

А. ШЕВЕЛЁВ

DbfWebServer

СПОСОБ
ЭФФЕКТИВНОЙ РАБОТЫ
С ТАБЛИЦАМИ DBF
В СРЕДЕ ИНТЕРНЕТ

А. Шевелёв
DbfWebServer. Способ
эффективной работы
с таблицами DBF в среде
Интернет

http://www.litres.ru/pages/biblio_book/?art=24154062
ISBN 9785448522871

Аннотация

Данная книга предназначена для всех, кто трепетно относится к накопленным данным и хочет их сохранить и приумножить. Предлагается достаточно простой способ создания информационных систем различного назначения.

Содержание

Всё о сервере http://ixbase.info	5
Введение	7
Глава 1. Эффективный сервер для таблиц DBF	9
Установка сервера	13
Запуск сервера	14
Файл инициализации	16
Глава 2. Особенности программирования	20
Дополнительные функции	20
Немного истории	22
Динамический загрузчик	22
Глава 3. Управление сервером	27
Основные операции	27
Редактирование	29
Запросы к серверу	30
Управляемые списки	34
Блоки кода	35
Передача данных	35
Реализация спецификаций	37
Семейство переменных	39
Построение формы	39
Конец ознакомительного фрагмента.	49

DbfWebServer
Способ эффективной
работы
с таблицами DBF в среде
Интернет

А. Шевелёв

© А. Шевелёв, 2017

ISBN 978-5-4485-2287-1

Создано в интеллектуальной издательской системе Ridero

Всё о сервере <http://ixbase.info>

Благодарности

Предлагаемый подход появился благодаря неустанным разработкам по совершенствованию и поддержке стиля xBASE со стороны xHarbour (<http://xharbour.com>) – разработчика программного обеспечения (ПО), ставшего достойным преемником великого наследия CLIPPER.

Особая благодарность владельцам авторского права (Массачусеттский технологический институт, Национальный институт исследований в области информатики и автоматизации, Университет Keio) за прекрасную документацию HTML

Девиз:

сервер, который всегда с тобой и с каждым днем лучше

Статические страницы могут быть превращены в динамические с помощью нового сервера, который поддерживает работу с базой данных. Под базой данных здесь подразумевается набор таблиц типа *.dbf.

Такой способ построения динамических страниц открывает новые возможности для формирования информационных ресурсов, настольных приложений и передачи данных.

Данная книга предназначена для всех, кто трепетно отно-

сится к накопленным данным и хочет их сохранить и приумножить. Предлагается достаточно простой способ создания информационных систем различного назначения

Введение

Работа с таблицами типа *.dbf пока не представлена должным образом в среде Интернет. Это связано, наверное, с повальным увлечением технологией SQL. Считается, что таблицы DBF – это устаревший подход и будущее только за технологиями типа сервер-клиент. Тем не менее, были попытки (и они не прекращаются) сблизить технологии Интернет и системы управления базами данных – СУБД (1,2). Эта разработка является также инструментом, способствующим насыщению Интернет массивом данных, хранящихся в базах данных. Теперь статические страницы становятся динамическими с легкостью, присущей системам управления данных типа CLIPPER. Работа с таблицами DBF проста и интуитивно понятна.

Таблица представляет собой набор колонок (полей) по вертикали и строк (записей) по горизонтали. Таким достаточно простым и наглядным образом формируются и собираются однотипные записи в списки, которые могут быть очень большими. Допустимый размер таблиц типа *.dbf намного превышает размер таблиц типа *.exl, используемых в программе Excel, и может достигать 3 млн записей, а СУБД DBF намного проще и легче системы SQL. На сегодняшний день именно в этих таблицах хранится достаточно много данных, которые не должны лежать мёртвым грузом или

быть утеряны. Свободно обратиться к данным, хранящимся в таблицах *.dbf, организовать передачу данных между информационными узлами или приложениями позволяет новый сервер dbfWebServer (рис. В.1).



Рис. В.1. Экранная форма программы dbfWebServer
Список сокращений

HTTP – HyperText Markup Language

VM – виртуальная машина

WWW – World Wide Web

DOS – дисковая операционная система

ИТ ixBASE – информационные технологии ixBASE

ОС – операционная система

ПО – программное обеспечение

ПО ixBASE – программное обеспечение ixBASE

СУБД – система управления базами данных

Глава 1. Эффективный сервер для таблиц DBF

Сервер, который всегда с тобой

DbfWebServer – это сервер, который всегда с тобой: простой, мобильный и компактный. Каждый день он пополняется новыми данными и сервисами и улучшается. Сервер очень удобен тем, что его можно запускать и на локальном компьютере. Для этого предусмотрены специальные режимы настройки сервера. Эти настройки находятся на стартовой странице сервера. С их помощью запуск сервера можно настроить автоматически при включении компьютера и/или из эксплорера с помощью специальной кнопки, что является весьма полезным и удобным и повышает эффективность работы.

Отличительной особенностью данного сервера является то, что для полноценной работы с базами данных достаточно одной программы, в отличие от комплекса SQL, например APACHE + MySQL + PHP. Для такого комплекса необходимо запустить как минимум два сервера – APACHE и SQL – или другую родственную пару серверов.

Скорость работы сервера dbfWebServer, по наблюдению и мнению автора, превосходит другие комбинации, что позволяет создавать быстрый пользовательский интерфейс для

конечного пользователя, используя при этом браузер, использовать разнообразный спектр баз данных, что становится особенно актуальным, когда бизнес-приложения перемещаются в сторону платформы Интернет.

Размер программы составляет порядка 1400 Кб. В программу включены все основные функции CLIPPER, которые необходимы для работы с базой данных, строками, математическими операциями.

Запустить сервер можно с любого носителя, даже флэш-памяти, что также может стать весьма удобным методом работы, если необходимо часто переносить данные. Относительно небольшой объём базы данных, можно сказать мобильный, позволит носить с собой все данные, запускать сервер с любого компьютера и пользоваться этими данными. Такой подход, в свою очередь, позволит помочь при распределённой или удалённой работе, когда необходимо обслуживать или развивать несколько информационных ресурсов или проектов, использующих данные и базы данных.

Устойчивость сервера

Устойчивость работы сервера как программы и не разрушающий индексы таблиц режим работы сервера с базой данных является важным моментом. Для таблиц типа *.dbf всегда было достаточно трудно подобрать устойчивый сервер. В данном случае речь идёт об операционной системе (ОС). в этом отношении лучше всего показал себя сервер NOVELL. В других операционных системах при многополь-

зовательском режиме работы достаточно часто наблюдается разрушение индексных файлов. Этот фактор в значительной мере тормозит использование таблиц в других конфигурациях не NOVELL и мешает широкому распространению приложений на основе баз данных DBF.

По отношению к операционной системе программа dbfWebServer является однопользовательской, а потребление информации имеет, наоборот, многопользовательский характер. Для такого случая следует отметить, что семейство операционных систем WIN32 работает более устойчиво.

Сервер dbfWebServer в своей работе для организации многопользовательского режима использует систему сокетов, а с сокетами ОС WIN32 работают надёжнее, нежели с системой разделения файлов между приложениями.

Программа dbfWebServer может работать параллельно с другими программами, обеспечивая одновременный разделённый доступ к таблицам и индексам.

Сервер обрабатывает ошибки, возникающие при интерпретации блоков кода, и возвращает соответствующие сообщения, поэтому пользователю или разработчику легко отлаживать скрипты, которые запускаются на сервере. Это, правда, не относится к бесконечным циклам. Здесь может возникнуть проблема.

Сервер построен в каскадном стиле, без использования потоков, поэтому прервать бесконечный цикл без перезагрузки сервера нет возможности.

С точки зрения безопасности и устойчивости работы сервера отладку скриптов лучше всего осуществлять в локальном режиме, на локальном сервере, а потом переносить на удалённый сервер.

Сама по себе работа СУБД CLIPPER с базами данных проверена временем и является самой надёжной на сегодняшний день.

Состав сервера

Состав сервера определяет следующий основной перечень файлов и директорий, входящих в состав поставки (дистрибутив) сервера dbfWebServer:

`\www\` – рабочая директория сервера

`\www\dbf\` – поддиректория с таблицами *.DBF

`\www\html\` – поддиректория, в которой находятся исходные скрипты HTML

`\www\HTML4\` – поддиректория с документацией HTML

`\www\xHarbour\` – поддиректория, в которой находится документация xHarbour

`dbfWebServer.exe` – программа

`dbfWebServer.ini` – файл инициализации

`dbfWebServer.txt` – первичный список таблиц

В директории с таблицами представлены только основные таблицы, необходимые для начала работы и понимания сути работы сервера. Пользователь или разработчик может самостоятельно расширять этот список таблиц исходя из задач, которые ему предстоит решать. Способ формирования спис-

ка таблиц или проекта будет обсуждён далее.

\dbf – директория с таблицами

DBFHTM00.CDX

DBFHTM00.DBF – блоки кода для HRB. exe

DBFRUN00.CDX

DBFRUN00.DBF – блоки кода общего назначения

DIRECT00.CDX

DIRECT00.FTP

DIRECT00.DBF – для списков директорий

SPIDBF00.CDX

SPIDBF00.DBF – проект базы данных: список таблиц

SPIDBF10.CDX

SPIDBF10.DBF – проект базы данных: список полей

SPIDBF20.CDX

SPIDBF20.DBF – проект базы данных: список индексов

Директория \www\ добавлена с точки зрения совместимости с другими серверами, например Apache.

Примечание. От версии к версии состав сервера может меняться.

Установка сервера

Загрузить из Интернета архивный файл dbfWebServer.zip

Распаковать архивный файл или скопировать содержимое с инсталляционного диска на свой локальный жёсткий диск.

Программа не требует процедуры установки.

Если установка сервера производится с инсталляционным-го диска, то следует просто скопировать содержимое CD-диска на локальный жёсткий диск.

Рекомендуется скопировать всё в корневой каталог.

Запуск сервера

Сервер запускается из командной строки набором имени файла

`dbfWebServer.exe`

или кликом мыши по этому имени в проводнике.

После запуска появится окно с программой, как показано на рис. В.1.

После того как программа запустится, в адресной строке браузера введите и активизируйте следующий адрес:

`http://localhost:4180`

по умолчанию будет загружена страница `index.html`:

`http://localhost:4180/html/index.html`

Примечание. Загрузку стартовой страницы необходимо обеспечить как минимум для двух случаев:

- когда сервер устанавливается параллельно существующему другому серверу. это будет рассмотрено ниже (привязка сервера);
- когда сервер устанавливается самостоятельно.

Если сервер устанавливается параллельно, то использует-

ся файл dbf.html, чтобы не пересекаться с существующим сервером.

В любом случае по умолчанию загружается файл index.html из директории /html/.

После загрузки страницы index.html в браузере должна появиться картинка следующего содержания (рис. 1.2):



Рис. 1.2. Вид загруженной страницы index.html из сервера dbfWebServer

Для успешного старта программы и дальнейшей продуктивной работы с таблицами программе требуется два служебных файла.

Один файл – это файл инициализации, другой – текстовый файл, содержащий минимальный перечень используемых

мых таблиц. Второй файл можно назвать стартовым файлом проекта.

Файл инициализации содержит инициализацию системных переменных и набор настроек.

Файл проекта содержит в основном системные таблицы, которые требуются для работы сервера и входят в состав дистрибутива.

Файл инициализации

Запуск программы dbfWebServer.exe начинается с загрузки стартового файла dbfWebServer.BEG.

Инициализация программы (а это для нее ответственный момент) производится с помощью специального файла. Это делается для того, чтобы объявить и инициализировать глобальные и системные переменные вне программы, предоставив пользователю возможность тонкой настройки, если такая необходимость возникнет.

В файле инициализации устанавливаются предварительные настройки сервера и базы данных. Эти настройки можно редактировать или дополнять. Редактировать настройки можно с помощью любого редактора. Изменять что-либо желательно только после того, как вы хорошо изучили предмет и понимаете смысл и назначение установок. Начальное содержание этого файла следующее:

Файл dbfWebServer.BEG

```
Set (01,«OFF»),Set (03,4 ),Set(04,"dd.mm.yy»),Set  
(08,«ON»), Set (09,«ON»)
```

```
Set (11,«ON»),Set (14,10),Set (31,.F.),Set (32,«OFF»),Set  
(35,«ON»)
```

```
ixPUB («arrDbfRun», arRay (512,2))
```

```
ixPUB («arrDbfOpe», arRay (64))
```

```
ixPUB («_DBF_», «STA»)
```

```
ixPUB («M», «001»)
```

```
ixPUB («ixPORT», 4180)
```

Список наиболее значимых переменных для программы, объявленных и инициализированных в стартовом файле программы, следующий:

arrDbfRun

– массив, предназначенный для блоков кода, готовых к выполнению

arrDbfOpe

– массив, предназначенный для хранения списка открытых файлов. Эта переменная является элементом системы динамического открытия файлов

ixPORT

– номер порта, на котором программа слушает и принимает сообщения

Стартовый файл проекта

Файл, содержащий список таблиц, называется dbfWebServer.txt

Этот файл используется динамическим загрузчиком. Назначение этого файла и формат записей будут рассмотрены ниже. Здесь следует отметить только то, что сервер может стартовать без этого файла, но в дальнейшем обращение к таблицам без проекта таблиц в последующей работе будет затруднительно, можно использовать только простые страницы, без обращения к таблицам.

Привязка сервера

Иногда сервер dbfWebServer будет устанавливаться параллельно с другим сервером. У пользователя уже может действовать какая-либо система работы с данными. В этом случае установка требует нескольких манипуляций в ручном режиме.

Рассмотрим это на примере, когда на компьютере уже установлен комплекс из сервера Apache и базы данных MySQL.

Для сервера Apache и ряда других WEB серверов, как правило, создаётся рабочая директория, содержащая рабочие скрипты, с именем

```
\www\
```

В многопрофильной системе IxBase, в которую входит данный комплекс, используется следующий путь расположения компонент системы:

```
\ixb\bin\www\
```

В директории `\ixb\bin\` находится сам сервер (программа) `Apache.exe`.

В директории `\ixb\mysql\` находится база данных MySQL.

Рабочие файлы дистрибутива сервера `dbfWebServer` копируются в этом случае соответственно в директорию `\ixb\bin\`.

Содержимое директории `\www\` сервера `dbfWebServer` копируется в директорию `\ixb\bin\www\` уже установленного, например, сервера многопрофильной системы `ixBase`.

Глава 2. Особенности программирования

Главной особенностью является использование блоков кода. Синтаксис блоков кода несколько отличается от синтаксиса исходного кода программы в стиле xBASE. В блоках кода отсутствует возможность использования итераторов WHILE, FOR. Вместо этого используются итераторы по базе данных или массиву dbEval () и aEval (). В блоке кода функции объединяются в группы в явном или неявном виде, то есть представляют собой одну строку, в которой операторы разделены запятой, например

```
DBFRUN00-> (dbGoGop (),FieldGet (1))
```

Дополнительные функции

Сервер имеет свою специфику, поэтому разработан ряд дополнительных функций, предназначенных для более эффективной работы с сервером.

ixPUB (cArg1,value) – объявление глобальной переменной. Первый аргумент задаёт имя переменной в виде строки. Вторым аргументом функции – значение, присваиваемое переменной. Значение может быть любого допустимого типа.

ixWhile (cKey, cValue) – итератор, используемый в блоках кода. Первый строковый аргумент задаёт условие цикла, вто-

рой строковый аргумент определяет блок кода (функцию).

`dbfRun (cName)` – функция, запускающая блок кода на выполнение. Может вызываться из блока кода. Работает совместно с таблицей `DBFRUN00`.

`DbfOut (bLine, RelNam, RelVal, KeyNam, KeyVal, Filter, PrUni, MaxEle)` – функция, предназначенная для вывода списка из базы данных согласно заданным параметрам. Представляет собой альтернативу функции `dbEval ()`, в некоторых случаях позволяет добиться лучших результатов.

`_x_sta (cStr_1,cStr_2,cStr_3)` – функция, предназначенная для вывода сообщений в окне программы.

`encode (cString)` – функция перекодировки данных, передаваемых по каналу Интернет.

Системные таблицы сервера

Основным постулатом системы является то, что весь исходный код хранится в базе данных. Исходный код заносится в виде блоков кода, совместимых с системой `CLIPPER 5.3`.

Чтобы разделить управление между различными программами реализациями информационных технологий `ixBASE`, используются следующие системные таблицы:

`DBFHMTM00`. Используется CGI-скриптами `HRB. EXE` (Harbour), `CLP. EXE` (`CLIPPER 5.3`), `ALS. EXE` (Alaska). Скрипты работают под управлением `WIN32`. Для операционной системы `UNIX` скрипт находится в разработке, желающие могут скомпилировать его самостоятельно.

`DBFRUN00`. Используется программой `IXBASE. EXE`,

созданной системой xHarbour. Работает под управлением WIN32 и WIN64.

DBFBLO00. Используется программой IXBASE. EXE (CLIPPER 5.3, FiveWin 1.95). Работает под управлением WIN32.

DBFMEN00. Таблица функциональных клавиш. Используется программой IXBASE. EXE (CLIPPER 5.3, FiveWin 1.95).

Все таблицы для вышеперечисленных программ находятся в директории

`/ixb/sys/dfw/`

В укороченной версии программы эти правила не используются, все таблицы находятся в директории

`./dbf/`

Немного истории

Для программы IXBASE. exe (DOS) использовалась директория `/ixb/sys/dbf/`.

В связи с тем, что в идеологии системы ixBase произошли большие изменения, пришлось разделить таблицы базы данных на две группы: `/DFW` и `/DBF`.

Динамический загрузчик

Программа открывает таблицы с помощью динамическо-

го загрузчика. Список таблиц, входящих в проект, представляется в двух видах:

первый вид – это текстовый файл dbfWebServer.txt;

второй вид – это список таблиц: SPIDBF00.dbf, SPIDBF10.dbf, SPIDBF20.dbf (рис. 2.1).

Таблица SPIDBF00.dbf содержит список наименований таблиц проекта.

Таблица SPIDBF10.dbf содержит список полей таблиц.

Таблица SPIDBF20.dbf содержит список ключей сортировок таблиц.

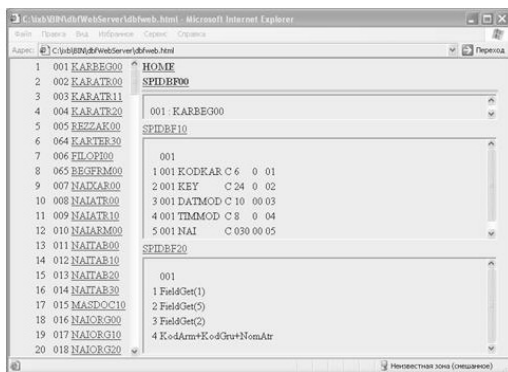


Рис. 2.1. Экранная форма результата запроса к серверу dbfWebServer

Порядок работы динамического загрузчика

Динамический загрузчик построен на обработке ошибки, возникающей при обращении к таблице, которой нет в рабочей области системы CLIPPER (xHarbour), код ошибки subCode = 1002.

Эта ошибка возникает, как было сказано, при попытке обращения к таблице, которая ещё не открыта. Это может быть, например, в следующих конструкциях:

- SPIDBF00-> (dbGoTop ())
- dbSelectArea («SPIDBF10»)

или других конструкциях, используемых как в исходном программном коде, так и блоках кода.

При возникновении ошибки программа обрабатывает эту ситуацию следующим образом.

Сначала сканируется текстовый файл dbfWebServer. dbf. Если запрашиваемая таблица есть в списке этого файла, то она открывается с помощью функции ixOpen (cName). Имя этой таблицы заносится в массив arrDbfOpe. Если таблица отсутствует физически, то она сначала создаётся и затем открывается. Все необходимые атрибуты для этого случая должны находиться в списке в специальном формате, как приведено в файле dbfWebServer. dbf для определённых в нём таблиц.

Формат этого файла следующий.

Таблицы разделены специальной строкой из звёздочек —

Первая строка раздела описывает саму таблицу: путь

к таблице, имя таблицы, алиас таблицы, список полей таблицы.

Следующие строки представляют описание индексов таблицы.

Сколько индесов, столько строк (по одной строке на индекс).

Если требуемая таблица не найдена в текстовом файле, динамический загрузчик производит поиск в базе данных (таблице) SPIDBF00.dbf. Если поиск успешен, то он открывает запрашиваемую таблицу и заносит имя таблицы в массив активных таблиц `arrDbfOpen`.

Если массив активных таблиц переполняется, то первая таблица удаляется из массива и закрывается.

Размер массива задаётся в файле инициализации программы.

Текстовый файл для удаленного управления недоступен, поэтому список таблиц можно и следует хранить в проекте базы данных, который содержится в триаде таблиц SPIDBF00, SPIDBF10 и SPIDBF20 (Рис. 2.1).

Правило формирования наименований таблиц в системе ixBASE

Наименование таблиц формируется, как правило, из двух слов. Из каждого слова выбирается три буквы. Итоговое слово из шести букв дополняется порядковым номером из двух цифр.

SPIDBF00 ->

SPI – список

DBF – dbf таблиц

00 – стартовый

$(SPI+DBF+00) \Rightarrow SPIDBF00$

Три таблицы, логически объединённые таким образом одним именем, образуют фолдер SPIDBF (00,10,20).

Глава 3. Управление сервером

Основные операции

Сервером можно управлять непосредственно из браузера. Основные операции управления сервером следующие: показать сервер (show), спрятать сервер (hide), закрыть программу (shutdown). Основные операции представлены на рис. 3.1. Эти операции выполняются с помощью стандартной функции WIN API из адресной строки браузера, например

- `http://localhost:4180/ShowWindow(hWnd,5)` – показать окно программы;
- `http://localhost:4180/ShowWindow(hWnd,0)` – спрятать окно программы;
- `http://localhost:4180/sendMessage(hWnd,2,NIL,NIL)` – закрыть программу.



Рис. 3.1. Экранная форма стартовой страницы dbf.html с элементами управления сервером dbfWebServer

Документация

Документация представлена тремя ссылками:

- Manuel of dbfWebServer;
- Manuel of HTML4;
- xHarbour functions.

ссылка «Manuel of dbfWebServer» представляет данный вариант документа в электронном виде.

ссылка «Manuel of HTML4» переводит пользователя в документацию для разметки текста в среде Интернет HTML4 на русском языке в оригинальном представлении.

по ссылке «xHarbour functions» осуществляется переход к информационному блоку, который предоставляет доступ

к документации xHarbour относительно функции в файловом виде.

Редактирование

Режим редактирования предназначен для работы со скриптами на расстоянии. Скрипты хранятся в виде файлов типа *.html. Эти файлы можно редактировать обычным редактором.

Другой способ редактирования этих файлов предлагается осуществлять через интерфейс системы. Переход к этому режиму работы находится по ссылке «HTML editor».

Если при входе в форму редактирования файлов список файлов оказывается пустым, следует кликнуть по заголовку таблицы, и список заполнится перечнем имён файлов, которые находятся в директории

/html/*.html

Количество записей в списке регулируется с помощью элементов управления «-» и «++». Нажимая тот или иной элемент, можно добиться того количества записей в списке, которое наилучшим образом соответствует размеру зоны вывода.

Размер поля редактирования регулируется элементами управления по вертикали «-x++» и по горизонтали «-y++».

Сохранить введённые значения можно, нажав кнопку,

на которой стоит знак «плюс».

Если требуется изменить интерфейс, можно изменить код соответствующих скриптов.

Запросы к серверу

Вот мы и подошли к сути задачи. А суть эта состоит в том, что серверу нужно посылать определённые запросы и получать необходимую информацию, запрошенную у сервера.

Здесь необходимо напомнить, что запросы к серверу строятся в виде блоков кода. Это обусловлено тем, что используется СУБД CLIPPER-xHarbour, которая ориентирована в данном случае на работу с блоками кода.

Например, следующий запрос к серверу, использующий свойства блока кода, будет выглядеть следующим образом:

```
src="http://localhost:4180/cb:='<table width=100%>»,  
i:=1,DBFRUN00-> (dbgotop (), ixWhile (>! eof ())», 'cb+= [<tr>  
<td>] + str (i) + [<td>] +FieldGet (1) + [<td>] +FieldGet (2) +  
[<td width=100%>] + STRTRAN (FieldGet (3), [<, []],dbskip  
( ),i++>)), cb»
```

Данный запрос заставляет сервер строить ответ, сразу содержащий непосредственно таблицу, данные и разметку, как показано на рис.3.2 в левой части формы.

Простой запрос в виде математического выражения, на-


```
urlencode (trim ($txt));
$np=strlen ($sp);
$fp=fsockopen ($h,$p,$errno,$errstr,10);
if (!$fp) {
echo «I can't connect to $h»;
} else {
fputs ($fp,«POST $page HTTP/1.0
Accept: image/gif, image/x-xbitmap, image/jpeg, image/
jpeg, application/x-shockwave-flash, application/vnd.ms-
powerpoint, application/vnd.ms-excel, application/msword,
application/x-icq, */*
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0;
Windows 98)
Accept: */*
Referer: http://$h/
Host: $h
Content-Length:». $np.»
Pragma: no-cache

«.$sp);

$reply=«»;
while (!feof ($fp)) $reply.=fgets ($fp,256); fclose ($fp);
echo "$reply»;
```

Примечание. Запросы к серверу из HTML-скриптов передаются с помощью метода GET или POST.

Как было показано выше, более мощный инструмент работы с данными получается при использовании PHP. HTML-скрипты без PHP позволяют создавать сравнительно простые пользовательские формы, которые можно отнести, безусловно, уже к динамическим страницам.

Управляемые списки

Основной задачей создания динамических страниц для наших задач при помощи таблиц является формирование управляемых списков. На это следует обратить особое внимание.

Чаще всего в информационных системах в качестве пользовательской формы строится карточка со списком. Карточка может быть выполнена в виде списка, но сути карточки в данном случае это не меняет. Карточка представляет развернутую информацию об объекте, полный перечень данных об объекте находится рядом со списком. По списку можно перемещаться, выбирая, таким образом, объект для детального просмотра. Карточка представляет информацию о выбранном объекте в нужном виде и объёме и позволяет редактировать эти данные.

Данное общее описание более подробно будет представлено в разделе «Спецификации интерфейса списков v.1.0».

Блоки кода

Отличительной особенностью сервера является то, что используются блоки кода. Это обусловлено тем, что программа создана на платформе CLIPPER – xHarbour. Прежде всего, xHarbour – это виртуальная машина (VM). Такой подход уже использовался в системе управления базами данных (СУБД) CLIPPER, используется в JAVA.

В отличие от виртуальной машины JAVA, которая устанавливается на компьютере клиента и выполняет скрипты, полученные с сервера, виртуальная машина xHarbour работает на стороне сервера и выполняет блоки кода, посылаемые клиентом.

Блоки кода – это безымянные функции. Ссылку на эту безымянную функцию можно присвоить переменной и выполнить её с помощью специальной функции Eval ().

Блоки кода предоставляют прекрасную возможность строить информационные системы с гибкой структурой и различными видами связи и взаимодействия. Именно блоки кода положены в основу сервера dbfWebServer.

Передача данных

Мы были свидетелями парада технологий: майнфрам, клиент-сервер, распределённые сети, интернет. На всех эта-

пах этой череды постоянно возникала проблема передачи данных. Каждый раз она каким-либо образом решалась, но всегда оставалось чувство неудовлетворённости. Не было полного понимания, что такое передача данных. В рамках технологии клиент-сервер это достаточно трудно осуществить, поскольку в такой конфигурации всегда доминирует сервер, от клиента в плане хранения данных мало что зависит.

Новый подход, предложенный в работе [4], и данная разработка дают возможность поставить передачу данных во главу угла создания информационных систем нового поколения, где передача данных становится определяющим условием существования информационных систем.

Из вышесказанного следует, что передача данных в утилитарном смысле для информационных систем – это общение информационных объектов. Передаётся смысл информации, значения данных, а не структура данных, как таковая. Исходя из этого, информационные системы следует строить со слабоформализованными структурами данных, но мощными средствами обработки данных и представления их в естественном виде и передаваемые по стандартным телекоммуникационным каналам.

DbfWebServer станет весомым помощником в этом деле, поскольку учитывает как накопленный опыт, так и вышеописанные спецификации.

Передача данных – это совместный доступ к одной

и той же информации различных приложений.

Передача данных – это однотипный предсказуемый пользовательский интерфейс приложений.

Передача данных – это слабоформализованные структуры данных, обмен только информацией, а не структурой данных вместе с данными.

Передача данных – это распределённые базы данных.

Эти соображения также легли в основу реализации спецификаций, изложенных в седьмой главе этой книги.

Реализация спецификаций

Перейдём к практической стороне вопроса – реализации. Реализация вышеизложенных спецификаций и спецификаций, приведенных в гл. 7 в конкретной информационной системе – задача непростая. Если бы все вопросы были решены, не было бы необходимости в разработках в этом направлении.

В рамках данной разработки будем использовать подход разделения на процессы.

С появлением сервера dbfWebServer стало понятно, что работа с базами данных делится на два процесса. Такое разделение является весьма продуктивным направлением в области обработки данных и построения информационных систем. Попробуем убедиться в этом на конкретном примере.

Один процесс работает с базой данных. Этот про-

цесс – программа dbfWebServer, которая занимается задачей управления базой данных и обработкой запросов от клиентов.

Другой процесс отображает и строит пользовательский интерфейс. Пользовательский интерфейс строится исходя из данных, полученных от сервера, с помощью такого стандартного браузера, как Microsoft Internet Explorer, Netscape, Opera или Firefox.

Эти два процесса согласуются между собой с помощью спецификации HTML. Это обусловлено тем, что используются стандартные браузеры, и передача данных осуществляется в рамках технологии Интернет.

Рассмотрим построение интерфейса на примере рабочего блока, входящего в состав дистибутива.

Построение пользовательского интерфейса картотеки пациентов на базе управляемых списков (рис. 3.3).

Для создания пользовательского интерфейса сначала определим список глобальных переменных, которые будут использоваться в разработке.

Чтобы создать информационную систему или элемент информационной системы априори или исходя из опыта, определяется некий минимальный набор средств:

- объявить и инициализировать глобальные переменные;
- позаботиться о построении пользовательских форм;
- определиться с методологией построения форм;
- обозначить набор правил построения и развития форм.

Семейство переменных

Семейство переменных, которые помогают в управлении пользовательским списком, представлено следующим перечнем:

,ixpub («R_karbeg», 1) – row
,ixpub («L_karbeg», 10) – curele
,ixpub («V_karbeg»,»») – search value string
,ixpub («E_karbeg», 10) – end of list
,ixpub («U_karbeg»,»») – first element of list
,ixpub («N_karbeg»,»») – next element of list
,ixpub («O_karbeg», «2») – order index

Построение формы

Построение пользовательской формы начинается с разметки экрана.

Сначала экран делится на две зоны. Одна зона отводится под основной список, вторая – под карточку и элемент управления списком.

Основная таблица и форма управления имеют, как правило, одинаковое наименование. Так, для таблицы KARBEG00.dbf создаётся одноимённая форма karbeg00.html. Исходный текст этой формы следующий:

```
<eval>
ixpub («R_karbeg», 1)
,ixpub («L_karbeg», 10)
,ixpub («V_karbeg»,»»)
,ixpub («E_karbeg», 10)
,ixpub («U_karbeg»,»»)
,ixpub («O_karbeg», «2»)
,[
<frameset cols=4,6>
<frame                height=100%                width=100%
name=A marginwidth=0 marginheight=0>
<frame                height=100%                width=100%                name=B
src=karbeg002.html>
</frameset>
]
```

Разберём текст скрипта более подробно. В начале скрипта, который строит форму, идет тег `<eval>`, показывающей, что далее следует блок кода, который необходимо выполнить, а не просто отобразить на экране. Далее идет перечень инициализации глобальных переменных, которые будут использоваться при построении пользовательской формы:

```
ixpub («R_karbeg», 1)
,ixpub («L_karbeg», 10)
,ixpub («V_karbeg»,»»)
,ixpub («E_karbeg», 10)
,ixpub («U_karbeg»,»»)
,ixpub («O_karbeg», «2»)
,
```

```
,ixpub («E_karbeg», 10)  
,ixpub («U_karbeg»,»»)  
,ixpub («O_karbeg», «2»)
```

За ним идёт разметка, с помощью которой строятся два фрейма с именами «А» и «В» соответственно.

```
<frameset cols=4,6>  
  <frame          height=100%          width=100%  
name=A marginwidth=0 marginheight=0>  
  <frame          height=100%          width=100%          name=B  
src=karbeg002.html>  
</frameset>
```

текст разметки помещён в строку с помощью дополнительных символов, принятых в системе CLIPPER, – квадратных скобок.

Примечание.

Строковый константы в системе CLIPPER можно задавать с помощью трёх пар символов:

- 1) одинарные кавычки – «это строка текста»;
- 2) двойные кавычки – «это строка текста»;
- 3) квадратные скобки – [это строка текста].

Во фрейме с именем «В» выводится элемент управления списком, который описан в форме karbeg002.html, и карточка.

Файл karbeg002.html

```
<frameset rows=56,* border=0>
  <frame height=100% width=100% name=M
src=karbeg0021.html marginwidth=0 marginheight=0>
  <frame height=100% width=100% name=K
src=karbeg0022.html>
</frameset>
```

Форма karbeg002.html, в свою очередь, строится из двух фреймов. Фрейм с именем «М» содержит элемент управления списком karbeg0021.html.

Фрейм с именем «К» содержит карточку элемента, выделенного в списке

Файл karbeg0021.html

```
<eval>
[<table width=100% cellpadding=0 cellspacing=0> <tr
bgcolor= c0c0c0> <td>
  <b> <a target=A href=naiarm001.html>
  АРМ: ] +HB_oemToAnsi (NAIARM00-> (FieldGet (3))) + [
  <tr bgcolor=e0e0e0> <td> <table> <tr bgcolor=c0d0c0>
  <form name=F target=_top action=rezzak00.html
method=get>
  <td> <input name=kareg onfocus=kS ()
onkeydown=kD(event.keyCode)
onkeyup=kU(event.keyCode)>
```

```
<td> <input type=hidden name=R>
```

```
</td> </form>
```

```
<td> <a target=A href=karbeg001.html? V_karbeg:=«»>
```

```
Home </a> &nbsp;
```

```
<td> <a target=A href=karbeg001.html? kC:=33> PgUp </a> &nbsp;
```

```
<td> <a target=A href=karbeg001.html? kC:=34> PgDn </a> &nbsp;
```

```
<td> <a target=A href=karbeg001.html? maxele - > - </a> &nbsp;
```

```
<td> <a target=A href=karbeg001.html? maxele++> ++ </a> &nbsp;
```

```
</table>
```

```
<tr> <td bgcolor=c0c0c0 height=4>
```

```
<tr> <td height=100% width=100%>
```

```
</table>
```

```
<script> F.karbeg.focus ();
```

```
function kD (kC) {
```

```
  kV=F.karbeg.value;
```

```
  if(F.R.value=='l') {return;} 
```

```
  F.R.value='l';
```

```
    if(kC==38)
```

```
      {parent.parent.A.location.href='karbeg001]+O_karbeg  
      +[.html? R_karbeg - '};
```

```
    else
```

```
      if(kC==40)
```

```
      {parent.parent.A.location.href='karbeg001]+O_karbeg
```

```
+ [.html? R_karbeg++;]
else if(kC==33)
{ parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? kC:='+kC;]
else if(kC==34)
{ parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? kC:='+kC;]
else if(kC==35)
{ parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? kC:='+kC;]
else if(kC==36)
{ parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? R_karbeg:=1»; }
else if (kC==27) { parent.parent.location.href='index.html'; }
}
```

```
function kU(kC){ window.status=kC;
kV=F.karbeg.value;
```

```
if(kC==120){ location.href='imagix0021.html';}
else if(kC>40)
{ parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? kC:='+kC+», V_karbeg:=«»+kV+«"»»; }
else if(kC==8)
{ parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? V_karbeg:=«»+kV+«"»»; }
}
function kS()
{ parent.parent.A.location.href='karbeg001]+O_karbeg
```

```
+['.html'];}
```

```
</script>
```

```
]
```

На что следует обратить внимание здесь? Элемент управления строится с помощью HTML разметки формы ввода

```
<form name=F target=_top action=rezzak00.html  
method=get>  
<td> <input name=karbeg onfocus=kS ()  
onkeydown=kD(event.keyCode)  
onkeyup=kU(event.keyCode)>  
<td> <input type=hidden name=R>  
</td> </form>
```

Для организации механизма управления списком определены функции обработки событий:

- получение фокуса формой `onfocus=kS ()`;
- нажатие клавиши в поле ввода `onkeydown=kD(event.keyCode)`;
- отжатие клавиши в поле ввода `onkeydown=kU(event.keyCode)`.

При нажатии клавиши `<ENTER>` вызывается форма `rezzak00.html` и осуществляется переход к другому списку

со всеми вытекающими отсюда последствиями.

Форма содержит вспомогательное поле R. С помощью этого поля регулируется скорость нажатия клавиш управления списком. Это поле, например, блокирует бесконечное нажатие одной клавиши без перевывода информации на экране.

Поле karbeg является активным, в нем принимаются нажатия всех клавиш и затем соответствующим образом обрабатываются.

Активное поле ввода нужно, в первую очередь, для отслеживания нажатия клавиш с целью:

- передвижения по списку;
- ввода ключевого слова поиска;
- выполнения команды.

Обработка нажатия клавиш осуществляется с помощью кода, написанного на языке JavaScript.

Блок этого скрипта следующий:

```
<script> F.karbeg.focus ();  
function kD (kC) {  
kV=F.karbeg.value;  
if(F.R.value=='l') {return;}  
F.R.value='l';
```

```
if(kC==38)
```

```
{parent.parent.A.location.href='karbeg001']+O_karbeg
```

```

+ [.html? R_karbeg - '];
else if(kC==40)
{parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? R_karbeg++'];
else if(kC==33)
{parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? kC:='+kC; }
else if(kC==34)
{parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? kC:='+kC; }
else if(kC==35)
{parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? kC:='+kC; }
else if(kC==36)
{parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? R_karbeg:=1»}; }
else if(kC==27){parent.parent.location.href='index.html';}
}

```

```

function kU(kC){window.status=kC;
kV=F.karbeg.value;

```

```

if(kC==120){location.href='imagix0021.html';}
else if(kC>40)
{parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? kC:='+kC+», V_karbeg:=«»+kV+«"»}; }
else if(kC==8)
{parent.parent.A.location.href='karbeg001]+O_karbeg
+ [.html? V_karbeg:=«»+kV+«"»}; }

```

```
}  
function kS ()  
{parent.parent.A.location.href='karbeg001']+O_karbeg  
+['.html'];}
```

</script>

Здесь представлена функция обработки нажатия клавиши kD (), отжатия клавиши kU () и получения фокуса kS ().

После того, как данная форма будет выведена на экран, активное поле ввода получит фокус с помощью команды

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.