании параметра **usefilesd** указывает использование дескриптора безопасности самого VHD. Как вы уже поняли, по умолчанию используется **usefilesd**.

Как говорилось выше, динамически расширяющийся VHD увеличивает свой размер соразмерно наполнению его данными. Однако при удалении данных, размер VHD не уменьшается. Так вот, для оптимизации размера HD используется команда **compact vdisk**.

Также существует специальная команда, предназначенная для слияния дочернего VHD с родительским:

```
merge vdisk depth=<число родительских дисков для слияния>
```

Подразумевается, что в данный момент выбран дифференцированный VHD.

Если работа с виртуальным жестким диском закончена, то его необходимо отсоединить при помощи команды **detach vdisk**. Помимо этого, вы можете вывести список всех подключенных VHD командой **list vdisk**.

На этом были рассмотрены стандартные средства для работы с виртуальными жесткими дисками.

## Преимущества загрузки ОС Windows 7\Server2008R2 с VHD

Как уже говорилось, Windows 7\Server2008R2 имеют в своем составе средства для работы с виртуальными жесткими дисками. Таким образом, администратор имеет возможность создавать VHD для хранения пользовательских данных – архивация и восстановление данных совершались бы при помощи подобного bat-файла:

```
Net use Z: \\server\share {для использования в качестве хранилища архива сетевого размещения}
del Z:\<computername>.vhd ▶
```

# Администрирование

```
copy C:\<computername>_data.vhd Z:\  
echo Архивация завершена!
```

Согласитесь, довольно удобно, не так ли? Но рано или поздно встает вопрос о использовании VHD в качестве хранилища системного диска. Благодаря преимуществам, предоставляемым VHD, в разы повышается легкость архивации и восстановления.

Раньше для архивации образа системы с диска С: вам приходилось выполнять следующие операции:

1. Загрузиться с Windows PE с включенным инструментом архивации (imagex).

2. Выполнить команды:

```
net use M: \\server\share {для использования в  
качестве хранилища архива сетевого размещения}  
imagex /capture C: M:\<computername>_system.wim  
"BACKUP"  
wpeutil reboot
```

Из собственного опыта применения подобного метода архивации скажу, что архивация таким методом занимает примерно час. Притом пользователь не может работать с ОС в данный момент.

Если хранить системный диск в VHD, то это дает следующие преимущества:

1. Для каждой ОС не требуется отдельный раздел, например на компьютере может быть установлено 10 ОС в 10 VHD, которые будут храниться на одном физическом диске С:

2. Идеально для тестирования. Например, вам необходимо выполнить тестирование в физической среде какого-либо приложения. Для этого установите операционную систему в VHD и выполните тестирование. При этом основная ОС не будет затронута.

3. Возможна архивация VHD «на ходу».

4. Легко выполнить перенос системы в виртуальную среду.

Также стоит отметить, что возможность загрузки с VHD поддерживается загрузчиком bootmgr той же версии, что и в Windows 7\Server 2008R2, а также операционными системами, не ниже вышеуказанных версий.

## Установка Windows 7 в VHD

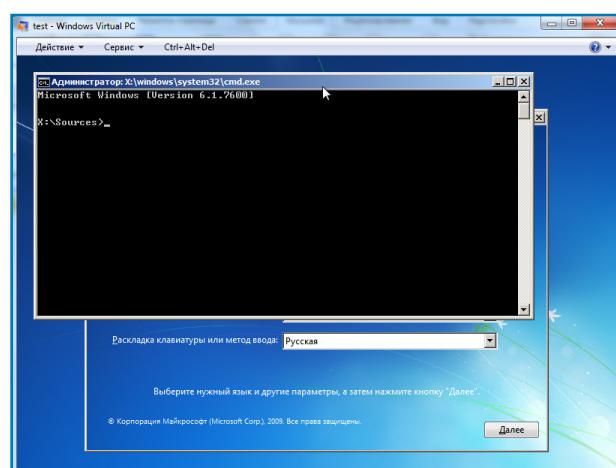
→ Если установить систему на виртуальный жесткий диск, это будет выглядеть следующим образом. Физический жесткий диск будет содержать один (или несколько) разделов. При этом на одном из них будет располагаться загрузчик и файл виртуального жесткого диска, на котором хранится системный раздел.

Таким образом, чтобы заархивировать текущее состояние ОС, достаточно лишь скопировать файл VHD. Для повышения быстродействия операционной системы рекомендуется использовать виртуальные жесткие диски фиксированного размера.

Для того, чтобы установить ОС Windows 7\Server2008R2 на виртуальный жесткий диск, необходимо выполнить следующие операции:

→ Загрузиться с установочного диска системы.

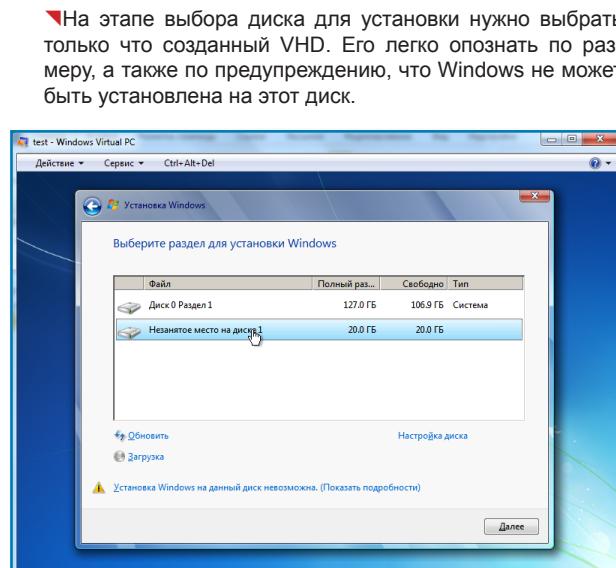
→ Нажать комбинацию клавиш **Shift+F10**, чтобы открыть командную строку:



▼ В командной строке выполнить следующие команды (подразумевается наличие жесткого достаточного размера, без разделов):

```
diskpart  
select disk 0 {Выбираем физический диск}  
create partition primary  
format fs=ntfs quick  
active  
assign letter=C {Создаем основной раздел и соответствующим образом готовим его к работе}  
create vdisk file="C:\os.vhd" type=fixed  
maximum=20480 {Создаем VHD фиксированного размера в 20ГБ.}  
select vdisk file="C:\os.vhd" {Выбираем только что созданный VHD}  
attach vdisk {Подключаем VHD}  
exit  
exit
```

▼ Далее выполняется работа с программой установки: выбор языка, принятие лицензионного соглашения, выбор редакции ОС для установки.



▼ Проигнорируйте это предупреждение, нажав кнопку **Далее**. Начнется установка ОС.

▼ После перезагрузки и завершения работы с ООВЕ, система готова к работе. В окне **Мой компьютер** будут отображаться два жестких диска – один из них физический, а другой – виртуальный. Сделать резервную копию ОС можно очень легко – для этого нужно всего лишь скопировать файл os.vhd. ■

# 1С:Net:Предприятие

## многопоточный http-сервер с обработкой запросов в коде 1С



### Автор

Сергей Карташев

<http://www.1centerprise.com/dotnet/>  
[support@1centerprise.com](mailto:support@1centerprise.com)

В сентябрьском номере журнала MSDeveloper.ru была опубликована обзорная статья, посвященная технологии 1С:Предприятие и использованию .Net Framework внутри 1С:Предприятие. Данная статья – продолжение, описывающее практическое применение технологии при решении разного рода задач. Метод изначально был разработан для подключения медицинского лабораторного оборудования к 1С:Предприятие 8.2, но может быть использован для подключения любого другого оборудования и организации обменов данными между базами 1С и другими системами. Статья сопровождается готовыми примерами, доступными для работы, расширения и экспериментов.

В процессе автоматизации деятельности предприятия при помощи 1С: Предприятие часто возникают задачи интеграции и обмена с оборудованием и другими сторонними информационными системами, например, банками, веб-сайтами, информационными системами партнеров. Традиционно 1С:Предприятие выступает в качестве потребителя услуг, и реже – в качестве поставщика. До этого момента у разработчиков популярными технологиями при получении информации от 1С были COM и веб-сервисы, которые появились только в версии 8.1.

У обеих технологий (COM и веб-сервисы) есть свои минусы. Главные недостатки заключаются в следующем. COM-технология позволяет в каждый момент времени выполнять только один запрос. Обращение возможно только внутри своей локальной сети. Обработка одновременных запросов для COM возможна, но потребует затрат на организацию пула соединений и отладку многопоточного приложения. Веб-сервисы же сложны по настройке и негибкие для программирования: жестко привязаны к SOAP-стандартам. Поключение устройств, обменивающихся простыми пакетами, работающих по http-прото-

колу, невозможно.

Новая идея организации http-сервера внутри 1С, предложенная в статье, опирается на проверенную временем богатую функциональность .Net Framework. Решение на основе идеи лишено недостатков COM и веб-сервисов 1С. По сравнению с COM, http-сервер можно использовать вне локальной сети и поддерживается одновременная обработка нескольких запросов. По сравнению с веб-сервисами, 1С решение на базе http-сервера обладает большей гибкостью, так как программист волен сам выбирать формат ответа сервера (в том числе HTML, JSON, графические изображения, RSS и т. д.), а также контролировать при ответе URL-адрес, идентификацию пользователей, коды ошибок, куки, кодировку и осуществлять кеширование. Настройка же http-сервера внутри 1С сводится к простому запуску внешней обработки.

### Описание примера

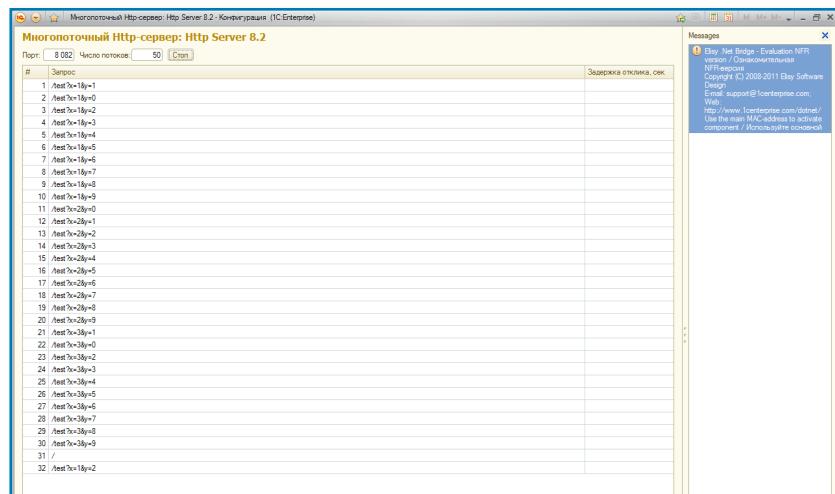
Пример, приложенный к статье, состоит из двух файлов: http-сервера (внешняя обработка 1С 8.2 HttpServer82) и

тестового приложения эмуляции одновременных запросов к серверу (внешняя обработка 1С 8.2 TestHttpServer82). Обе обработки выполнены на основе управляемых форм. По умолчанию оба приложения настроены на работу с портом 8082.

Сервер и тестовое приложение разработаны на 1С:Предприятие 8.2 и используют .Net Framework 4.0 и компонент Elisy .Net Bridge 4. Соответственно, для работы примера требуется установленный .Net Framework 4.0 и версия библиотеки Elisy .Net Bridge v.4.0.2.0 и выше. Elisy .Net Bridge позволяет гармонично использовать классы и технологии .Net Framework на 1С, ведущую роль оставляя 1С.

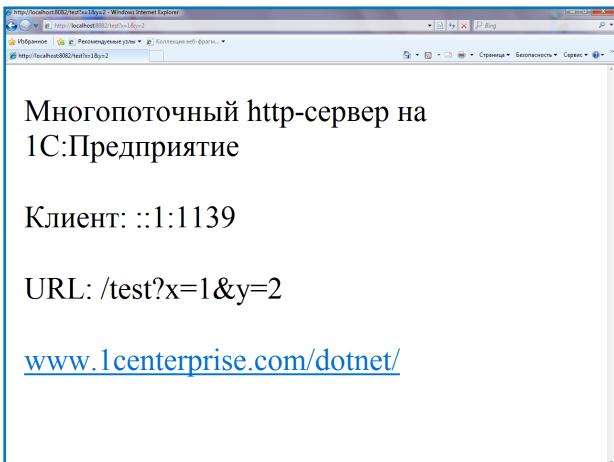
Для проверки работы достаточно запустить внешнюю обработку HttpServer82.elf из 1С:Предприятие 8.2. Если у вас Windows с включенным UAC, то запустить 1С:Предприятие необходимо под администратором, иначе у приложения не будет достаточно прав на прослушивание запросов.

Внешняя обработка позволяет задать порт, по которому будет осуществляться прослушивание и число создаваемых



мых потоков для обработки одновременных запросов. По умолчанию установлен порт 8082 и 50 потоков.

После нажатия на кнопку **Старт** сервер переходит в рабочее состояние и осуществляет обработку запросов на заданный порт. Например, теперь можно обратиться из вашего браузера по адресу <http://localhost:8082> и открыть страницу, которую вернет сервер. В запросе же можно передавать параметры, например, так: <http://localhost:8082/test?x=1>.



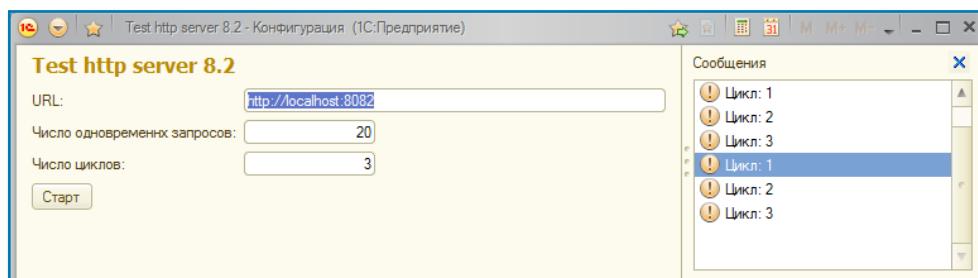
Для проверки сервера в многопоточном режиме придумана внешняя обработка TestHttpServer82.epf, которая осуществляет одновременный запуск запросов в цикле. В основу обработки для организации параллельной работы положена замечательная технология PLINQ (Parallel LINQ) из .Net framework 4.

Запускать тестовую обработку TestHttpServer82.epf следует из отдельного от сервера сеанса 1С:Предприятие, иначе одновременный запуск в одном сеансе двух обработок приведет к зависанию. В качестве параметров тестового примера выступают адрес запроса, число одновременных запросов и число циклов. По умолчанию запуск алгоритма приведет к трем циклам обращений по адресу <http://localhost:8082> с 20 одновременными запросами (надо отметить, что число одновременных запросов ограничено числом ядер процессора).

## Принцип работы

.Net framework предлагает своим разработчикам класс `HttpListener`, отвечающий за прослушивание http-протокола. Используя `HttpListener`, вы можете создать прослушивание http-трафика, которое отвечает на http-запросы. Вы можете использовать этот класс только на операционных системах Windows XP SP2 или Windows Server 2003 и выше. Попытка использования класса на более ранних системах вызовет исключение.

Ниже приведен пример кода для 1С, который инициализирует объект типа `HttpListener`, настраивая его на прослушивание всех URL по порту 8082. При запуске в 1С работа программы приостанавливается, пока не последует запрос на порт, например, из браузера. Как только вы пошлете из браузера запрос, например, <http://127.0.0.1:8082/>, 1С вернет определенную в программе html-строку.



За что ответственен каждый участок кода, можно понять из комментариев, приведенных внутри исходного кода.

```
ПодключитьВнешнююКомпоненту(«Elisy.NetBridge4»);
AddIn = New(«AddIn.ElisyNetBridge4»);
net = AddIn.GetNet();
```

```
Если НЕ net.GetStatic(«System.Net.HttpListener», »IsSupported») Тогда
```

```
    Сообщить(«Для класса HttpListener нужна ОС Windows XP SP2/2003 и выше»);
```

```
    Возврат;
```

```
КонецЕсли;
```

```
listener = net.New(«System.Net.HttpListener»);
listener.Prefixes.Add(«http://*:8082/»);
listener.Start();
```

```
Сообщить(«Прослушивание...»);
```

```
//Метод GetContext блокирует выполнение программы пока ждет запрос.
```

```
context = listener.GetContext();
```

```
request = context.Request;
```

```
// Получить объект ответа
```

```
response = context.Response;
```

```
// Создать ответ - HTML-строку
```

```
responseString = «Ответ от HttpListener»;
buffer = net.GetStatic(«System.Text.Encoding», «UTF8»).GetBytes(responseString);
```

```
// Получить поток ответа и записать ответ в него.
```

```
response.ContentLength64 = buffer.Length;
```

```
output = response.OutputStream;
```

```
output.Write(buffer, 0, buffer.Length);
```

```
// Необходимо закрыть выходной поток и остановить прослушивание
```

```
output.Close();
```

```
listener.Stop();
```

Это самый простой пример. Главный смысл примера в том, что 1С превращается в полнофункциональный http-сервер, а 1С-программист получает инструмент для гибкой настройки ответа с сервера. Он может вернуть его в любом формате (html, рисунок или JSON), затребовать идентификацию пользователя или вообще вернуть ошибку. Но вместе с тем есть недостатки: блокирование всего приложения и обработка только одного запроса. Пример к статье содержит более сложный код, за счет чего устранены недостатки простейшего примера из листинга.

## От простого к сложному

В готовый пример вошел видоизмененный код работы с `HttpListener`, в котором вызов метода `listener.GetContext()` заменен вызовом асинхронных аналогов `listener.BeginGetContext()` и `listener.EndGetContext()`. Кроме этого создаются N отдельных потоков и заложена синхронизация между потоками с вызовом кода-обработки запроса на стороне 1С:Предприятие. ►

Достоинством предлагаемой реализации http-сервера является возможность одновременной обработки N-запросов в разных потоках с передачей логики обработки в метод формы 1С:Предприятие. Пример при каждом запросе возвращает html с описанием источника и URL запроса. Решение очень гибкое, так как все доработки можно делать прямо из конфигуратора 1С. Никакие дополнительные проекты (например, C# или VB.Net) не задействованы.

Пример условно можно разделить на две части: код, который выполняется на стороне 1С:Предприятие и код, который выполняется на стороне .Net framework. При этом .Net framework взял на себя все то, что нельзя реализовать средствами 1С:Предприятие, например, создание и синхронизацию потоков.

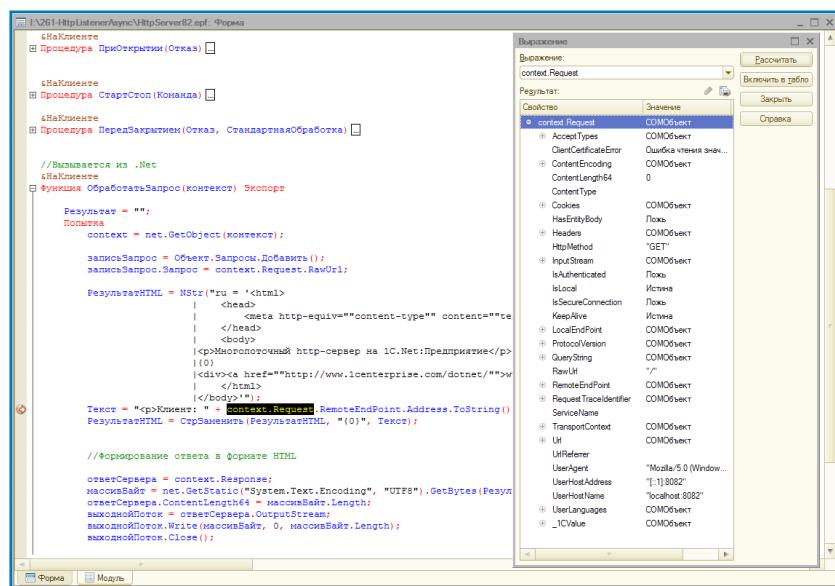
Http-сервер оформлен в виде управляемой формы. При нажатии на кнопку **Старт** происходит создание объектов классов HttpServer и Helper. Оба класса описываются на C# в макете обработки ИсходныйКод и компилируются «на лету» в обработчике ПриОткрытии формы. Класс Helper отвечает за перенаправление .Net-события в функцию ОбработатьЗапрос на форме 1С и формирование сообщение об ошибке.

Опустив не относящиеся к теме задачи по инициализации формы и вспомогательных классов, следует остановиться на методе ОбработатьЗапрос, который отвечает за возврат результата http-клиенту. Вызывается обработчик из класса Helper, в качестве параметра принимает объект контекста, и может возвратить сообщение об ошибке, которое будет возвращено клиенту.

Объект контекста типа [HttpListenerContext](#), передаваемый в метод ОбработатьЗапрос содержит два важных свойства: [Request](#) и [Response](#), отвечающие за информацию о запросе и ответе соответственно.

Свойство Request позволяет получить параметры и содержимое, переданные клиентом при запросе. Информация о запросе, помещенная в свойство Request содержит такие основные свойства как:

■ [AcceptTypes](#) – MIME-типы, поддерживаемые клиентом,



■ [ContentEncoding](#) – информация о кодировке при ответе,

■ [Headers](#) – набор заголовков,

■ [HttpMethod](#) – метод HTTP, определенный клиентом,

■ [InputStream](#) – поток, содержащий данные тела, приведенные от клиента,

■ [IsAuthenticated](#) – булево значение, показывающее, что пользователь идентифицирован,

■ [IsLocal](#) – значение, показывающее локальный ли запрос (через localhost),

■ [QueryString](#) – строка запроса из запроса,

■ [RawUrl](#) – информация об URL без хоста и порта,

■ [UrlReferrer](#) – URL ресурса-источник данного перехода,

■ [UserAgent](#) – информация о агенте-браузере пользователя.

Через свойство Response осуществляется возврат контента клиенту, передача информации об ошибке или перенаправление. Сопровождаются все может выставлением необходимых заголовков. Многие свойства ответа (Response) схожи со свойствами запроса. Значимые среди них следующие:

■ [ContentEncoding](#) – информация о кодировке при ответе,

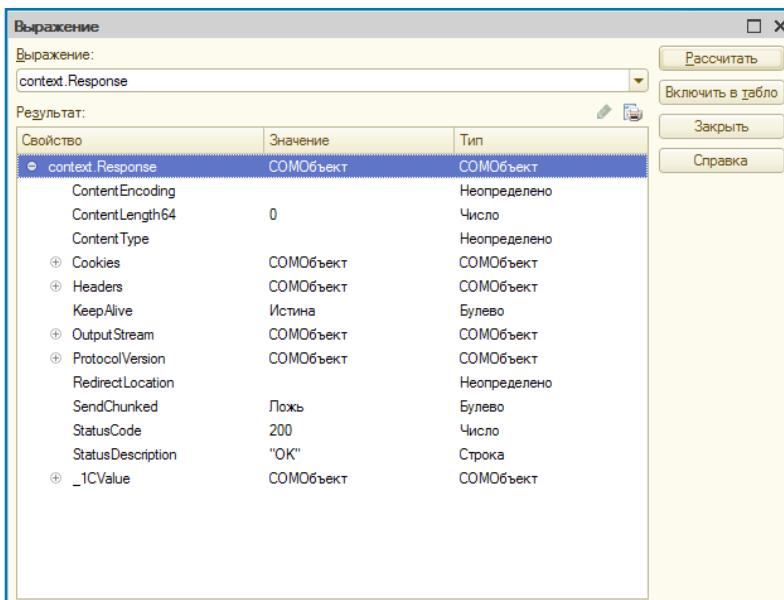
■ [Headers](#) – набор заголовков,

■ [OutputStream](#) – поток, в который будет записан ответ (например, текст Html, XML или байтовый массив изображения),

■ [RedirectLocation](#) – свойство отвечает за HTTP-заголовок Location и позволяет перенаправить вызов,

■ [StatusCode](#) – код статуса при возврате клиенту, например: 200 (OK), 404 (ресурс не найден),

■ [StatusDescription](#) – описание статуса при возврате клиенту.



Следующий код метода ОбработатьЗапрос преобразует сформированную строку РезультатHTML с HTML-кодом в набор байт и ►

записывает этот набор байт в выходной поток, который будет возвращен клиенту. Ссылка на выходной поток получена через параметр метода.

```
ответСервера = context.Response;
массивБайт = net.GetStatic(`System.Text.Encoding`,
`UTF8`).GetBytes(РезультатHTML);
ответСервера.ContentLength64 = массивБайт.Length;
выходнойПоток = ответСервера.OutputStream;
выходнойПоток.Write(массивБайт, 0, массивБайт.Length);
выходнойПоток.Close();
```

Основой обработки является класс `HttpServer`, который создает объект `HttpListener` и нужное число потоков для обработки. При вызове метода `Start` происходит запуск всех потоков-обработчиков и отдельного потока для работы `HttpListener`. Благодаря этому, можно продолжать работать с 1С во время работы http-сервера. При поступлении запроса, `HttpListener` помещает запрос в очередь, где каждый запрос последовательно обрабатывает первый свободный поток. При обработке потока срабатывает цепочка вызовов: событие `HttpServer.ProcessRequest`, обработчик события `Helper.HttpServer_ProcessRequest` и 1С-функция `Форма.ОбработатьЗапрос`. Код C# класса `HttpServer` при запуске 1С из макета ИсходныйКод компилируется «на лету».

```
_listener.AuthenticationSchemes = authenticationScheme;
_listener.Prefixes.Add(String.Format(@`http://+:{0}/`, port));
_listener.Start();
_listenerThread.Start();

for (int i = 0; i < _workers.Length; i++)
{
    _workers[i] = new Thread(Worker);
    _workers[i].Start();
}
```

Начало получения запросов происходит в методе `HandleRequests` до тех пор, пока обработка не прекратится пользователем. При поступлении запроса, запрос передается в метод `ContextReady` и работа процесса прослушивания продолжается.

```
private void HandleRequests()
{
    while (_listener.IsListening)
    {
        var context = _listener.BeginGetContext(ContextReady, null);

        if (0 == WaitHandle.WaitAny(new[] { _stop, context.AsyncWaitHandle }))
            return;
    }
}
```

Метод `ContextReady` ставит в очередь `_queue` контекст с информацией о запросе. Очередь запросов обрабатывается в методе `Worker` каждым потоком в отдельности. Метод `Worker` извлекает контекст из очереди и вызывает событие `ProcessRequest`.

```
private void Worker()
{
    WaitHandle[] wait = new[] { _ready, _stop };
    while (0 == WaitHandle.WaitAny(wait))
    {
        HttpListenerContext context;
        lock (_queue)
        {
            if (_queue.Count > 0)
                context = _queue.Dequeue();
            else
            {
                _ready.Reset();
                continue;
            }
        }
        ProcessRequest(context);
    }
}
```

```
    }

    try { ProcessRequest(context); }
    catch (Exception e) { Console.Error.WriteLine(e); }
}
}
```

## Заключение

Многопоточный сервер, описанный в статье, по простоте настройки и функциональности превосходит традиционные методы доступа извне к информационным базам 1С:Предприятие 8.х. Это простой по запуску метод, позволяющий одновременно обрабатывать несколько запросов. При этом приложение 1С не блокируется, и пользователь может продолжать работу после запуска сервера.

Основное преимущество предложенного подхода – полная подконтрольность программисту процесса, от получения запроса до формирования ответа. Например, на этапе получения запроса может быть выполнен парсинг URL, получена информация о том, как себя идентифицировал пользователь, а также полная информация о клиенте (поддерживаемые языки, записанные куки, заголовки, метод доступа). Ответ же можно вернуть практически любой, начиная от ошибки 404 Not found, заканчивая разными графическими форматами, форматами Word, Excel и популярными форматами на основе XML (JSON, HTML, RSS).

Пример, приложенный к статье, спроектирован так, что его функциональность можно легко расширить. Например, для организации кеша применить `System.Web.Caching.Cache` класс из .Net framework. А при парсинге URL попробовать поработать с классом `RouteCollection` из Asp.Net MVC. При создании RSS-ленты вам поможет класс `System.ServiceModel.Syndication.SyndicationFeed`, а при Json-сериализации обратите внимание на класс `System.Runtime.Serialization.Json.DataContractJsonSerializer`.

Конкретно для предложенного подхода на данный момент не выявлено явных недостатков. Есть вероятность того, что 1С:Предприятие в силу своих ограничений не сможет обеспечить на своей стороне должного распараллеливания и значительного увеличения производительности. Тем не менее опыты проведенные ранее для 1С:Предприятие 8.2 показали, что в схожем применении 1С достигается увеличение производительности, и 1С работает при этом стабильно.

Необходимо обратить внимание разработчиков еще на несколько моментов. Любая публикация информации в Интернете связана с риском взлома вне зависимости от способа публикации. Но в предложенном способе благодаря гибкости есть больше возможностей противостоять угрозам извне. Например, в запросе теперь доступен IP-адрес клиента, который можно блокировать по каким-то правилам (осуществляя поиск в черном списке локально или в специализированных сервисах). Параметры запроса доступны в виде строк и их можно анализировать на стороне 1С или .Net framework и блокировать опасное содержимое. Кроме этого есть несколько специализированных .Net-библиотек, доступных для решения данной проблемы, которые можно привлечь в 1С, например AntiXSS.

Второй момент заключается в том, что известные методы (в том числе и этот) предоставления услуг со стороны 1С не предназначены для массовых обращений и будут всегда уступать с этой точки зрения профессиональным серверам, например, IIS. При решении таких задач для массового доступа к информационным базам 1С рекомендуется обратить внимание на технологию прямого доступа к данным Elisy Open InfoBase. Более подробно о ней можно узнать в октябрьском выпуске журнала MSDDeveloper.ru, ознакомившись со статьей «Elisy Open Infobase: необычно-экономное использование 1С:Предприятие 8 на Asp.Net-хостинге». ■

# Зачем и как можно создать свой персональный блог на сайте журнала MSDeveloper.RU



## Автор

Станислав Горнаков  
@Gornakov  
gornakov@hotmail.com  
www.gornakov.com

**Л**юбой пользователь сети Интернет может абсолютно бесплатно и без каких-либо ограничений создать и вести свой персональный блог на сайте журнала MSDeveloper.RU. Единственное условие, которое мы ставим перед нашими читателями – это причастность или отношение к продуктам и технологиям компании Microsoft.

Журнал MSDeveloper.RU – это тематический и узконаправленный проект, рассчитанный в первую очередь на разработчиков и гиков технологий Microsoft. Очевидно, что записи, не имеющие отношений к тематике портала, не будут интересны читателям ресурса.

## Что нужно, чтобы создать персональный блог?

Ничего особенного от вас не требуется и лишь пара минут свободного времени отделяет вас от регистрации на сайте и заведения своего персонального блога. Все что вам нужно – это зарегистрироваться на сайте и создать блог, выбирая поочередно соответствующие ссылки в шапке сайта.

После создания блога, пройдитесь в профиль созданного вами пользователя и заполните информационные поля, описывающие вас, или по крайней мере поменяйте свой аватар (приветствуются личные фотографии). После чего можете смело писать свою первую запись в блоге.

## Где транслируются персональные записи?

Записи из блогов всех пользователей портала и компаний, автоматически транслируются в основную ленту сайта, попадая на главную страницу ресурса, а также во все социальные сети и RSS-канал журнала. Нет ни каких

ограничений на трансляцию записей на главную страницу – у нас отсутствует рейтинг записей, по которому идет отдельная выборка.

Также все персональные записи доступны в отдельном разделе **Записи → Персональные записи**, а перечень всех блогов наших читателей можно найти в разделе **Блоги → Персональные блоги**. Кроме всего прочего, мы внимательно читаем все записи читателей и наиболее интересные материалы могут быть напечатаны в журнале, естественно с вашего согласия.

## Автоматическая RSS-трансляция в наш блог

Наш ресурс поддерживает автоматическую RSS-трансляцию записей из вашего блога или сайта в наш блог журнала. Опции по настройке RSS-транс-

ляции доступны как на этапе создания блога, так и в настройках самого блога. У нас нет ограничений на эту возможность, но мы следим за качеством и оформлением постов. Если записи буду некорректно оформлены или не соответствовать тематике портала, то RSS-трансляция будет приостановлена.

## Рейтинг и карма

У нас отсутствует рейтинг записей, мы имеем только рейтинг пользователей, который влияет лишь на выдачу списка пользователей в общем рейтинге в специальном разделе. Записи не ранжируются и не учитываются в рейтинге, а соответственно просить в Twitter «поддержите плиз» нет необходимости, запись автоматом попадет на главную страницу сайта. ►

The screenshot shows the MSDeveloper.RU website interface. At the top, there's a navigation bar with links like 'Лента', 'Записи', 'Блоги', 'Читатели', 'Журнал', 'События', 'Фотогалерея', 'Форум', 'Подписка', and 'Поиск'. Below the navigation, there's a banner for 'ГЛАВНЫЕ ОСЕННИЕ МЕРОПРИЯТИЯ ИНДУСТРИИ ХОСТИНГА' and a sidebar for 'Новинки на сайте' and 'Новые блоги'. The main content area displays a list of personal blogs with their authors, titles, and creation dates. At the bottom right, there's a link to 'facebook'.

## О СТАТИСТИКЕ САЙТА

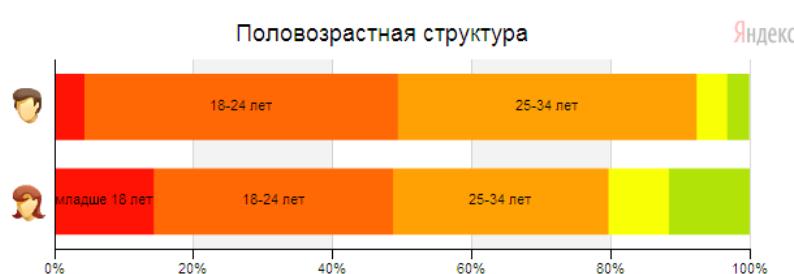
На данный момент проекту всего полтора месяца и при этом он уже пользуется огромной популярностью у читателей. Так на сегодняшний день ежесуточно сайт посещают более тысячи человек. За статистикой посещений можно наблюдать по счетчику Яндекс Метрики, который расположен внизу каждой страницы сайта. Клик на счетчике откроет статистику за неделю.



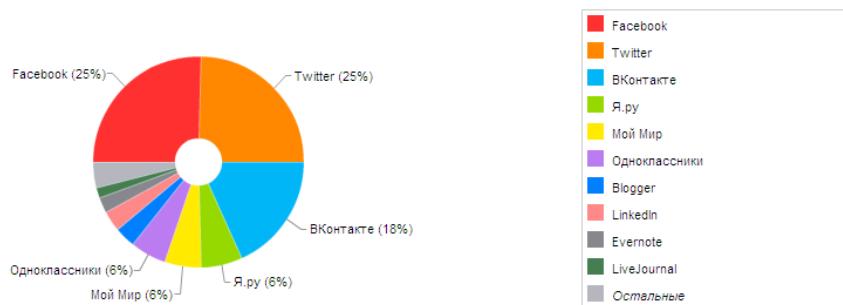
Охват аудитории пользователей еще более интересен. Нас читают от Редмонда США до Курильских островов. При этом хороший трафик идет из Редмонда, видимо от наших соотечественников, работающих в Microsoft. Друзья, спасибо, мы с Вами!



Портрет половозрастной структуры посетителей указывает на то, что ресурс интересен в первую очередь людям в возрасте от 18 до 34 лет.



Большой трафик идет из наших социальных сетей Twitter, Facebook и ВКонтакте, в которые мы транслируем все записи из блогов сайта. Кроме этого многие пользователи и сами делятся ссылками на записи, используя для этих целей кнопки **Поделиться**, которые есть под детальным текстом каждой из записей. ■





# MSD журнал eveloper.RU

## Сайт журнала MSDeveloper.RU

-  Каждый день свежие новости
-  Каждый месяц новый номер журнала
-  Блоги пользователей
-  Система личных сообщений
-  Большой тематический форум
-  Календарь важных событий
-  Стена друзей
-  Подборка книг
-  Горячие вакансии
-  Бесплатная подписка на журнал