

Крис Таваре

Венди, которая верила в меня, даже когда я сам в себя не верил.

Кирк Фертитта

Маме и Эду, за их непоколебимую веру в меня и поддержку.

Брент Ректор

Лизе, которая остается со мной и всегда рада мне вот уже почти тридцать лет.

Карли и Сину, которые не считают, что писать книги сложно; в конце концов, их отец только этим и занимается.

Крис Селлз

Это издание посвящается программистам под Windows, которые еще не перешли на .NET. В будущем вы, надеюсь, перейдете на .NET, а до тех пор — эта книга для вас.

Предисловие ко второму изданию

Прошло много времени с тех пор, как я написал предисловие к первому изданию этой книги. Сейчас, читая старое предисловие, я погружаюсь в воспоминания. Я с трудом могу поверить, что прошло почти восемь лет. Вскоре после его написания я вошел в команду разработчиков Windows в Microsoft, затем, спустя год, ушел из нее, снова вернулся (на этот раз в команду Visual C++) и сейчас работаю с несколькими командами разработчиков, связанными с Visual C++. Одна из этих команд занимается библиотеками, в том числе и ATL, и мне приятно снова работать с ATL. Ян и Кристиан уже ушли из этой команды, а Ненад вынес оконные классы ATL, которые я упоминал в старом предисловии, в отдельную библиотеку, которая называется WTL (Windows Template Library). Сейчас WTL — это проект Microsoft с открытыми исходными кодами, которым управляет Ненад.

ATL изменилась так, как мы даже не могли себе представить, и было интересно наблюдать, как она развивается без моего вмешательства. Над ATL в разные моменты времени работали многие замечательные люди. С некоторыми из этих людей я знаком очень хорошо, а о некоторых — ни разу не слышал.

Когда я упоминал в старом предисловии “новые способы доступа к функциональности”, я имел в виду атрибуты. Эта технология появилась в Visual Studio .NET 2002, но она так и не стала тем, на что мы рассчитывали. Атрибуты работают и в новой версии Visual Studio, и они дают немало интересных возможностей. Однако развивать их дальше не планируется. В новой версии этой книги атрибуты рассматриваются, но она не предполагает, что вы будете активно их использовать. Кроме того, в этой версии есть хороший обзор ATL Server, позволяющий быстро создавать эффективные Web-приложения. Если для вас прежде всего важна производительность, ATL Server — это именно то, что вам нужно. Среди других новшеств ATL 8 следует упомянуть новые функции безопасности, поддержку 64-битовых вычислений, лучшую масштабируемость, улучшенные возможности отладки, поддержку C++/CLI и управляемых компонентов ATL.

В 1998 году то, что сейчас превратилось в платформу .NET, только зарождалось. Для многих программистов .NET стала революцией, и с течением времени эта платформа будет обрести все новые возможности. Однако программирование под СОМ (в том числе с использованием ATL) по-прежнему востребованно, и им занимаются множество программистов и в Microsoft, и вне ее. Второе издание этой книги, как и первое, содержит детали, которые позволят вам извлечь максимум пользы из этих технологий.

*Джим Спрингфилд (Jim Springfield)
Апрель 2006 года*

Предисловие к первому изданию

Когда я увидел название этой книги, то сразу же сказал Крису Селлзу, что всегда хотел написать эту книгу сам. С тех пор, как мы выпустили первую версию ATL, многие из нас считали, что нужно написать книгу, в которой рассказывалось бы, как ATL работает. Прочитав эту книгу, я решил, что в ней уже рассказано почти все, что я хотел бы рассказать. На мой взгляд, в ней рассмотрено большинство аспектов ATL, и она представляет собой замечательный источник информации о внутренних механизмах ATL. Поэтому Крис попросил меня изложить в предисловии информацию, которую нельзя получить, просто изучая исходный код ATL.

Краткая история ATL

Я впервые столкнулся с шаблонами в 1995 году, когда был разработчиком в команде, работавшей над MFC. Мой друг в то время изучал варианты STL от разных разработчиков, выбирая наиболее подходящий для включения в Visual C++, и он постоянно обсуждал со мной шаблоны. Я немного повозился с шаблонами, но особо ими не заинтересовался. Позже из команды Visual C++ была выделена отдельная группа, занимавшаяся исключительно Visual C++ 4.2 Enterprise (первый продукт Visual C++ серии Enterprise). В этой группе я отвечал за библиотеки. В то время мы рассматривали целый ряд идей. В то время Microsoft Transaction Server только зарождался, и мы много говорили о COM, транзакциях, базах данных и коммерческих приложениях. Вскоре мы поняли, что нам нужен более удобный механизм создания объектов COM в C++. В то время со мной работали Ян Фалкин (Jan Falkin) и Кристиан Бомонт (Christian Beaumont). Ян занимался интерфейсом автоматизации для источников данных ODBC (предшественником современных потребительских шаблонов OLEDB). Я работал над инфраструктурой COM, потому что уже в то время все сводилось к COM.

На первых порах я просто пробовал то одно, то другое, но постепенно мне стало понятно, что нужно сосредоточиться на нескольких базовых концепциях. Я с самого начала хотел обеспечить поддержку всех потоковых моделей, но не хотел тратить на эту поддержку ресурсы, если она оказывалась не востребованной. То же самое относилось и к агрегированию. Я не хотел, чтобы у кого-то были поводы отказываться от ATL (вроде, *да, ATL — это круто, но я могу сэкономить пару байтов там и тут, если сделаю все сам*). Таким образом производительность и гибкость стали первоочередными требованиями, и ценой за их выполнение стала сложность. Одной из основных идей стало то, что на самом деле должны создаваться объекты не того класса, который напишет пользователь. Такой подход позволял реализовать множество оптимизаций, которые в других вариантах были невозможны. Среди других концепций были множественное наследование для интерфейсов, функции “создателей” и таблицы

макросов. Мы продемонстрировали результаты наших усилий и получили в общем положительные отзывы. Некоторые люди считали, что мы должны предоставить наши достижения потребителям как можно быстрее, поэтому сетевой выпуск состоялся в начале лета 1996 года. Это была ATL 1.0. Рабочим названием для нее было MEC (Microsoft Enterprise Classes), но спецы по маркетингу решили, что нам нужно придумать что-то, более четко отражающее назначение библиотеки. Поскольку эта библиотека предназначалась для работы с COM, а в то время в названиях чего угодно обязательно присутствовало слово Active, мы решили назвать библиотеку Active Template Library. Отзывы о ней были неплохи, и в конце лета 1996 года мы выпустили ATL 1.1. К тому времени Ян и Кристиан стали работать непосредственно над ATL. В ATL 1.1 был исправлен ряд ошибок и добавлены новые возможности, например, поддержка точек соединения, служб NT, реестровых скриптов RGS и функций безопасности.

Выпустив ATL 1.1, мы начали работу над ATL 2.0. В ней основной целью было упрощение создания элементов управления ActiveX. Ян и Кристиан много работали над этой задачей, а я по-прежнему возился с базовыми механизмами (например, переписал точки соединения, чтобы сделать их более компактными). В то время к нам присоединился Ненад Стефанович (Nenad Stefanovic), который занялся поддержкой оконных классов в ATL и поддержкой составных элементов управления в Visual C++ 6.0. Изначально мы планировали распространять ATL 2.0 по сети для Visual C++ 4.2. Однако позже наши планы изменились — мы собрались выпустить ATL 2.0 вместе с Visual C++ 5.0 (в декабре 1996 года) и выпустили ATL 2.1 с альфа-версией Visual C++ 5.0. Единственным отличием ATL 2.1 от ATL 2.0 были несколько исправленных ошибок для платформ Alpha, MIPS и PowerPC. Кроме того, мы одновременно выпустили ATL 2.1 в сети вместе с поддержкой мастеров для Visual C++ 4.2. После пары месяцев работы над ATL 3.0 (в то время она называлась ATL 2.5) Кристиан и я перегорели и решили на некоторое время переключиться на другую работу, а ведущим разработчиком ATL стал Ян. Через несколько месяцев мы вернулись, и ведущим разработчиком стал Кристиан, а я занялся некоторыми другими вещами в Visual C++, хотя, как и раньше, время от времени работаю с исходными кодами.

В июне 1998 года мы выпустили Visual C++ 6.0 и начали работу над следующей версией. В ATL появилось много новых классов и новые способы доступа к ранее существовавшим классам. Я рад, что ATL продолжает развиваться, позволяя в то же время генерировать компактный, быстрый код. Читайте эту книгу, учитесь новым фокусам и разбирайтесь в том, как все это работает.

*Джим Спрингфилд
Октябрь 1998*