

Использование COM-технологии в C++Builder:

- удаленный запуск серверов автоматизации
- создание и использование OLE-документов в приложениях

Наталья Елманова

Последняя статья данного цикла посвящена двум немаловажным вопросам использования COM-технологии в C++Builder: удаленному запуску серверов автоматизации и использованию в приложениях C++Builder OLE-документов.

Удаленный запуск серверов автоматизации

В предыдущих статьях данного цикла было рассказано о создании серверов и контроллеров автоматизации. Отметим, что серверы автоматизации могут выполняться как в адресном пространстве контроллера (такие серверы называются in-process-серверами и реализуются обычно в виде динамически загружаемых библиотек), так и в собственном адресном пространстве (такие серверы называются out-of-process-серверами и реализуются, как правило, в виде исполняемых файлов). Приложения MS Office, а также сервер автоматизации, создание которого рассматривалось в одной из предыдущих статей, относятся ко второму типу COM-серверов (такие COM-серверы называются локальными).

Отметим, что локальный COM-сервер можно превратить в удаленный, то есть запускаемый на компьютере сети, отличном от компьютера, содержащего контроллер. Такой способ использования серверов особенно удобен в случае, когда сервер требует для своей работы наличия особых ресурсов, которые недоступны на всех компьютерах, содержащих контроллеры, — дополнительного объема оперативной памяти, уникального оборудования, дополнительного программного обеспечения, требующего сложной конфигурации, настройки и поддержки (а если обратить внимание на другие механизмы использования сервисов, доступных в сети, отличные от COM, то, возможно, и другой операционной системы).

Один из часто встречающихся примеров практического использования удаленного доступа — использование удаленного запуска MS Excel, установленного на одном-единственном компьютере локальной сети, для печати отчетов и диаграмм. В этом случае можно снизить требования к ресурсам рабочих станций, содержащих приложения-контроллеры Excel (и сэкономить некоторые средства на покупке нескольких копий Excel).

Второй пример использования удаленного запуска серверов автоматизации, вызывающий в после-

днее время немалый интерес разработчиков информационных систем, это пример трехзвенной информационной системы с «тонким» клиентом — контроллером автоматизации, и сервером автоматизации (в данном случае он иногда называется сервером приложений), предоставляющим «тонкому» клиенту сервисы, связанные с доступом к данным, содержащимся, в свою очередь, в какой-либо серверной СУБД. В этом случае, помимо экономии ресурсов компьютеров, содержащих контроллеры, имеется еще одно преимущество, связанное с тем, что для доступа к данным используется программное обеспечение, требующее отдельной установки, сложной настройки и поддержки, такое как клиентская часть серверной СУБД, а также, как правило, библиотека Borland Database Engine, драйверы SQL Links, а иногда и ODBC-драйверы. Если это программное обеспечение используется только сервером, но не используется контроллерами, то сопровождение такой системы значительно облегчается.

При рассмотрении работы таких серверов возникает естественный вопрос: каким образом контроллер обращается к свойствам и методам объектов, расположенных за пределами его адресного пространства? Для этого следует рассмотреть, каким образом вообще взаимодействуют COM-клиент и COM-сервер.

Использование интерфейсов

Единственным способом взаимодействия клиента с COM-объектом является использование интерфейсов. Интерфейс связан с группой функций, связанных, в свою очередь, с COM-объектом, и не содержит их реализации. На этапе выполнения интерфейс всегда представляется указателем, идентифицируемым с помощью идентификатора IID (Interface Identifier — по существу, это тот же GUID) и указывающим на другой указатель, который, в свою очередь, указывает на таблицу, содержащую адреса реализации каждой функции из этой группы (рис.1).

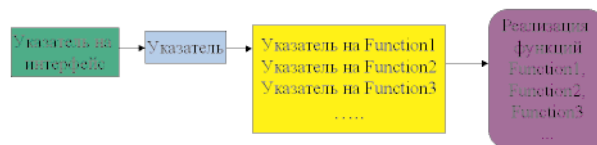


Рис.1. Структура интерфейса

Эта структура позволяет обеспечить маршалинг — пересылку указателя между процессами (и в общем случае между компьютерами).

Реализация интерфейса COM-объекта представляет собой создание в памяти подобной структуры и предоставление указателя на нее. Фактически такая структура похожа на структуру, используемую для построения таблиц виртуальных функций в C++, поэтому таблица указателей на функции в этой структуре иногда называется «vtable» (или таблицей виртуальных методов).

Весьма существенно, что адреса самих функций определяются на этапе выполнения с помощью указателя на интерфейс в момент обращения к ним, а не хранятся в памяти статически. Отметим также, что при наличии указателя на функцию в таблице указателей всегда существует и ее реализация, поэтому клиент не должен ни обрабатывать сообщения об обращении к несуществующим функциям, ни иметь собственную версию их реализации.

Когда клиенту требуется COM-объект, он пытается найти сервер, посылает серверу запрос на создание объекта, а затем получает от него указатель на исходный интерфейс, с помощью которого можно получить дополнительные указатели на другие интерфейсы этого объекта.

COM находит местоположение сервера на основании записи в реестре, загружает его в оперативную память, запрашивает у сервера нужный объект (иницилируя его создание) и получает указатель на интерфейс. Если сервер и клиент выполняются в разных процессах, COM автоматически передает указатель на интерфейс в процесс клиента.

Маршалинг и удаленный доступ

Если сервер является внутренним («in-process»), то есть выполненным в виде DLL, она загружается в адресное пространство клиента с помощью функции Win32 API LoadLibrary. В этом случае значение указателя на интерфейс непосредственно доступно клиенту.

Если сервер локальный, COM использует функцию CreateProcess, загружая исполняемый файл и инициализируя COM в адресном пространстве последнего. Затем COM в процессе сервера связывается с COM в процессе клиента каким-либо доступным способом. В этом случае обычно нет возможности передать клиенту точное значение указателя на интерфейс, так как это указатель на объект в другом адресном пространстве. В этом случае используется маршалинг, создающий так называемый «marshalling packet», содержащий необходимую информа-

цию для соединения с процессом, в котором создан объект. Этот пакет создается с помощью функции COM API CoMarshalInterface; затем он передается процессу клиента любым доступным способом, где другая функция CoUnMarshalInterface превращает этот пакет в указатель на интерфейс. В действительности маршалинг создает в обоих адресных пространствах два объекта: Stub («заглушка») — представитель клиента в адресном пространстве сервера, имеющий дело с реальным указателем на интерфейс, и Proxy — представитель сервера в адресном пространстве клиента. Эти два объекта соединяются между собой каким-либо доступным способом, и представитель сервера передает клиентскому процессу указатель на интерфейс. Сходная технология используется при осуществлении вызовов удаленных процедур (RPC — Remote Procedure Calls).

Естественно, Proxy-объект не содержит реализации интерфейса. Все аргументы вызываемых функций помещаются в пакет, передаваемый Stub-объекту с использованием RPC. Stub-объект распаковывает переданные аргументы, помещает их в стек и обращается к реальному объекту, используя существующий указатель на интерфейс. Результат выполнения функции упаковывается в пакет и посылается Proxy-объекту, который распаковывает его и передает клиенту.

В случае сервера, расположенного на удаленном компьютере, при обращении к серверу COM соединяется со специальным резидентным процессом удаленного компьютера, контролирующим удаленный запуск сервисов на нем. Этот процесс, называемый иногда Service Control Manager — SCM, осуществляет запуск сервера на удаленном компьютере и возвращает указатель на интерфейс клиентскому компьютеру и клиентскому процессу. В качестве SCM могут быть использованы средства различных производителей, как основанные непосредственно на технологии Microsoft DCOM (Distributed COM), так и являющиеся расширением этой технологии (например, ObjectFactory из комплекта поставки OLEnterprise корпорации Inprise). Обязательное наличие такого процесса (и возможностей его конфигурации) диктуется элементарными соображениями безопасности: было бы неестественно, если бы любой пользователь сети мог запустить лю-

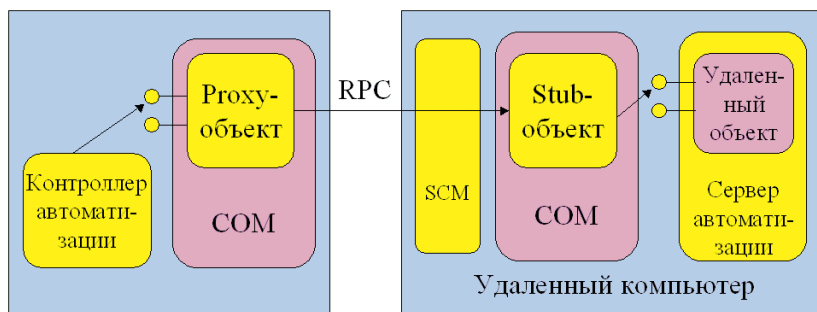


Рис. 2. Осуществление удаленного доступа к серверу автоматизации

бой процесс на любом из компьютеров, подключенных к этой сети. В остальном маршрутинг осуществляется точно так же, как и в случае локального сервера, за исключением того, что проxy и stub, общающиеся с помощью того же самого механизма RPC, физически находятся на разных компьютерах (рис. 2).

Как осуществить удаленный доступ

Итак, для осуществления удаленного запуска сервера автоматизации следует запустить на стороне компьютера-сервера Service Control Manager, предварительно сконфигурировав его так, чтобы удаленный пользователь мог осуществить запуск данного сервера.

Использование Microsoft DCOM в качестве Service Control Manager

Рассмотрим использование Microsoft DCOM в качестве Service Control Manager. Для конфигурации DCOM существует утилита DCOMCNFG, входящая в комплект поставки Windows NT.

Использование DCOM в качестве SCM базируется на предоставлении прав на удаленный запуск конкретного приложения тем или иным пользователям или группам пользователей. Поэтому компьютер, содержащий сервер, который планируется запускать удаленно, должен иметь в своем реестре описание прав доступа к данному серверу различных пользователей или их групп.

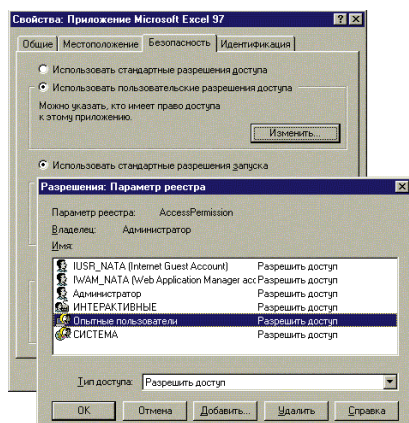


Рис. 3. Настройка прав доступа к серверу автоматизации

Для этого требуется, в свою очередь, экспорт сведений о пользователях сети с первичного контроллера домена, что возможно только в случае выбора на таком компьютере установки User Level Access Control раздела Network панели управления Windows.

В реестре компьютера, на котором запускается контроллер автоматизации, также должны содержаться сведения о том, что соответствующий сервер автоматизации запускается удаленно. Для этого нужно, чтобы в реестре о сервере автоматизации уже была соответ-

ствующая запись, поэтому его необходимо хотя бы один раз запустить локально (или перенести соответствующие записи из реестра компьютера, содержащего сервер, на компьютер, содержащий контроллер).

После этого можно с помощью той же утилиты DCOMCNFG указать, что данное приложение запускается удаленно.

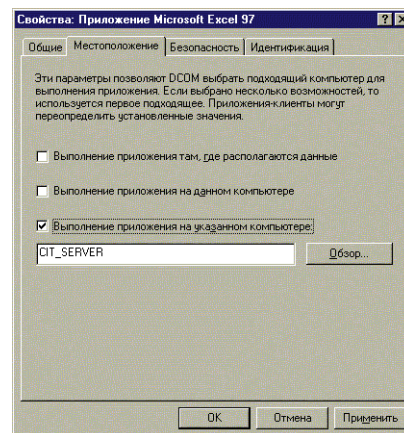


Рис. 4. Настройка удаленного запуска приложения-сервера

После этого можно попытаться запустить контроллер. Сервер при этом запустится удаленно.

Хотелось бы особо обратить внимание на достоинства и недостатки использования DCOM в качестве SCM. Несомненным достоинством этого способа является то, что DCOM интегрирован в операционную систему Windows NT и не требует установки и запуска каких-либо утилит для своего функционирования. Недостатки же этого способа удаленного запуска серверов автоматизации таковы:

- в сети обязательно должен быть первичный контроллер домена;
- могут потребоваться существенные изменения настроек сети (для установки опции User Level Access Control);
- сервер автоматизации нужно хотя бы один раз запустить на компьютере с контроллером для его регистрации.

Помимо этого, есть определенные сомнения в возможности использования Windows 95 в качестве операционной системы для компьютера, содержащего удаленный сервер автоматизации. По крайней мере, на сегодняшний день автору неизвестны успешные случаи удаленного запуска подобных серверов на компьютерах, оснащенных этой операционной системой, с помощью DCOM. Использование Windows 98 в этом качестве автором не проверялось.

Использование Inprise OLEnterprise в качестве Service Control Manager

Рассмотрим альтернативный способ запуска удаленных серверов автоматизации, использующий в качестве Service Control Manager продукт OLEnterprise

(Inprise Corporation), входящий в комплект поставки C++Builder Client/Server Suite.

Использование OLEnterprise в качестве SCM базируется на предоставлении прав на удаленный запуск конкретного приложения всем пользователям, имеющим установленную клиентскую часть этого продукта. В первую очередь следует запустить на компьютере, содержащем сервер автоматизации, утилиту Object Factory, которая, по существу, и выполняет роль SCM. Предоставление прав на запуск сервера владельцем компьютера, содержащего этот сервер, осуществляется путем публикации соответствующих записей реестра (или, в терминах OLEnterprise, экспорта соответствующего сервера автоматизации в глобальный реестр) с помощью утилиты Object Explorer:

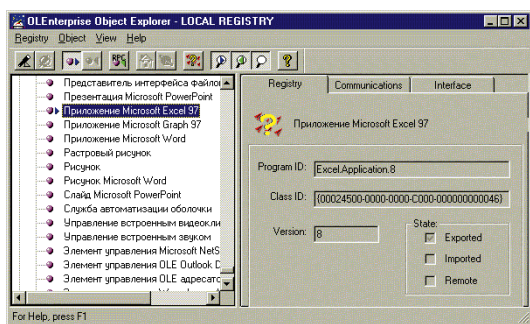


Рис. 5. Публикация сведений о приложении-сервере в глобальном реестре

На компьютере, содержащем контроллер, с помощью этой же утилиты можно соединиться с удаленным реестром компьютера, содержащего сервер автоматизации, и импортировать запись о нем в собственный реестр:

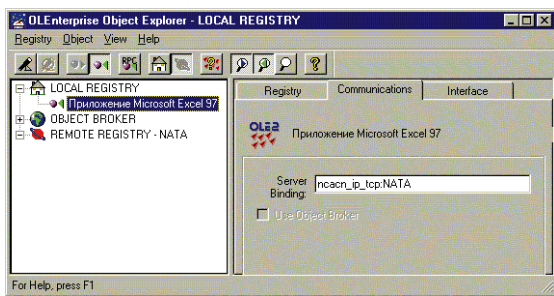


Рис. 6. Импорт записи из глобального реестра в реестр компьютера, содержащего контроллер автоматизации

После этого при запуске контроллера сервер запустится удаленно.

В данном примере компьютер, на котором запускается контроллер автоматизации, «знает» об одном-единственном компьютере, содержащем сервер (точно так же, как и в рассмотренном выше примере использования DCOM). Однако возможности OLEnterprise этим не исчерпываются. В комплект его поставки входит утилита Business ObjectBroker, назначение которой — вза-

имодействовать с опубликованными частями реестров сети в поисках сервера автоматизации для обратившихся к нему клиентов. Если сконфигурировать OLEnterprise так, чтобы использовать Business ObjectBroker, клиенты автоматизации должны обращаться к нему, а не непосредственно к серверам автоматизации, как в предыдущем случае. В ответ они будут получать указание, с каким из компьютеров сети соединиться для использования сервера автоматизации.

Если компьютеров, опубликовавших в глобальном реестре сведения об одном и том же сервере автоматизации, несколько, обращения компьютеров, содержащих контроллеры, будут распределены между ними случайным образом, что позволяет возложить равномерную нагрузку на эти компьютеры.

Рассмотрим достоинства и недостатки использования OLEnterprise в качестве SCM. Достоинства этого способа очевидны: наличие первичного контроллера домена в сети в этом случае не является обязательным; существенных изменений настроек сети не требуется; сервер автоматизации не нужно запускать на компьютере, содержащем контроллер. Есть возможность запускать удаленные серверы автоматизации под управлением Windows 95. Помимо этого, можно иметь несколько дублирующих друг друга серверов автоматизации на разных компьютерах сети, и можно написать код контроллера таким образом, чтобы в случае сбоя используемого сервера данный контроллер с помощью Business ObjectBroker был подключен к другому компьютеру, содержащему сервер, что повышает надежность такой системы. Отметим, что в случае использования DCOM это практически невозможно — в настройках DCOM местоположение сервера всегда указывается явно, а использование брокеров не предусмотрено.

Недостатки у этого способа тоже есть — в отличие от DCOM, автоматически устанавливающегося вместе с Windows NT, OLEnterprise требует отдельного приобретения и установки, в том числе установки клиентской части OLEnterprise на компьютеры, содержащие контроллеры автоматизации.

Отметим, что список возможных SCM отнюдь не исчерпывается DCOM и OLEnterprise. Помимо этого, возможно использование других подобных резидентных утилит, позволяющих при тех или иных условиях запускать удаленным пользователям серверы автоматизации. Например, Borland Socket Server, входящий в комплект поставки C++Builder Client/Server Suite, является еще одним типом SCM и позволяет запускать определенные типы серверов автоматизации, не требуя при этом на компьютере, содержащем контроллер, никаких дополнительных клиентских частей и настроек. ■

Координаты автора:

Центр Информационных Технологий.

Тел.: (095)932-92-12, 932-92-13, elmanova@citmgu.ru,
<http://www.citmgu.ru>, <http://www.citforum.ru>