**MIDAS и создание серверов приложений**

Наталия Елманова, Центр Инфомационных Технологий

После выхода статьи "Создание серверов приложений с помощью Delphi 3" от читателей и слушателей проводимых в Учебно-консалтинговом центре Interface Ltd. курсов, посвященных Delphi 3 и MIDAS, поступило большое количество вопросов, касающихся технических и организационных аспектов разработки и эксплуатации трехзвенных информационных систем с "тонким" клиентом. Данная статья посвящена особенностям и наиболее предпочтительным архитектурам построения подобных информационных систем в случае различных схем организации функционирования обслуживаемых ими предприятий, новым возможностям создания "тонких" клиентов, предоставляемым Delphi 3.01/3.02 и С++Builder 3.0, ответам на наиболее часто встречающиеся вопросы, а также наиболее распространенным ошибкам при создании серверов приложений и "тонких" клиентов.

**Различные архитектуры построения трехзвенных информационных систем**

Нередко у потенциальных пользователей MIDAS возникают вопросы о способах построения многозвенных информационных систем, наиболее удовлетворяющих нуждам именно их предприятия, в том числе вопросы о том, сколько нужно иметь серверов приложений, как конфигурировать клиентские рабочие станции, каковы возможности осуществления и изменения конфигурации рабочих станций удаленно, например, через Internet или с помощью модемной связи.

Многозвенные информационные системы можно построить разными способами, и выбор той или иной архитектуры существенно зависит от числа пользователей такой ИС, степени территориальной разбросанности предприятия, потребности в централизованном хранении и обработке данных, частоты обновления версий используемых автоматизированных рабочих мест и структуры используемой базы данных.

Первый часто встречающийся случай - крупное или среднее предприятие, расположенное компактно и имеющее общую локальную сеть. В этом случае нередко корпоративные данные хранятся в общей для всех серверной СУБД. Переход к трехзвенной архитектуре в этом случае производится главным образом с целью снижения издержек, связанных с конфигурацией рабочих станций (установка и настройка клиентской части серверной СУБД, настройка BDE, обновление версий автоматизированных рабочих мест) и поддержкой их в рабочем состоянии, и, во вторую очередь - с целью снижения сетевого трафика и повышения надежности эксплуатации системы в целом. В этом случае наиболее приемлемым представляется наличие в сети нескольких компьютеров, на которых установлены MIDAS и OLEnterprise и эксплуатируются одни и те же серверы приложений, при этом сведения о них экспортированы в глобальный реестр (например, с помощью утилиты Object Explorer). На рабочих станциях в этом случае устанавливается Run-time - версия OLEnterprise, и все описания серверов приложений импортируются в локальные реестры. При такой конфигурации рабочие станции подключаются к серверам приложений случайным образом и в случае отказа одного из серверов эти подключения окажутся равномерно распределенными по другим серверам.

В случае очень большого числа пользователей и высокой интенсивности транзакций можно рекомендовать как альтернативу "самодельным" серверам приложений серверы приложений Borland Entera, которые, в отличие от серверов, созданных с помощью Delphi, могут управляться не только помощью Windows 95 и Windows NT, но и с помощью других операционных систем (в частности, различных версий Unix). В этом случае вместо Delphi Client/Server Suite следует использовать Delphi Enterprise, что позволит создать "тонких" клиентов для использования Entera в качестве сервера приложений.

Второй случай, характерный, в частности, для многих предприятий нефтяной и газовой промышленности, крупных торговых объединений, а также различных межрайонных и межрегиональных служб, от налоговых инспекций до авиационных касс - предприятие, имеющее несколько территориально разбросанных подразделений (возможно, даже в разных городах), не связанных локальной сетью между собой. Нередко такие предприятия имеют несколько локальных информационных систем и набор связанных с этим проблем, таких, как поддержка актуальности данных во всех этих информационных системах, репликация данных, объединение данных из нескольких информационных систем для их совместного анализа и составления итоговых сводок и отчетов. Переход к трехзвенной архитектуре в случае таких предприятий осуществляется с целью устранения этих проблем и организации централизованного хранения и обработки данных (а нередко также с целью снижения тех же затрат на конфигурацию рабочих станций, что и в предыдущем случае). Обычно в этом случае наиболее приемлемым представляется установка сервера баз данных и сервера приложений в центральном офисе (не обязательно на одном и том же компьютере), при этом последний должен быть оснащен MIDAS и быть доступным через Internet. Рабочие станции филиалов должны быть оснащены Internet Explorer 3.0 и выше, при этом и разработчики, и пользователи рабочих станций удаленных филиалов должны иметь доступ не только к компьютеру, содержащему сервер приложений, но и к какому-либо web-серверу (его местоположение не играет существенной роли, так как первые лишь отправляют туда новые версии "тонких" клиентских приложений, а вторые их сгружают на свои рабочие станции). В этом случае никаких особых требований к рабочим станциям, помимо наличия Internet Explorer 3.0, а также доступа в Internet, нет. В этом случае, естественно, также возможно использование Borland Entera и Delphi Enterprise вместо MIDAS и Delphi Client/Server Suite.

В случае невозможности доступа через Internet возможны иные варианты, такие, как использование выделенной линии, модемная связь и др. Реализация связи не так принципиальна - лишь бы при этом была возможность использования протокола TCP/IP. Естественно, при частых сбоях и разрывах соединения рекомендуется активно использовать кэширование данных на рабочих станциях, сохранять содержимое кэша в файлах и загружать их оттуда, благо у компонента TClientDataSet есть методы SaveToFile и LoadFromFile.

Возможны различные комбинации этих вариантов, например, доступ по протоколу TCP/IP и наличие нескольких однотипных серверов приложений. Дать рекомендации на все случаи жизни, естественно, невозможно, но имеющиеся благодаря средствам разработки Borland технические возможности предоставляют разработчикам и руководителям отделов автоматизации достаточно большой простор для фантазии при организации информационных систем предприятий, и достаточный выбор вариантов их построения.

Создание сервера приложений и "тонкого" клиента с доступом по TCP/IP в Delphi 3.01/3.02 и в С++Builder 3.0.

Созданные с помощью Delphi 3.0 клиент и сервер приложений могли взаимодействовать посредством OLEnterprise либо DCOM. В версии Delphi 3.01появилась также возможность организовать связь клиента и сервера непосредственно с помощью протокола TCP/IP. Для этого в палитре компонентов последних версий Delphi (3.01 и 3.02) имеется компонент TMidasConnection, обладающий большей функциональностью по сравнению с TRemoteServer из Delphi 3.0. Этот компонент позволяет выбрать тип соединения с сервером приложений (DCOM, OLEnterprise, непосредственное использование протокола TCP/IP). При его использовании процесс создания сервера приложений и клиента заметно упрощается, так как в случае доступа по протоколу TCP/IP в общем случае нет необходимости иметь в реестре компьютера, на котором установлен "тонкий" клиент, какие-либо сведения о сервере приложений. Рассмотрим простейший пример создания такой системы.

Начнем с создания сервера приложений. Создадим главную форму приложения, основное назначение которой - служить индикатором запущенного сервера. С этой целью можно разместить ее где-нибудь в углу экрана, а свойство FormStyle установить равным fsStayOnTop, чтобы не потерять это окно среди других открытых окон. Далее следует создать удаленный модуль данных, выбрав пиктограмму Remote DataModule из репозитария объектов, поместить в него компонент TTable или TQuery, установив необходимые значения свойств DatabaseName и TableName. Свойство Active также следует установить равным True (или установить его динамически при создании модуля данных). В противном случае компонент не будет содержать никаких данных, и тем более предоставлять их клиентскому приложению (рис.1).

Рис.1. Форма и удаленный модуль данных сервера приложений

Затем нужно обязательно экспортировать этот компонент из модуля данных (либо связать его с компонентом TProvider и экспортировать последний). Если этого не сделать, клиентское приложение не будет иметь доступа к этому объекту (эта ошибка является довольно типичной). Основное назначение удаленного модуля данных заключается именно в предоставлении содержащимся в нем компонентам доступа к данным возможности быть видимыми и управляемыми извне, то есть из других приложений (и в общем случае - с других компьютеров, в том числе доступных не только посредством локальной сети, но и посредством Internet). Проконтролировать наличие в удаленном модуле данных объектов, видимых извне, можно, просмотрев соответствующую библиотеку типов (рис.2).

Рис.2. Библиотека типов, связанная с удаленным модулем данных

Если для доступа к данным необходим компонент TDatabase, его обычно размещают на главной форме приложения, либо в обычном модуле данных. Почему при использовании компонента TDatabase его обычно не помещают в удаленный модуль данных? Если этот компонент разместить в удаленном модуле данных, то могут возникнуть проблемы, связанные с тем, то при нескольких подключениях могут быть конфликты между несколькими его экземплярами - ведь каждое новое подключение создает свой экземпляр удаленного модуля данных. Поэтому следует позаботиться о разных именах пользовательских сессий во избежание таких конфликтов, либо, если это позволяют соображения безопасности, использовать для всех подключений общий компонент TDatabase.

Затем, если есть желание проконтролировать число соединившихся с сервером клиентов, можно создать обработчики событий OnCreate и OnDestroy удаленного модуля данных, в которых значение какой-либо целой переменной соответственно увеличивается или уменьшается на единицу и при необходимости отображается в каком-либо интерфейсном элементе главной формы сервера приложений.

После этого сервер приложений можно скомпилировать и запустить на выполнение. Это нужно для того, чтобы зарегистрировать его как OLE-сервер в реестре Windows. По поводу регистрации сервера также следует сделать одно замечание. Автору пришлось наблюдать случай, когда системный администратор сумел настроить операционные системы на компьютерах разработчиков отдела автоматизации одного из московских предприятий таким образом, что у них не было никакой возможности менять что-либо в своих реестрах. При этом, естественно, и серверы приложений также в реестрах не регистрировались. Если при создании сервера приложений он не регистрируется в реестре, возможно, этот случай подобен описанному. А может быть, просто по ошибке вместо удаленного создан обычный модуль данных...

Если обобщить данный случай, то, образно говоря, серверы приложений такого класса обязаны содержать либо генерировать некий набор SQL-запросов для изменений в базе данных и посылать их на сервер баз данных по команде клиентского приложения. Наш сервер приложений использует для генерации запросов библиотеку Borland Database Engine. Другие серверы приложений, такие, например, как Borland Entera, используют иные механизмы генерации запросов, обусловленные в известной степени той платформой, на которой этот сервер приложений функционирует. В любом случае именно набор SQL-запросов является самой главной составляющей функциональности такого сервера приложений. Что же касается пользовательского интерфейса сервера приложений (форма со счетчиком подключений или что-то в этом роде), он совершенно необязателен. В конце концов, сервер приложений может не иметь главной формы или не показывать ее, или может быть запущен в виде сервиса. Если же говорить опять об общем случае (когда речь идет не только о серверах приложений, созданных с помощью Delphi, но и о других подобных серверах приложений, в том числе для других платформ), то ведь не все платформы обладают графическим пользовательским интерфейсом, и, следовательно, создание такого объекта, как форма, возможно не на всех платформах. Не стоит обольщаться - самые дорогие серверы приложений нередко просто запускаются из командной строки и не обладают интерфейсом в привычном для пользователей Windows понимании.

В известном смысле сервер приложений и "тонкий" клиент представляют собой разделенное на две части классическое клиентское приложение, называемое в совокупности с клиентом серверной СУБД и библиотеками доступа к данным, такими как BDE, "толстым" клиентом. Первая часть (созданный нами сервер приложений) содержит компоненты доступа к данным (и требует наличия BDE и клиента серверной СУБД), а вторая (клиент) должна содержать пользовательский интерфейс (и не требовать наличия BDE и клиентской части серверной СУБД).

Прежде чем приступить к созданию "тонкого" клиента, запустим Borland MIDAS Socket Server scktsrvr.exe (его можно найти в каталоге Delphi 3.0\Bin), предоставляющий доступ извне к имеющимся на данном компьютере серверам приложений, созданных с помощью Delphi 3, по протоколу TCP/IP. Отметим, что в этом случае любой из имеющихся серверов приложений может быть запущен с любого компьютера, имеющего доступ к Вашему компьютеру с помощью данного протокола, поэтому при использовании подобных систем следует рассматривать различные вопросы, связанные с безопасностью их эксплуатации.

Рис.3. Borland MIDAS Socket Server

Теперь, наконец, можно создать клиентское приложение. Для этого создадим обычную форму (или выберем со страницы ActiveX репозитария объектов пиктограмму ActiveForm для создания клиентского компонента ActiveX). На форму поместим компонент TMidasConnection и установим его свойство ComputerName равным IP-адресу компьютера, на котором должен выполняться сервер приложений. Если этот компьютер в данный момент доступен в сети, можно выбрать его имя из списка, появляющегося при щелчке напротив имени этого свойства. Но нужно понимать, что в общем случае, особенно если клиентское приложение предназначено для доступа к серверу через Internet, указание именно IP-адреса является более предпочтительным. Далее следует установить значение свойства ServerName в следующем формате: <имя исполняемого файла>.<имя OLE-сервера>, например MyAppSrv.MyRemDataMod1. Свойство ServerGUID можно не устанавливать. Если сервер приложений не зарегистрирован на компьютере, где разрабатывается клиент, значение этого свойства должно остаться пустым, и это не помешает совместной работе сервера и клиента - ведь в общем случае при распространении клиентского приложения или ActiveX через Internet в реестре компьютера, где оно будет выполняться, нет и не может быть сведений о сервере приложений.

Свойство Connected можно установить равным True (или произвести установку этого свойства равным True на этапе выполнения в момент создания формы клиентского приложения). После этого должен запуститься сервер приложений (в том числе удаленный).

Далее следует поместить на форму компонент TClientDataSet, выбрать имя компонента TMidasConnection в качестве значения свойства RemoteServer и выбрать значение свойства Provider из выпадающего списка объектов, экспортированных из удаленного модуля данных сервера приложений. Теперь можно установить свойство Active равным true. Далее следует поместить на форму компонент TDataSource и связать его с компонентом TClientDataSet.

После этого можно заняться пользовательским интерфейсом, поместив на форму необходимые компоненты отображения данных (рис.4). Следует также в качестве обработчика какого-нибудь события вызвать метод ApplyUpdates компонента TClientDataSet, иначе изменения будут накапливаться в кэше и не будут сохраняться в базе данных.

Рис. 4. Активная форма на этапе проектирования.

Далее можно запустить и протестировать клиентское приложение, а если он выполнено в виде ActiveX - осуществить его поставку на Web-сервер вместе с библиотекой dbclient.dll и протестировать, обратившись к сгенерированной HTML-странице (рис.5).

Рис.5. Тестирование "тонкого" клиентского приложения

Отметим также, что можно обратиться к созданной HTML-странице с другого компьютера локальной сети.

Следует обратить внимание на то, что web-сервер вовсе не обязан находиться на том же самом компьютере, что и сервер приложений. Назначение web-сервера в такой информационной системе весьма утилитарно - он представляет собой лишь хранилище, откуда можно сгрузить ActiveX, и ничего более. Если говорить о передаче ActiveX через Internet и доступе к серверу приложений через Internet, то в принципе возможна ситуация, когда сервер приложений, Web-сервер и пользователь находятся в трех разных городах, пользователь обращается к своему провайдеру в своем городе, а как именно он получает ActiveX и обращается к серверу приложений, его как пользователя Internet интересовать не должно.

Нередко при создании таких ActiveX задаются вопросы типа: "А если приложение должно содержать несколько форм, как мне поступить?". Ответ достаточно прост: в этом случае возможна генерация дополнительных форм динамически при наступлении какого-либо события (например, нажатия на кнопку). Нужно только не забыть уничтожить созданную форму, когда она станет не нужна. Следует также помнить, что библиотека OCX, содержащая ActiveX, содержащий, в свою очередь, несколько форм, будет иметь больший размер, чем в случае ActiveX c одной формой.

Довольно часто встречаются вопросы, связанные с тем, что ActiveX не отображается в броузере. Вот одно из таких писем:

Здравствуйте, Наталия Елманова!

Прочел вашу статью на Web-сайте www.interface.ru. Пробую создать сервер приложений с использованием ActiveForm. После создания и переноса CAB-файла и Web-страницы на Web-сервер страничка открывается, но на месте предполагаемой аппликации только квадратик. Разъясните, в чем может быть проблема.

С уважением Слава. Slava@ruslan-com.ru

Причин такого поведения может быть несколько. Первая причина связана с тем, что далеко не все броузеры поддерживают отображение ActiveX с помощью тега <OBJECT>. Для отображения ActiveX следует использовать только MS Internet Explorer версии 3.0 и выше. Вторая причина может быть связана с настройкой уровня безопасности броузера. В случае максимального уровня безопасности (значение, принятое по умолчанию) никакой исполняемый код (в том числе и ActiveX) в броузере не выполняется, а некоторые версии Internet Explorer не только не сгружают ActiveX и тем более не выполняют его, но при этом еще и ничего не сообщают пользователю. Есть еще и третья причина - настройки операционной системы компьютера пользователя таковы, что пользователю запрещено изменять реестр, и в этом случае ActiveX в нем, естественно, не зарегистрируется.

**Поставка "тонких" клиентов**

Как уже неоднократно отмечалось, поставка "тонких" клиентских приложений представляет собой намного более простой процесс, чем поставка "толстых" клиентов. Для функционирования такого клиента требуется, помимо исполняемого файла, наличие библиотеки dbclient.dll (ее можно найти в каталоге Windows\System на компьютере разработчика и скопировать либо в аналогичный каталог компьютера пользователя, либо в каталог, содержащий исполняемый файл клиентского приложения). Если речь идет о создании стандартного дистрибутива с помощью InstallShield Express или иного аналогичного средства, следует отказаться от включения BDE и SQL Links в его состав, а включить лишь файлы, необходимые для функционирования клиентского приложения, и библиотеку dbclient.dll.

Если клиентское приложение выполнено в виде иcполняемого в броузере ActiveX, следует выбрать dbclient.dll в закладке Deploy Additional Files диалоговой панели Web Deployment Options. При распространении ActiveX через Internet рекомендуется выбрать также опцию "Use CAB file compression" для уменьшения времени передачи его по сети.

**Поставка серверов приложений**

Если с поставкой "тонких" клиентов все более или менее ясно, то поставка серверов приложений и связанные с ней проблемы вызвали самый большой наплыв вопросов.

Нередко разработчики, создавшие сервер приложений и успешно протестировавшие его на компьютере, на котором производилась разработка, сталкиваются с трудностями при переносе этого сервера приложений на другой компьютер - он отказывается там выполняться либо работает некорректно. Вопросы по этому поводу обычно содержат описание проблемы и формулируются примерно следующим образом: "Что должно быть установлено на компьютере, на котором эксплуатируется сервер приложений?". Например, Георгий Буриков и Михаил Шунин из ВЦ ОАО "Аэрофлот-Российские Международные авиалинии" спрашивают в своем письме:

Мы пытаемся установить удаленное соединение с базой данных, используя компонент TMidasConnection. На серверной части (WindowsNT) работает серверное приложение - TProvider, TQuery и TDataBase в RemoteDataModule и главная форма, показывающая счетчик клиентов (аналог примера из DEMOS\ACTIVEFM\EMPEDIT\Server.dpr из поставки Delphi 3.01).

В директории с приложением-сервером находится программа scktsrvr.exe (она запускается как приложение ) и dbclient.dll. Приложение-сервер запущено.

Клиентское приложение содержит DataModule c ClientDataSet, Query и MidasConnection и главную форму с DBGrid, настроенным на Query. Свойства MidasConnection выставлены следующим образом:

ComputerName - IP-адрес сервера,

ConnectType - stSockets

ServerGuid - {56791B61-7625-11D1-A3B8-00C0DF817EF4} - (взято из текста

приложения сервера)

ServerPort - 211

UserBroker - False

При установке свойства Connected компонента MidasConnection равным true счетчик клиентов в приложении-сервере увеличивается на 1 и при сбросе Connected в false уменьшается. Т.е. вроде бы приложение-сервер "чувствует" подключение приложения-клиента. Однако при установке ClentDataSet.Active равным true возникает ошибка - "Ошибка при загрузке библиотеки" без дополнительных пояснений. О какой библиотеке идет речь?

С уважением, Михаил Шунин, Георгий Буриков ВЦ ОАО "Аэрофлот-Российские Международные авиалинии" burikov@hotmail.com shunin@aha.ru

Речь идет об одной из библиотек, которые входят в состав MIDAS и должны находиться на сервере приложений. Это библиотеки IDPROV32.DLL (она должна быть в том же каталоге, что и BDE), DBCLIENT.DLL и STDVCL32.DLL (эти библиотеки должны быть в каталоге Windows\System и должны быть зарегистрированы в реестре).

В целом же на этот и другие подобные вопросы можно ответить следующее. На компьютере, где выполняется сервер приложений, должен быть установлен MIDAS. В его состав входит BDE, SQL Explorer (ConstraintBroker Manager), вышеупомянутые библиотеки, а также OLEnterprise. В состав Delphi 3 Client/Server Suite входит только MIDAS Development Kit, включающий лицензию разработчика и ограниченную версию MIDAS, позволяющую создавать серверы приложений, тестировать их, но не позволяющую заниматься их поставкой. Для поставки и промышленной эксплуатации серверов приложений, созданных с помощью Delphi 3, требуется полная версия MIDAS, предоставляющая, наряду с необходимыми библиотеками и утилитами, право эксплуатации любого количества серверов приложений на том компьютере, где она установлена. Естественно, комплектов MIDAS нужно столько, сколько в сети имеется компьютеров с поддерживаемыми MIDAS серверами приложений. Эксплуатация серверов приложений без наличия полной версии MIDAS на компьютере, где они установлены, является нарушением лицензионного соглашения Delphi.

Некоторые другие вопросы

Посмотрел Вашу статью: http://www.interface.ru/Borland/MIDAS1/d3ap.htm. Очень понравилось. Не могли бы Вы уточнить, как вручную отредактировать реестр (какие ключи добавить или отредактировать), чтобы OLE-объект воспринимался не как локальный, а как удаленный? К сожалению, из статьи я этого не смог понять.

С уважением Алексей Гудков, ОАО ИК "РИФ" AGudkov@rif.nsk.su

В принципе это делается с помощью утилиты DCOMCNFG (в Windows NT 4 она есть, а для Windows 95 можно найти и эту утилиту, и сам клиент DCOM на Web-сервере Microsoft). Но если осуществлять соединение клиента и сервера с помощью DCOM, сервер при этом не сможет работать под управлением Windows 95. Кроме того, требуется на странице Access Control раздела Network в Control Panel выбрать опцию User Level Access Control, что отличается от установок, принятых по умолчанию, а также экспортировать с первичного контроллера домена сведения о пользователях, а затем описать, кому из них Вы разрешаете этот сервер запускать. Естественно, в сети при этом обязательно должен быть первичный контроллер домена.

С помощью OLEnterprise (в частности, Object Explorer) все это сделать проще, так как в этом случае наличие первичного контроллера домена не обязательно, экспорт имен тоже не требуется, и сервер может работать под управлением Windows 95. Чтобы сервер воспринимался как удаленный, в раздел компьютера-сервера HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes\CLSID\{<CLSID Вашего сервера>} добавляется подраздел OLEnterpriseExport, а в раздел компьютера-клиента HKEY\_CLASSES\_ROOT\CLSID\{<CLSID Вашего сервера> } добавляется раздел Dap\DCEClient\<имя выполняемого файла>.<имя OLE-сервера> c набором параметров, представленных на рис. 6.

Рис.6 Раздел Dap\DCEClient\ реестра клиента при импорте объекта

Уважаемая Наталия Елманова!

Прочитав Вашу статью, я попробовал создать приложение-сервер, и у меня возник вопрос - как установить связь master-detail в удаленном модуле данных без компонентов DataSource?

Буду очень вам признателен за ответ.

С уважением Денис Бокатый den-nbk@usa.net

Можно создать модифицируемый SQL-запрос к обеим таблицам, содержащий предложение WHERE, и поместить его в удаленный модуль данных вместо компонентов TTable. Кроме того, можно поместить компонент TDataSource, используемый для связи таблиц, в обычный модуль данных или на форму. Иногда также можно связь master/detail установить не на сервере приложений, а в клиенте (при небольших объемах таблиц это может быть даже выгоднее с точки зрения производительности). Отметим, что наиболее общепринятый способ установки такой связи подразумевает описание ее в словаре данных сервера приложений с помощью SQL Explorer (ConstraintBroker Manager), входящего в состав MIDAS, с целью передачи сведений о ней клиентскому приложению.

Здравствуйте, Наталия!

Большое спасибо Вам за внимание, которое Вы нам уделили. Ваши советы нам помогли. ... Думаю, что было бы хорошо "обозреть" имеющиеся у пользователей Delphi средства для создания приложений для Internet, работающих с базами данных. По моему разумению, можно выделить три группы средств:

"Родные" средства Delphi с палитры компонентов Internet - позволяют работать с любым www-сервером под управлением Windows NT или Windows 95, но требуют знания HTML и неудобны при отладке приложений (может быт, ь в этом я и не прав, так как сам этот способ не реализовывал).

Разработка с использованием библиотек Baikonur от Epsylon Technologies - они замечательны, удобны в разработке и отладке, не требуют (не затрагивая сложных случаев) знания HTML , но необходима установка сервера Baikonur, что возможно, чаще всего, в корпоративной сети, а провайдеры неохотно идут на установку широко не известного (к сожалению) продукта.

Разработка с использованием ActiveForm и многозвеннных приложений - это очень удобно в разработке, отладке и сопровождении, не требуется (не затрагивая сложных случаев) знание HTML, но при работе в "медленном" Internet не каждый пользователь дождется окончания загрузки ActiveX. Думаю, что более подробный анализ этих средств поможет сориентироваться многим разработчикам на Delphi - ведь переход к переносу приложений в Internet, пожалуй, неизбежен.

С уважением, Михаил Шунин начальник отдела программирования КИВЦ ОАО "Аэрофлот" shunin@aha.ru, www.aha.ru/~gemis

Разработка web-приложений, связанных с динамической генерацией HTML-страниц - это действительно интересная и важная тема, которой, по-видимому, будет посвящена одна из ближайших статей, посвященных продуктам Borland.

Что касается сервера Baikonur от Epsylon Technologies - это действительно интересный продукт, наглядно демонстрирующий, какие возможности предоставляют разработчикам технологии Borland, использованные при его создании. Нельзя не отметить, что этот продукт является, по существу, сервером приложений, предоставляющим возможность их визуально создавать. Своей технологической направленностью этот продукт заметно выделяется среди отечественных программных продуктов, ориентированных в основном на национальную российскую специфику (не секрет, что это главным образом бухгалтерские, финансовые и офисные приложения, юридические базы данных, игры, мультимедиа-энциклопедии, OCR, различные программы лингвистического назначения - достаточно посетить очередной SofTool, чтобы в этом убедиться). Остается только пожелать удачи компании Epsylon Technologies, разработавшей этот интересный продукт и надеяться на то, что авторы Baikonur смогут написать о нем и использованных при его создании технологиях более подробно.

Что касается использования ActiveForm и многозвенных приложений - я надеюсь, что настоящая статья в определенной степени выполняет это пожелание.

Несколько слов для пользователей C++Builder.

Все изложенное в данной статье (и в других статьях, посвященных серверам приложений и MIDAS) в равной степени (с точностью до написания кода, которого, по существу, не так уж и много) относится и к С++Builder 3.0. В новую версию C++Builder включена поддержка MIDAS на уровне компонентов (имеются компоненты TClientDataSet, TRemoteServer, TMidasConnection, TProvider), эксперты для создания удаленных модулей данных и ActiveX, и имеется MIDAS Development Kit для создания серверов приложений. Отличие от Delphi заключается в том, что для создания ActiveX используется библиотека Active Template Library (ATL), тогда как Delphi использует собственные средства создания ActiveX. Одновременная разработка сервера приложений и клиента в C++Builder осуществляется даже более просто, чем в Delphi, благодаря новым средствам управления проектами, позволяющим загрузить в среду разработки несколько проектов одновременно. Отметим также, что с помощью C++Builder Enterprise возможна разработка "тонких" клиентов для Borland Entera. Таким образом, C++Builder, наряду с Delphi 3, теперь является инструментом, позволяющим создавать серверы приложений и "тонкие" клиентские приложения, в том числе в виде выполняемых в web-броузере элементов ActiveX.

В заключение хотелось бы поблагодарить авторов многочисленных (как процитированных, так и многих других) писем, пришедших после публикации статьи о серверах приложений, свидетельствующих о том, что данная тема представляет определенный интерес для читателей.