

Введение

Платформа .NET пронеслась по миру Windows-разработчиков, как торнадо, мгновенно сделав привычные нам способы написания программ устаревшими. В ней есть практически все, что нужно для Web-приложений и служб, а также для бизнес-приложений, для которых раньше мы использовали Visual Basic и MFC.

Однако существуют упрямые разработчики, которые перейдут на новую платформу и забросят свои прежние наработки только под угрозой расстрела. Это люди с многолетним опытом программирования на чистом C++, недоверчиво относящиеся к новомодным ключам компиляторов, превращающим их код в “управляемый”. Они не станут переносить свой код на новую платформу, независимо от того, выгодно это или нет. Отчасти эта книга предназначена именно для них, если их можно убедить перенести проекты с ATL 3/Visual Studio 6 на ATL 8/Visual Studio 2005.

На других разработчиков, обитающих в “трущобах” Windows, торнадо не оказывают воздействия — они вообще не замечают этих торнадо. Эти разработчики создают приложения, которые должны работать на любых системах, начиная с Windows 95, со слишком старыми процессорами и малыми объемами памяти для .NET. Они не могут позволить себе роскошь тратить драгоценные ресурсы на сбор мусора, JIT-компиляцию или на огромные библиотеки, набитые ненужными им классами. Для них важнее всего скорость загрузки программ, их быстрдействие и способность работать в любых условиях. Чтобы они стали использовать какую-то платформу, она должна быть свободна от накладных расходов на ненужные им возможности, обладать гибкостью и высокой производительностью. Именно для таких разработчиков и предназначена ATL 8, новейшая платформа для разработки Windows-приложений.

Клиентам ATL предоставляет оконные классы, интеллектуальные типы COM, поддержку элементов управления COM и размещения этих элементов, интеграцию с MFC (включая поддержку части классов MFC без необходимости использовать остальную часть MFC) и генерацию интерфейсов Web-служб. Серверам ATL предлагает поддержку COM-серверов и объектов и механизмы для реализации высокопроизводительных многопоточных Web-служб и приложений. И в клиентском, и в серверном коде в ATL широко используются макросы и шаблоны, обеспечивающие максимум гибкости при минимуме накладных расходов. Это гарантирует, что в приложения не будут включены неиспользуемые возможности, а разработчики смогут легко понять, как классы их приложений связываются с функциями платформы. Кроме того, ATL снабжена обширным набором мастеров для создания разнообразных проектов.

Атрибуты

Как попытка повысить продуктивность труда разработчиков в ATL 7 и Visual Studio 2003 в ATL появились атрибуты, которые позволяли добавлять в исходный код элементы метаданных, обычно использовавшиеся в файлах IDL и дополнитель-

ных ресурсах (например, атрибут `uuid`). Собственно говоря, все были в таком восторге от появления атрибутов, что в Visual Studio 2003 все мастера по умолчанию генерировали код с атрибутами. Однако в том, что касается атрибутов, отнюдь не все так безоблачно. В .NET и IDL атрибуты являются частью модели программирования; они поддерживаются повсеместно. В ATL атрибуты представляют собой просто надстройку над компилятором, вроде супермакросов, которая генерирует базовые классы, таблицы макросов, реестровые скрипты и файлы IDL.

Однако в отличие от макросов атрибуты в ATL не являются прозрачными — вы не можете ясно увидеть, какой код будет сгенерирован для каждого атрибута. У компилятора есть ключ, который должен позволять просмотреть этот код, но в Visual Studio 2005 этот ключ работает с ошибками. Соответственно, появились проблемы с отладкой программ, в которых использовались атрибуты. Я не хочу сказать, что в остальном в ATL (или вообще в какой-то программе) ошибок нет, но если ошибки находятся в базовых классах или макросах, их можно было обойти. Собственно говоря, солидная часть кода в первом издании этой книги как раз и была предназначена для обхода ошибок.

Определенные признаки в Visual Studio 2005 показывают, что атрибуты, похоже, перестали считаться будущим ATL. Например, ключи компилятора, связанные с атрибутами, выдают меньше информации. Однако самое важное знамение — то, что мастера в Visual Studio 2005 больше не генерируют код с атрибутами по умолчанию¹. Поэтому в этом издании мы не будем использовать атрибуты во всех главах, хотя им и посвящено приложение Г, “Атрибуты в ATL”. По моему скромному мнению, вам вряд ли понравится применять в своих проектах атрибуты, если вы не можете точно выяснить, что они делают.

Кому адресована эта книга

Она адресована программистам, использующим C++/COM и собирающимся перейти на ATL 8, работая в Visual Studio 2005. ATL основывается на ряде предположений, и чтобы эффективно ее использовать, вы должны понимать не только, как устроена ATL, но и почему она устроена именно так, а не иначе. Разумеется, чтобы понять это, вам нужно понять, в какой среде ATL создавалась: COM. Вместо того чтобы спрессовать в пару глав все, что вам нужно знать о COM, авторы предполагают, что вы уже знаете COM, поэтому мы будем рассматривать только непосредственно ATL. Если вы хотите изучить COM, попробуйте почитать, например, книгу Box D. *Essential COM* (Addison-Wesley Professional, 1997).

Структура книги

За исключением первой главы каждая глава в книге требует прочтения предшествующих глав. Глава 1 представляет собой краткий обзор областей применения ATL и мастеров, помогающих в работе с ней. Однако если приходится погружаться в детали, эта глава будет отсылать вас к главам, в которых эти детали рассматриваются.

Главы 2–5 посвящены ядру ATL. В главе 2, “Строки и текст”, рассматриваются вопросы, связанные с обработкой строк в C++, COM и ATL. В главе 3, “Интеллектуальные типы ATL”, обсуждаются интеллектуальные типы, например, `CComPtr`, `CComQIPtr`,

¹ За исключением генерирования кода проектов Web-служб — в мастере, который это делает, флажок, включающий генерацию кода с атрибутами, установлен и сделан недоступным.

`CComBSTR` и `CComVariant`. Глава 4, “Объекты в ATL”, посвящена реализации объектов в ATL и возможным вариантам реализации интерфейса `IUnknown`. В главе 5, “COM-серверы”, рассматривается код, предоставляющий доступ к COM-объектам через COM-серверы. В главе 6, “Таблицы интерфейсов”, мы возвращаемся к реализациям `IUnknown`, но на этот раз сосредотачиваемся на реализации метода `QueryInterface`; в этой главе рассматриваются отделяемые интерфейсы и агрегирование. Главы 7, “Сохраняемость в ATL”, 8, “Коллекции и перечислители”, и 9, “Точки соединения”, посвящены соответственно поставляемым с ATL реализациям интерфейсов, предоставляющих поддержку сохраняемости объектов, коллекций и перечислителей и точек соединения. Главы 10, “Работа с окнами”, 11, “Элементы управления ActiveX”, и 12, “Использование элементов управления”, посвящены созданию оконных приложений и компонентов пользовательских интерфейсов. В них рассматриваются оконные классы ATL, элементы управления и использование этих элементов в программах. И наконец, главы 13, “Здравствуй, ATL Server”, и 14, “Внутренняя структура ATL Server”, посвящены ATL Server, части ATL, позволяющей создавать Web-приложения на основе IIS. В главе 13 рассматриваются ISAPI и основы работы с ATL Server, а в главе 14 — внутренние механизмы ATL Server.

Значительная часть сложностей в разборе исходного кода ATL вызвана широким применением в ней шаблонов. Создание и применение шаблонов подробно рассматриваются в приложении А, “Шаблоны C++ в примерах”. Если вы просматривали исходный код ATL и вам интересно, зачем имя производного класса передается в качестве параметра шаблону базового класса, это приложение вам пригодится. В приложении Б, “Заголовочные файлы ATL”, приведен список заголовочных файлов с кратким описанием содержимого каждого файла. Это приложение позволит быстро найти файл, содержащий нужную часть ATL. Если вы уже знакомы с ATL 3 и хотите быстро узнать, что нового появилось в ATL 8, для вас предназначено приложение В, “Переход к ATL 8”. И наконец, если вы хотите побольше узнать об атрибутах (и о том, почему мы вынесли их обсуждение в последнее приложение), прочитайте приложение Г, “Атрибуты в ATL”.

Оформление

В этой книге нам придется не столько писать свой код, сколько изучать код реализации ATL. Можете считать ее вашим персональным гидом по исходному коду ATL. Чтобы отличить код, написанный авторами книги, от кода, написанного разработчиками Microsoft, мы будем оформлять код следующим образом.

```
// Этот код написан авторами книги.
// На фрагменты кода, выделенные полужирным начертанием,
// следует обратить особое внимание.
CComBSTR bstr = OLESTR("Hello, World.");

// Код, выделенный цветом фона, является частью ATL или Windows.
CComBSTR(LPCOLESTR pSrc) { m_str = ::SysAllocString(pSrc); }
```

Поскольку разработчики ATL не рассчитывали, что написанный ими код будет публиковаться в книгах, нам часто приходилось изменять его оформление или даже удалять некоторые его фрагменты. Мы сделали все, что было в наших силах, для сохранения нужных фрагментов, но в любом случае исходные коды ATL будут истинной в последней инстанции.

Код примеров и дополнительная информация

Исходный код примеров из книги и дополнительная информация доступны по адресу www.sellbrothers.com/writing/atlbook. Кроме того, на этом сайте вы можете оставить отзыв о книге или сообщить о найденных ошибках.

Благодарности

Авторы хотели бы выразить свою благодарность множеству людей, которые внесли свой вклад в создание этой книги. Крис Селлз (Chris Sells) хотел бы поблагодарить свою жену, Мелиссу, и сыновей, Джона и Тома, за их терпение и вежливость в бесчисленные вечера и уик-энды, проведенные им за работой. Кроме того, Крис искренне благодарен Бренту Ректору (Brent Rector) за разрешение воспользоваться первым изданием книги, Кирку Фергитта (Kirk Fertitta) за обновление изрядной части кода в книге под ATL 7, и Крису Таваре (Chris Tavares) за доведение этого проекта до ума.

Брент хотел бы поблагодарить свою жену, Лизу, и детей, Карли и Сина, за то, что они существенно задержали выход этой книги. Если бы не они, он, вероятно, просидел бы за компьютером в течение многих дней. Кроме того, Брент благодарен Крису Селлзу за терпение, ум и отличный характер.

Кирк Фергитта благодарен: читателям, ради которых и создавалась эта книга; Крису Селлзу за настойчивость, с которой тот втягивал его в ее написание, и за неоценимый опыт, полученный от него; Брэду Ханде (Brad Handa) и Гуго Валуа (Hugues Valois) за бесконечные часы, проведенные за изучением COM и ATL; всем подписчикам DevelopMentor, участвовавшим в обсуждении ATL, за их замечания и советы; работникам MusicMatch (теперь принадлежащего Yahoo) за опыт в разработке приложений на основе ATL, которым они с готовностью делились (именно они впервые обнаружили множество ловушек, связанных с использованием атрибутов в ATL); и Стефани Томас (Stephane Thomas) из издательства Addison-Wesley за ее терпение при работе над этой книгой.

Крис Таваре прежде всего благодарен своей многострадальной жене, Венди, за ее понимание, любовь и поддержку. Неисчислимы бутерброды и стаканы сока, которые она приносила, были абсолютно необходимы для сохранения рассудка при работе над книгой. Кроме того, Крис благодарен своему сыну Мэтью, который не слишком возражал, когда его отец целыми днями пропадал в офисе. И наконец, необходимо выразить благодарность Крису Селлзу за предоставленную им возможность написать второе издание книги об ATL.

Авторы хотели бы вместе выразить благодарность множеству людей, прежде всего — редакторам: Биллу Крауну (Bill Craun), Джоан Эрикссон (Joan Ericsson), Игорю Тандетнику (Igor Tandetnik), Киму Грассману (Kim Grasman), Джефу Галиновски (Jeff Galinovski) и Ненаду Стефановичу. Кроме того, авторы благодарят разработчиков ATL, включая Кристиана Бомонта (Christian Beaumont), Джима Спрингфилда, Уолтера Салливэна (Walter Sullivan) и Марка Крамера (Mark Kramer), за колоссальное терпение при ответах на глупые вопросы. Отдельной благодарности заслуживает Дон Бокс (Don Box) за множество ценных дополнений к книге. Также авторы благодарят рецензентов — Дона Бокса, Кейт Браун (Keith Brown), Джона Флэндера (John Flanders), Майка Фрэнсиса (Mike Francis), Кевина Джонса (Kevin Jones), Стэнли Липпмана (Stanley Lippman), Дхарма Шукла (Dharma Shukla), Джи-

ма Спрингфилда, Джефа Столлса (Jeff Stalls), Джаганатана Хангавелу (Jaganathan Thangavelu) и Джейсона Уиттингтона (Jason Whittington). Особая благодарность Дхарме за подробные и четкие рецензии. Благодарим Фрица Аниона (Fritz Onion) за работу над частями книги, касающимися размещения элементов управления. Спасибо бывшему ученику, Валдану Видаковичу (Valdan Vidakovic), за то, что он убедил авторов подробнее рассмотреть использование элементов управления HTML. Спасибо Тиму Эвальду, Джиму Спрингфилду и Дону Боксу за помощь в создании фокуса с передачей меток. Спасибо участникам конференций по ATL и DCOM, особенно Дону Боксу, Тиму Эвальду (Tim Ewald), Чарли Кинделу (Charlie Kindel), Валерию Прямикову (Valery Pryamikov), Марку Райленду (Mark Ryland) и Зейну Томасу (Zane Thomas). Также мы хотели бы поблагодарить Джорджа Шепарда (George Shepherd) за то, что он начал создание глав, посвященных ATL Server. И наконец, мы хотим поблагодарить работников Addison-Wesley, особенно Карен Гетман (Karen Gettman), Лори Лайонс (Lori Lyons) и Ким Бедигхаймер (Kim Boedigheimer), за создание среды, в которой у нас возникает желание писать книги (хотя, возможно, и не такие хорошие, как им хотелось бы...).

Об авторах

Крис Таваре в данный момент работает инженером по разработке программного обеспечения в Microsoft, изучая методики эффективной разработки ПО на платформах Microsoft. Он впервые начал эксперименты с компьютером в третьем классе, вручную составляя программы в машинных кодах для машины с процессором Intel 8080, 512 байтами (да, байтами) оперативной памяти и 7-сегментным ЖК-дисплеем.

Кирк Фертитта работает в компании Pacific MindWorks, передовом поставщике измерительных и контрольных электронных устройств. Он активно занимается технологиями генерации кода и дополнениями к Visual Studio. Кроме того, он работает инструктором по .NET/C# в компании Pluralsight.

Брент Ректор, президент и основатель компании Wise Owl Consulting — известный оратор, консультант и автор, специализирующийся на .NET, ASP.NET, XML, COM+ и ATL.

Крис Селлз работает менеджером по программному обеспечению в компании Connected Systems Division. Он написал несколько книг, например, *Programming Windows Presentation Foundation*, *Windows Forms Programming in C#* и *ATL Internals*. В свободное время Крис участвует в различных конференциях и проявляет худшие стороны своего характера, обсуждая на закрытых форумах продукты Microsoft. Дополнительная информация о Крисе и его проектах доступна на сайте <http://www.sellbrothers.com>.