

Шлюз is2sql в примерах



Содержание

Введение	7
На что это похоже	7
Где взять.....	7
Как это выглядит	7
Как это работает	7
Примеры	7
Роль SQL в создании страниц с использованием шлюза	8
Последовательность чтения	8
Шаблон (файл .htt)	8
Запрос SQL (файл .sql)	8
Файл .par	8
Часть 1. Пути и теги.....	10
Глава 1. Пути	10
Путь /show или /s	10
dbs. Имя базы данных.	10
Обязательный параметр form.	10
user, key. Имя и пароль пользователя СУБД.	10
[family.]parameter1	11
first. Номер первой записи	11
last. Номер последней записи	11
step. Шаг линейки листания.	11
cp. Перекодировка страницы.	11
push. Канал	11
hash, pwd, forminvalid. Вторичный пароль пользователя	11
forminvalid. Форма сообщений об ошибках.	11
Путь /rec или /r	12
Путь /info	12
Путь /setup	12
key. Пароль администратора.	12
newkey. Новый пароль администратора	12
banner, PatternPath, setup, info. Шаблоны страниц.	12
mknull. Замена пустых значений на строковое значение NULL	13
MaxThreads. Максимальное число потоков	13
CacheConnections. Кеширование потоков.	13
Путь /reload	13
Зарезервированные параметры для пути /show	13
form	13
dbs, user, key	13
cp	13
first, last, step	14
hash, pwd	14
forminvalid	14
Глава 2. Теги	15

Глава 3. Теги в html	15
<#parameter> или <#p>	15
Тег <#table> или <#t>	15
Тег <#execute> или <#e>	16
Тег <#metas>	16
Тег <#a>	16
<#a name=sys>	16
<#a name=sys value=..[newvalue=...]>	17
<#a name=IP>	18
Тег <#c>.	18
Копирование параметров	18
Инкремент параметров	18
Создание параметров	18
Передача параметров из формы в форму	19
Тег <#r>	19
Глава 4. Теги в шаблонах .htt	19
Тег <#field>. Вставка полей из таблиц БД	20
replace=emptysame	20
http=encode	20
fnt=currency	20
fnt=money	20
fnt=datetime	20
fnt=hash	20
Специальные поля	20
Внешние форматирующие функции	20
Тег <#X>	21
Другие теги	21
Глава 5. SQL запросы	21
Параметры :\$.. в sql запросах	21
Глава 6. Параметры форм	21
Часть 2. Примеры.....	23
Глава 1. Очень простые примеры (DBDEMOS)	23
1. Отображение html файла	23
2. Отображение таблицы	23
3. Более красивая таблица	23
4. Таблица, не содержащая данных	24
5. Параметры в sql выражениях	25
Пример 6. HTML- форма	25
Замечание к примеру 6	25
Пример 7. Улучшение результата	25
Пример 8. Совместимость с FrontPage	26
Резюме	26
Глава 2. Простые примеры (IBLOCAL)	27
Пример 1. Разделители заголовка и окончания <#@> и <#/@>	27

Пример 2. Разделение больших таблиц на страницы	28
Квалификаторы семейств и их назначение	28
Пример 3. Добавление записи в таблицу	28
Замечания о хранимых процедурах	29
Пример 4. Передача параметров из формы в форму	29
Пример 5. Случай нескольких одинаковых параметров	29
Пример 6. Изменение типа контента.....	32
Путь /record.....	32
Путь /show	32
Глава 3. Безопасность	33
Пример 1. Проверка прав доступа.....	33
Имена и пароли СУБД.....	33
Часть 3. Ресурсы	34
Оптимизация работы	34
Переустановка соединения	34
Число потоков (threads)	34
Алиасы баз данных	34
Алиасы BDE	34
Алиасы для баз данных прямого доступа к SQL серверу	34
Задание алиаса для Interbase Express.....	34
Задание алиаса для NC OCI 8.....	35
Глава 1. Реестр	35
MaxConnections.....	35
NoCacheConnections	35
Banner	35
Dbs	35
User.....	35
Key	35
PatternPath.....	35
Раздел DbAlias. Алиасы баз данных (для не BDE).....	35
TagPrefix	35
SQLCollection	36
Раздел ResponseCustomHeader. Задаваемые заголовки ответов	36
раздел Variables. Параметры по умолчанию	36
DoTransaction. Управление транзакциями.....	36
DbTransisolation. Выбор типа транзакций	36
раздел ReconnectError. Обработка ошибок SQL.	37
DbErrorPrefix	37
Коды ошибок	37
DBConnectFailForm. Форма извещения об ошибке соединения с базой данных по умолчанию	37
ErrorIcon. Иконка ошибки	38
AccessDeniedForm. Форма извещения недостаточности прав для просмотра формы по умолчанию	38
eccFileName, xlatname. Перекодирующие таблицы	38
pagelineXXX. Линейки страниц	38
MkNull. Отладка SQL	39

ValidateQuoteChar	39
OptimizeDoc	39
Раздел ResponseCustomHeader	39
LogFile. LogFunc. LogFuncStart. Журнал активности	39
Формат даты и времени	40
Время актуальности страницы	40
CurrencySimple. Формат простого представления валюты	40
shUniqueID. Ключ шифрации	40
RootPwd Пароль администратора	40
Область действия	40

Часть 3. Замечания по компиляции 41

Соглашение	41
Компилятор	41
Первый запуск	41
Коллекции шаблонов	42
Формат файла коллекции	42
Отладка	42
Символы условной компиляции	42
CGI вместо ISAPI/NSAPI	42

Часть 4. Утилиты 44

Глава 1. Утилита etxtcoll	44
Меню	44
Глава 2. Редактор файлов коллекций перекодировочных таблиц	44
Описание утилиты	44
Формат файла кодировок	45
<Заголовок>	45
<Количество записей N>	45

Часть 5. Примечания..... 47

Глава 1. Использование кириллицы в Interbase	47
Создание базы данных	47
Создание доменов	47
Алфавитный указатель	48



Введение

Шлюз is2sql предоставляет доступ для web- клиентов к большинству известных РСУБД, но особенно- к Interbase.

А, скажете вы еще один шлюз! Сегодня можно выбрать шлюз по вкусу, и это еще один шлюз. Меня он вполне устраивает, может, и вы его к чему-нибудь пристроите.

Плюс в том, что вы получаете полный исходный код, и ежели что не так, можете пройти по нему напильником. К тому же, Вам не надо платить за средства разработки. Для создания сайта, связанного с базой данных, вам нужны только

- FrontPage, HotMetal или вообще любой редактор HTML и
- web- сервер, поддерживающий ISAPI/NSAPI.

Многопоточность ISAPI дает преимущество в скорости обработки запросов на web- сервере, установленном на PC под управлением ОС Windows, и можно увеличить или уменьшить число потоков в кеше.

На что это похоже

Шлюз is2sql построен на использовании специальных тегов в файлах шаблонов, наподобие ColdFusion. Но это, конечно не ColdFusion, и вместо сотни тегов CFML is2sql не использует и десятка тегов. Cookie не используются, так как все параметры передаются методами http GET или POST. Все просто.

Где взять

Если Вы используете Delphi, скачайте себе исходники с <http://is2sql.al.ru>. Там же вы можете скачать только выполнимые файлы, скомпилированные для использования BDE, или для клиента Interbase. Исходные тексты также можно скачать с <http://www.torry.ru>, Components/Internet. Скачайте себе примеры готовых приложений с <http://is2sql.al.ru>, использующих эту dll- думаю, так будет удобнее освоиться на примерах. Свяжитесь со мной <mailto:aivanov@hotmail.ru>

Как это выглядит

Это выглядит примерно так- надо нарисовать страницы во FrontPage и затем вставить в него пару - другую тегов. Для того, чтобы обеспечить разборку этих тегов, имя файла html шаблона передается dll, которая подставляет реальные значения из БД. Теги можно вставлять во FrontPage без риска искажения.

Как это работает

Каждый раз при обращении посетителя web страницы, сгенерированной шлюзом, возобновляется соединение с БД. Имя и пароль (для доступа к серверу SQL) передаются в параметрах user и key. Если страница содержит большую таблицу, задаются номера начальной и конечной строки в таблице в параметрах first, last.

Основной тег `<#t>` - загружает шаблон (который хранится в файле с расширением .htt, а sql запрос- в .sql).

Тег `<#e>` выполняет запрос и не возвращает результата, поэтому нужен только файл .sql.

Почему шаблон и SQL выражение вынесено за пределы HTML страницы? На это есть такой ответ- это следствие желания разделить дизайнерскую часть и часть, отвечающую за ее наполнение из таблиц БД. Я не раз думал, является ли это преимуществом или недостатком, и стоит ли мне сделать так, чтобы писать все в одном html файле, и всякий раз убеждался, что такое разделение удобно. И потом, если это не так, то тогда можно использовать ColdFusion ;)

Примеры

Скачайте с <http://is2sql.al.ru> примеры, чтобы посмотреть, как сделать доступ к вашим данным. Глава 2 дает представление о том, как создавать страницы с использованием шлюза is2sql. Будьте

снисходительны к моим примерам, выложенным на сайте, если скачанные оттуда примеры сразу не заработают- возможно, созданные SQL скриптами базы данных потребуют наполнения таблиц или предоставления прав.

Роль SQL в создании страниц с использованием шлюза

Как для доступа к таким базам данных, как Corel Paradox или Microsoft FoxPro, так и к SQL серверам Interbase или Oracle, используется SQL разных диалектов. Поскольку вы скачали этот файл и читаете его, чтобы узнать, как шлюз будет взаимодействовать с вашей базой данных, то вам без сомнения не составит труда применять SQL и как могло быть иначе? Для доступа к этим БД требуется установка и настройка BDE на сервере (разумеется, на клиенте BDE устанавливать не нужно). Для доступа к Interbase 4.x, 5.x, 6.x используйте версию шлюза прямого доступа к Interbase.

Последовательность чтения

Я рекомендую дочитать введение и начать чтение со второй главы «Примеры». Если у вас возникнут вопросы, обращайтесь к содержимому главы 1 «Теги» как к справочнику. Не начинайте чтение с первой главы. Глава 3 «Ресурсы» описывает возможные настройки шлюза.

Спасибо, что посетили мою страничку,

Андрей Иванов, моя страничка <http://ensen.8m.com>

Пишите ваши замечания и пожелания: aivanov@hotmail.ru

Шаблон (файл .htt)

Шаблон .htt заголовок, тело и окончание, и может также содержать часть (ELSE) для случая, когда SELECT не возвращает ни одной записи. Тело находится между тегами <#@> и <#!/@>.

Все, что находится до <#@> считается заголовком. Заголовок содержит примерно такой текст: <table>

Затем <#@> тег открывает тело. Тело- это повторяемые поля для одной строки со следующими тегами: <#f NAME="ИМЯПОЛЯ "..> Атрибуты используются для форматирования страницы/таблицы.

Окончание следует за закрывающим тегом <#!/@>, содержит примерно такой текст: </table> Если Вы хотите специальным образом информировать пользователя о том, что SELECT не вернул ни одной записи, укажите перед тегом <#!/@> тег <#!@> и поместите между ними необходимое оповещение.

Если нужно вывести поля для редактирования внутри шаблона , используйте соответствующие теги <INPUT>.

Ограничения: В теле (между тегами <#@> и <#!/@>(или<#!@>) , в заголовке и в окончании можно указывать теги <#p>.

Теги <#f> можно использовать только внутри тела. Внутри ELSE (между тегами <#!/@> тег <#!/@>) ни один из тегов <#> не работает.

Запрос SQL (файл .sql)

В запросе SQL параметры передаются со знаком ":":

:PARAMETER

Семейство не надо указывать, только имя параметра.

Знак двоеточия ":", если его надо поставить в запросе, можно ввести двумя знаками двоеточия "::".

Специальные параметры начинаются со знака "\$". Например, параметр "\$FROM" подставляет в запросе значение ip адреса клиента. Имена специальных параметров перечислены в описании тега <#A name=sys> и <#A name=ip>

Файл .par

Файл параметров позволяет задать простые подстановки в параметры. При этом параметр разбивается на слова, и каждое слово участвует в sql запросе самостоятельно там, где это указывается символами %s.

Например, параметр :WORDS описан в файле .par так:

```
WORDS=" ( W.WWORD LIKE UPPER("%s") ) "  
WORDS delimiter=" AND "  
WORDS %chars=~.*"  
WORDS min=3  
WORDS empty=" ( W.WWORD LIKE "ENSEN" ) "
```

Допустим, параметр WORDS равен "слово1 слово2~ 12". Параметр разбирается на слова и заменяется на sql выражение

```
( W.WWORD LIKE UPPER("слово1") AND ( W.WWORD LIKE UPPER("слово2%")
```

Обратите внимание, знак тильды заменен на знак %, а слово 12 отброшено как короткое (короче 3 символов)

Теперь, если параметр WORDS пустой или, например, все слова оказались короче трех символов, то в sql выражение будет добавлено выражение, заданное в строке WORDS empty:

```
(W.WWORD LIKE "ENSEN")
```

кавычками)

? Не забудьте про пробелы в начале и конце этой строки, если это необходимо, (например, "_AND_", где знаком "_" изображены пробелы для отделения в выражении sql.

Длина слова больше нуля. Опустите эту строку, если не надо делать контроль за минимальной длиной слова в параметре. Если ПАР min меньше или равен 0, минимальная длина слова в параметре не проверяется

Строка в файле .par

ПАР=

ПАР delimiter=

Часть 1.

Пути и теги

Глава 1. Пути

Путь /show или /s

Это основной путь is2sql.dll. Путь /show действует по умолчанию, то есть его можно опустить. Также можно использовать укороченный путь - /s. В пути нужно указать два обязательных параметра (за исключением случаев, когда подходят значения по умолчанию): dbs и form

Назначение этого пути в том, что он производит подсоединение к заданной в параметре dbs базе данных, разбирает шаблон, имя которого передается в параметре form, и выполняет найденные в шаблоне запросы к БД. Если требуется указание имени и пароля пользователя СУБД, отличных от действующих по умолчанию, то их нужно задать в параметрах user, key.

Результат выполнения запросов возвращается в создаваемой по шаблону html странице.

Значения по умолчанию, как и все другие настройки шлюза, записываются в реестр Windows.

dbs. Имя базы данных.

Если не указаны значения по умолчанию алиаса БД, имени и пароля пользователя СУБД, тогда в пути /show обязателен параметр dbs. В зависимости от типа используемой СУБД, нужно указать два дополнительных параметра: user и key. Эти параметры обычно не нужны в случае использования настольных СУБД, и обязательны в случае SQL серверов. Так, например, при использовании Corel Paradox из приведенных параметров требуется только параметр dbs. В случае Interbase, Oracle требуются все три параметра dbs, user и key.

Если задано несколько параметров dbs, берется первое значение, остальные будут проигнорированы. Например:

```
http://localhost/scripts/is2sql.dll/s?dbs=db&user=IAM&key=multi&form=/login.htm&...
```

Обязательный параметр form.

Имя файла первой формы html. Файл шаблона может содержать ни одного, один или несколько специальных тегов. Если файл располагается в каталоге, отличном от принимаемого по умолчанию, нужно указать относительный или абсолютный путь. Можно использовать алиасы web-сервера и собственные алиасы шлюза, это зависит от настроек шлюза.

Во избежание путаницы алиасы пишите с использованием слешей («/»), а пути - с использованием обратных слешей («\»). Каталог по умолчанию задается в настройках. Например,

```
...dbs=paradox1&form=queryform.htm&...
```

будет загружать файл queryform.htm из каталога по умолчанию,

```
...dbs=paradox1&form=/queryform.htm&...
```

будет загружать файл queryform.htm из корневого каталога web- сервера. Если web- сервер не поддерживает алиасы или шлюз настроен так, что не использует алиасы web- сервера, то будут использованы алиас «/» самого шлюза. Следующие два примера

```
...dbs=paradox1&form=c:\autoexec.bat&...
```

```
...dbs=paradox1&form=..\..\autoexec.bat&...
```

не допускаются, так как шлюз не позволяет указывать относительный или абсолютный путь за пределами каталога по умолчанию. Правильным путем будет указание такого пути, который находится внутри каталога по умолчанию.

Использование алиасов позволяет обойти это ограничение. В общем случае рекомендуется использовать только алиасы для придания гибкости вашим web- приложениям.

Если в параметре form задана html форма, содержащая тег контроля доступа <#r>, обязательны два параметра uid и ukey, идентифицирующие пользователя (в данном случае имеется в виду не пользователь базы данных, а пользователь web- приложения)

Замечание: uid и ukey - условные названия параметров, на самом деле вы можете использовать любые имена параметров.

user, key. Имя и пароль пользователя СУБД.

Берется первое значение (если было задано несколько). Если задан этот параметр, обычно вам нужно указать и параметр key. Например,

...dbs=dermajet&user=mary&key=Kay

[family.]parameter1

Параметры будут переданы в запрос, ассоциированный с семейством family (задаваемый тегами <#f> или <#e>) если указанный параметр существует в запросе. Можно опустить имя семейства. Например,

...&family1.P1="Herb a life"&...

передает в SQL запрос, записанный в файле family.sql, значение параметра P1: Herb a life.

first. Номер первой записи

Синтаксис: [family.]first

По умолчанию 1. Действует на все теги <#table>, если не указано семейство.

Пример url: first=10

Пример указания в HTML- форме: <input type=hidden name=first value=10>

last. Номер последней записи

Синтаксис: [family.]last

По умолчанию 0- все записи. Действует на все теги <#table>, если не указано семейство.

Пример url: last=15

Пример указания в HTML- форме: <input type=hidden name=last value=15>

step. Шаг линейки листания.

Синтаксис: [family.]step

Действует на теги <#a name=sys value=pageline>, в семействе family. Если не указано семейство, то действует на все теги <#a name=sys value=pageline>. По умолчанию, как задано в настройках (если не задано, то 10).

Пример url: step=20

Пример указания в HTML- форме: <input type=hidden name=step value=20>

cr. Перекодировка страницы.

По умолчанию нет перекодировки.

Синтаксис: cr=coderpage-coderpage

Пример url: cr=winkoi (html страницу преобразовать из Win CP1251 в КОИ-8)

Пример: cr=doswin (html страницу преобразовать в Win CP1251 из DOS CP866)

Перекодировочные таблицы хранятся в файле usr.erc или другом файле, заданном в настройках. С помощью вспомогательной утилиты exlat вы можете создавать собственные перекодировочные таблицы.

push. Канал

Синтаксис: push=HH:MM:SS

HH:MM:SS означают часы, минуты и секунды между посылками.

Параметр push принуждает is2sql посылать страницу через заданные интервалы времени. При этом браузер должен поддерживать режим проталкивания. Документ отправляется с полем ContentType: multipart/x-mixed-replace;boundary=EnSeN_UsEfUl_UtIlItleS.

Когда наступит время обновления документа, будет отправлена строка, задаваемая boundary- то есть EnSeN_UsEfUl_UtIlItleS. Браузер по получении этой строки очистит содержимое и отобразит новую порцию информации. В режиме проталкивания соединение не теряется, и части документа вводятся в браузер, пока браузер сам не рассоединится.

Замечание: В текущей версии задание этого параметра приведет к ошибке, так как хотя заголовок формируется правильно, посылается только 1 раз.

hash, pwd, forminvalid. Вторичный пароль пользователя

Три параметра hash, pwd, forminvalid осуществляют проверку вторичного пароля пользователя. В параметре pwd передается пароль для проверки его правильности для пользователя, идентифицируемом параметром hash. Если pwd правильный, то выполнение запроса продолжается. Если pwd неправильный, возвращается форма, указанная в параметре forminvalid. При этом, если форма не определена или неправильно задана, выдается предупреждение. Запрос независимо от этого не выполняется.

В параметре pwd может передать уникальный идентификатор администратора, хранимом в в реестре. В таком случае is2sql.dll вычислит правильное значение pwd для пользователя с идентификатором, заданном в параметре hash. В этом случае в теге <#a name=sys value=_pwd_> будет возвращен правильный пароль. В теге <#a name=sys value=_hash_> возвращается идентификатор пользователя, по которому вычислен пароль.

Замечание: В настройках задаете пароль в HashUniqueID.

forminvalid. Форма сообщений об ошибках.

forminvalid используется не только для проверки вторичного пароля пользователя. forminvalid задает форму сообщения об ошибке соединения с базой данных (то есть если пользователь ввел неверный пароль пользователя базы данных, указал неверный алиас или свое имя). Если этот параметр не задан, то использует форма, действующая по умолчанию.

Такая форма задается параметром реестра DBConnectFailForm. Если в реестре не указана форма по умолчанию, или она не может загрузиться, то тогда выдается текстовое сообщение об ошибке. В форме сообщения ошибки соединения можно использовать тег <#metas>. Все другие теги не разбираются и оставляются в теле формы без изменений.

В параграфе «Зарезервированные параметры для пути /show» этой главы перечислены все параметры, значения и действия которых жестко предопределено.

Путь /rec или /r

Синтаксис:

rec?dbs&user&key&cp&qry[&qry.]param=..[&fld][&rec][&decode=z][&cp=][&content-type=]

Путь /rec выдает значение одного поля одной записи. Этот путь предназначен для получения значения поля из других программ, например, ActiveX модулей из браузера. Рекомендуется использовать не в браузере, а только в других приложениях, например, для доступа к BLOB полям. Например, BLOB поле может быть записано в сжатом виде. Для сжатия поля может быть использована, например, библиотека zLib, входящая в поставку Delphi.

Параметры dbs, user, key, cp, qry.param такие же, как в пути /show. Параметр qry задает имя файла запроса (расширение .sql нужно опустить).

Необязательный параметр fld - имя поля таблицы. Если параметр опущен, берется первое поле в возвращаемом наборе данных.

Необязательный параметр rec - номер записи (начиная с 0). Если параметр опущен, берется первая запись в возвращаемом наборе данных.

Необязательный параметр decode указывает, делать или нет декомпрессию значения поля на сервере. По умолчанию поле не декомпрессируется и передается вызывающему ActiveX для декомпрессии и отображения в браузере. decode=z - декомпрессировать zLib'ом. Например,

```

```

Необязательный параметр content-type задает тип возвращаемого контента. По умолчанию- text/html.

Путь /info

Синтаксис: /info?key=xxx&show=[copyright,list,queries,threads]

Queries- показывает число запросов со времени очередного старта is2sql.dll

threads- показывает число доступных, активных и неактивных потоков

list- показывает опции is2sql.dll, значения переменных по умолчанию

copyright- показывает информацию о is2sql.dll

Обязательный параметр key должен совпадать с паролем администратора. Необходимость указания пароля вызвана тем, что /info может выдать содержимое установленных по умолчанию переменных.

Путь /setup

Путь /setup для изменения некоторых настроек шлюза is2sql. Путь /setup будет выполняться на платформе Windows95/98 и в определенном случае- на Windows NT/2000, а именно, когда пользователь, от имени которого запущен процесс web- сервера, имеет право модифицировать реестр Windows. Обычно это действие запрещено.

Синтаксис: /setup?key=xxx[&newkey=xxx][&MkNull=bbb]

key. Пароль администратора.

Синтаксис: key=<administrator password>.

Обязательный параметр, пароль администратора. Пароль администратора хранится в реестре.

newkey. Новый пароль администратора

Синтаксис: newkey=<password>

Задает новый действующий пароль администратора

Пример: newkey=асТалависта

banner, PatternPath, setup, info. Шаблоны страниц.

Следующие параметры зарезервированы для последующего применения, не устанавливайте их:
banner=
PatternPath=
Setup=
info=

mknnull. Замена пустых значений на строковое значение NULL

Синтаксис MknNull=<BOOL>

булево значение:

T, true, 1, CHECKED - включить замену

F, false, 0, <ПУСТО>- выключить замену (по умолчанию).

Включает (и выключает) режим замены пустых значений на строку "NULL". Рекомендуется не включать замену во избежание проблем с разбором SQL- выражений.

MaxThreads. Максимальное число потоков

Синтаксис: MaxThreads=<maximum number of threads>

Задаёт максимальное число потоков. По умолчанию- 32. Значение по умолчанию задается соответствующим параметром в реестре.

CacheConnections. Кеширование потоков.

Синтаксис: CacheConnections=<BOOL>

Возможные булевы значения <BOOL> писаны выше в описании параметра mknnull.

Включает и выключает кеширование потоков. Не задавайте этот параметр, равный 0 (или No, False) вместе с другими параметрами (кроме key=<Admin password>), так как другие параметры не возымеют действия.

Более того, если кеширование отключено, то вы не сможете изменить любые значения ключом / setup. Параметры будут изменены только при включенном кешировании. Поэтому всегда указывайте ..&CacheConnections=1 при изменении параметров. Затем, если Вы хотите отключить кеширование, сделайте это отдельно: CacheConnections=0

Перечисленные последние три параметра задаются из реестра.

Путь /reload

key=<administrator password>. Обязательный параметр, пароль администратора. Пароль хранится в реестре.

Прерывает выполнение всех потоков и считывает заново параметры из реестра. Будьте осторожны при использовании этого ключа. Используйте /reload после внесения изменений настроек в реестре. Иначе остановите сервис http и запустите заново, чтобы is2sql.dll выгрузилась и загрузилась заново (при новом обращении).

Следующий параграф, как резюме главы, перечисляет наиболее важные параметры основного пути.

Зарезервированные параметры для пути /show

Путь шлюза is2sql /show содержит один обязательный параметр form, необязательные параметры соединения с сервером БД, необязательный параметр перекодирования страницы html и пользовательские параметры -то есть те параметры, которые были заполнены на предыдущей форме и передаются следующей форме или просто html странице (задаваемой обязательным параметром form).

form

Это единственный обязательный параметр, задающий выходной html. В имени файла можно использовать как физический путь, так и алиасы web- сервера.

dbs, user, key

Эти параметры задают подключение к базе данных. Если эти параметры не заданы, то их значения берутся из реестра .

cp

Параметр принуждает перекодировать содержимое создаваемой is2sql страницы из одной кодировки (в которой создан шаблон) в кодировку для представления пользователю.

Любые бинарные данные, если они оказываются на странице, подвергаются перекодированию.

Замечание: коды расширенных символов, записанные в виде ÿ (это символ буквы 'я' в кодировке Windows CP 1251), не будут перекодированы.

Смотрите также параграф «Задание файла для перекодирования в кодировку страницы в реестре» части «Настройки»

`first, last, step`

`first, last, step`- необязательные параметры, задающие начальную и конечную запись во всех таблицах, показываемых на странице.

`hash, pwd`

`hash, pwd` и `forminvalid` используются для вычисления вторичного пароля клиента

`forminvalid`

`forminvalid` используется для указания формы, на которую произойдет передача в случае сбоя SQL сервера.

Глава 2. Теги

Все теги имеют полное имя и короткое имя (состоящей из первой буквы) и выглядят так:

`<#имя_тега {параметр1[=значение]}>`

Вы можете в настройках изменить символ решетки «#», используемый в начале тега, но пожалуйста, не делайте этого. Обратите внимание, не допускается использовать знаки пробела перед знаком «#».

Теги разделяются по месту их применения на две категории- теги, используемые в html, и теги, используемые в шаблонах .htt.

Глава 3. Теги в html

В этой главе рассматриваются теги, используемые в html коде, по которому шлюз генерирует динамические html- страницы. Их всего несколько:

`<#parameter>`, `<#table>`, `<#execute>`, `<#a>`, `<#x>`, `<#copy>`, `<#hidden>`, `<#right>` и `<#metas>`

Последний тег используется для целей тестирования.

`<#parameter>` или `<#p>`

Тег `<#p NAME="parameterName" [fmt=currency|money]>` вставляет значение переданного параметра с заданным именем. Параметр, переданный в форму либо методом POST, либо GET- шлюз не делает между ними различия. Если задан параметр `fmt="currency"`, параметр будет написан прописными буквами. Если параметр не представляет десятичное число, будет выдано

ноль рублей 00 копеек.

`fmt=money` представляет денежный формат в соответствии с настройками системы, то есть

100,00 р.

Но, если в реестре задан иной формат валюты, будет использован заданный в реестре.

Если в реестре задан параметр `CurrencySimple=0x1 (True)`, то вместо формата из настроек системы будет использован простой формат 100=00

Пример: `<#p name="family1.P1">`

Тег `<#table>` или `<#t>`

Тег `<#t>` используется для выражений SQL, которые возвращают наборы данных: SELECT. Тег `<#t>` имеет один обязательный параметр `name`:

`<#t name="familyName">`

где `familyName`- имя семейства файлов: запроса .sql, шаблона .htt и, возможно, файла параметров .par. Можно указывать абсолютный путь файла, относительный или с использованием алиасов web- сервера и шлюза. Не забудьте удалить расширение файла- параметр `name` задает использование не одного, а двух или трех файлов с одинаковым именем, но с различным окончанием (расширением имени файла): .sql, .htt, .par.

Пример с использованием алиасов:

`<#t name="/query1">`

Пример с указанием абсолютного пути:

`<#t name="c:\scripts\query1">`

Пример с использованием относительного пути:

`<#t name=query1>`

В последнем примере файлы ищутся в каталоге по умолчанию. Будьте осторожны с указанием абсолютного и относительного пути- если файл располагается за пределами каталога по умолчанию, файлы не будут загружены.

Можно задать параметры, передаваемые в форму, используя знак \$ для обозначения параметра. Например,

`<#t name="$Param_Name1$Param_Name2">`

при значении параметров, переданных из предыдущей формы `Param_Name1=form` и `Param_Name1=12` будет равнозначен

`<#t name=form12>`

Конец имени параметра определяется по символу, отличному от '_', '0'..'9', 'A'-'z' либо по началу нового имени параметра (то есть если встречен символ \$). Специальный случай- два знака '\$' подряд. Он служит признаком конца параметра. Например,

`<#t name=$Param_Name1$Param_Name2>`

будет заменен на

`<#t name=formParamName2>`

В следующей главе «Теги шаблона .htt» рассматриваются теги, которые используются в шаблонах .htt для тега <#table>

Тег <#execute> или <#e>

Тег <#e> выполняет DML- выражения SQL, которые не возвращают наборов данных: INSERT, UPDATE, DELETE. Тег <#t> используется для выражений SQL, которые возвращают наборы данных: SELECT. В этом заключается разница между этими тегами.

Тег <#e NAME="sqlFile" [stored=boolean]> имеет обязательный параметр name, указывающий семейство также, как в теге <#t>. Однако, так как набор данных не возвращается, то в семействе нужен только один файл .sql.

Пример: <#e name="dosmth.sql">

Замечание: В случае interbase- версии шлюза нет ошибки в смешивании <#t> и <#e>, однако рекомендуется для лучшей совместимости использовать соответствующий тег в зависимости от того, является ли sql выражение SELECT или INSERT, UPDATE или DELETE.

Необязательный параметр stored может иметь значение:

t[true], f[false], 1, 0, Y[es], N[o], D[a], N[et]

что указывает, является ли указанное семейство просто sql выражением или вызывает хранимую процедуру. По умолчанию- просто sql- выражение.

Можно задать параметры, передаваемые в форму, используя знак \$ для обозначения параметра.

Пояснения смотрите в описании тега <#t>

Необязательный параметр stored=true указывает, что в файле .sql задано выражение EXECUTE PROCEDURE... в соответствии с синтаксисом Interbase (справьтесь в документации Interbase) или (в случае BDE и NC_OCI) как описано в файле SPPrsr.pas:

[EXECUTE|PROCEDURE]NAME (PARAMETER [IN|OUT] [DEFAULT VALUE], ...)

Тег <#metas>

Диагностический тег <#metas> может быть полезен при отладке web- приложений. Тег вставляет теги META:

<META NAME="Generator" CONTENT=>

<META NAME="Request" CONTENT=>

..

для отладки (в основном для контроля передаваемых параметров, в особенности в случае использования метода CGI POST вместо GET)

Тег <#a>

Тег <#a> предоставляет различную информацию о ходе исполнения шлюза.

Тег <#a name=sys|ip value=параметр> имеет обязательный параметр name, который может принимать два значения: sys и ip. Рассмотрим их по отдельности.

<#a name=sys>

Тег <#a name=sys value=Категория> имеет обязательный параметр value, который может принимать следующие значения:

STATUS | BANNER | SCRIPT | PATTERNPATH | QTY | PAGELINE | PAGELINELIMIT | PAGELINESTEP | ECCNAME | OPTMKNUL | OPTIONHTML | ECCXLAT | FIRSTNO | LASTNO | TODAY | TIME | NOW | MAXTHREADS | ACTIVETHREADS | INACTIVETHREADS | CACHECONNECTIONS | QUERIES | PWD

Смысл большинства из этих значений очевиден. Ниже приводится описание некоторых параметров тега <#a name=sys>

<#A name=sys value=OPTMKNUL>. Тег возвращает значение CHECKED или пустую строку и показывает, подставляются ли значения NULL в пустых параметрах, передаваемых в sql предложения. По умолчанию - нет, это приведет к возникновению ошибок в sql предложениях, когда пользователь какой- либо параметр сделает пустым. SQL требует применения значения NULL. Чтобы активировать режим, отредактируйте соответствующий параметр реестра или каждый раз устанавливайте режим

../setup/...&mknul=true

<#A name=sys value=NOW>. Тег возвращает комбинацию даты и времени (в форматах, задаваемых ShortDateFormat и LongTimeFormat).

<#A name=sys value=MAXTHREADS | ACTIVETHREADS | INACTIVETHREADS | CACHECONNECTIONS | QUERIES>. Тег возвращает максимальное число потоков, активных

потоков и неактивных (закрытых) потоков, уничтожаются ли потоки после завершения потока (CHECKED=нет, а кэшируются, "- уничтожаются). QUERIES показывает число запросов, выполненных is2sql.dll с момента последнего запуска is2sql.dll (что все равно, перезапуска web-сервера).

<#a name=sys value=pwd hash=Строка>. Тег вычисляет пароль для заданной строки. Нельзя в качестве параметра hash передать другой тег, например, <#f>. Вместо этого используйте опцию fmt=hash тега <#f>, рассматриваемого в следующем параграфе.

<#a name=sys value=_pwd_> помещает значение пароля пользователя (для получения пароля пользователя нужно задать пароль администратора в параметре is2sql.dll pwd)

<#a name=sys value=_hash_> помещает значение номер телефона пользователя, по которому вычислен пароль для текущего года. Если не был задан пароль администратора, то пароль не вычисляется, а значение будет то, которое ввел пользователь.

<#a name=sys value=mail> отправляет письмо адресату. Нужно задать следующие параметры:

Параметр

host	Сервер smtp (имя или ip адрес)
port	Порт сервера. По умолчанию 25.
userid	Учетная запись пользователя на сервере smtp.
from	email отправителя
fromname	Имя отправителя
address	email получателя или поле таблицы БД, где записан e-mail
subj	Тема сообщения. Можно указать имя поля таблицы, которое содержит тему сообщения.
banner	имя файла, содержимое которого помещается перед телом сообщения (банер в письме).
body	Имя параметра в котором находится тело письма, или Имя файла (можно указывать алиасы web сервера) или содержание письма
mime=[p]lain [h]tml [s]gml [t]ab	Тип кодировки письма. По умолчанию text/plain (plain). Достаточно первой буквы.
ok	Подтверждение об отправке, выводимое в форму в браузере. По умолчанию ничего не подставляется на месте тега.
fail	Сообщение об ошибке при отправке, выводимое в форму в браузере. По умолчанию выводится описание ошибки. Этот параметр фактически подавляет сообщение об ошибке. Рекомендуется использовать параметр fail="", так как
charset	Кодовая страница, в какой отправлять адресату. Например, koi8-r, us-ascii, win-1251. По умолчанию win-1251. Если тип mime- text/html, то внутри html также можно указать кодировку. Один знак пробела понуждает использовать значение, наиболее подходящее для письма. Не делайте предположений относительно этого значения. Считайте его в таком случае неопределенным. А вообще- то это us-ascii.

В конверте письма (поля subj, from) можно использовать кириллицу в кодировке WIN1251. Если обнаруживаются символы из расширенного набора символов ASCII, то поле кодируется.

Тег <#a name=sys> может использоваться с параметром newvalue в описанных ниже случаях.

<#a name=sys value=..[newvalue=...]>

Следующие значения value в теге <#a name=sys value=..[newvalue=...]> могут быть вставлены или выставлены новые значения:

STATUS- строка статуса операции

BANNER- заголовок ответа

SCRIPT- имя файла скрипта (DLL библиотеки)- изменение этого параметра не приводит к использованию другой DLL

PATTERNPATH- путь к файлам по умолчанию

ECCNAME- имя кодировочного файла

CONTENT-TYPE- тип ответа (по умолчанию -'text/html')

Последний случай представляет наибольший практический интерес. Например, используйте следующие теги в ваших шаблонах:

<#a name=sys value=Content-Type newvalue="text/plain">

- тогда браузеру будет передано содержимое текстового типа, без HTML разметки.

<#a name=sys value=Content-Type newvalue="text/plain">

На месте тега не будет ничего вставлено. Следующий пример вставляет текущее значение типа ответа:

<#a name=sys value=Content-Type>

Управление индикатором типа контента иногда очень необходимо и влияет на то, как обрабатывает его браузер клиента. Ваш web- сервер также может быть настроен на различные действия в зависимости от типа контента и это может привести как к положительным, так и отрицательным эффектам.

<#a name=IP>

Тег **<#a name=IP value=Категория>** вставляет информацию о текущем соединении

Параметр value может принимать значения:

USERAGENT|FROM|HOST|REFERER|DERIVEDFROM | CONNECTION | REMOTEADDR | REMOTEHOST | SERVERPORT | SCRIPTNAME | EXPIRES | SERVERVARLIST

Смысл большинства из этих значений очевиден. Ниже приводится описание некоторых параметров тега **<#a name=ip>**

Тег **<#a name=ip value=host>** может использоваться для управления предоставляемой клиенту информации внутри приложения. Ниже приведен пример, какая информация об ip соединении может быть доступна:

```
USERAGENT      Mozilla/4.0 (compatible; MSIE 5.0; Windows NT; DigExt)
REFERER        http://ensen/stat/log/statnav.htm
connection     Keep-Alive
host           10.8.1.15
server         10.8.1.15
port           80
script         /cgi-bin/is2sql.dll
expires        29.12.1899
```

В поле from возвращается email клиента, если браузер позволяет это делать.

Тег **<#a name=ip value="SERVERVARLIST">** возвращает список 29 полей ECB и серверных переменных (каждая на новой строке, имя переменной=значение переменной):

Описание Значение (пример)

Серверные переменные	Значение (пример)
SERVER_PROTOCOL	HTTP/1.1
URL	/scripts/is2sql.dll
HTTP_ACCEPT	image/gif, image/jpeg, application/x-comet, */*
HTTP_HOST	ensen
HTTP_REFERER	http://ensen/quiz/
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 5.0; Windows NT; DigExt)
CONTENT_TYPE	application/x-www-form-urlencoded
REMOTE_ADDR	62.76.127.26
REMOTE_HOST	62.76.127.26
SCRIPT_NAME	/scripts/is2sql.dll
SERVER_PORT	80
HTTP_CONNECTION	Keep-Alive

Тег <#c>.

Тег **<#c>** служит для создания копий параметров, инкремента значений параметров и создания новых параметров в зависимости от значения параметра тега action: «=», «++» и «==».

Копирование параметров

Тег **<#c>** служит для создания копий параметров. В параметрах указывается имя исходного параметра и имя нового, создаваемого параметра:

<#c name=старый_Параметр action="=" value=Новый_Параметр>

Тег создает новый параметр **<#p>**, копируя значение в него из старого параметра. Операция присвоения (action="=") нужна в случае, когда параметр формы нужно передать дальше, но с другим именем, как, например, этого требует указание квалификатора семейства:

<#c name=family1.parl action="=" value=family2.parl>

Инкремент параметров

<#c name=Параметр action="++" [type= i[nTEGER] | f[LOAT] | d[ATE] | t[IME]]>

инкрементирует значение параметра. В type указывается тип параметра:

type	Тип	Величина инкремента
i	целое число	1
f	вещественное число	1.0
d	дата	1 день
t	дата и время	1 день

По умолчанию i - целое число (десятичное или шестнадцатеричное , шестнадцатеричное должно предваряться символом '\$')

Создание параметров

<#c name=Значение action=="==" value=НовыйПараметр>

Создает новый параметр если параметр отсутствовал и записывает значение, заданное в параметре name. Для чего должен использоваться - неясно.

Передача параметров из формы в форму

Часто требуется передавать полученные в cgi запросе параметры из формы в форму. Для этого в HTML используется тег `<input type=hidden>`. Чтобы упростить передачу параметров из формы в форму, используйте:

```
<#h name=Параметр[,Параметр]>
```

который разворачивается шлюзом в серию тегов `<input type=hidden name=Параметр value=Значение_Параметра>`

Когда форма будет отправлена, параметр из скрытого параметра будет передан дальше. Через запятую (без пробелов) можно указать список передаваемых параметров. Наиболее часто нужно передавать параметры, отвечающие за соединение с базой данных db, user и key, например:

```
<#h name=dbs,user,key>
```

- передаются параметры соединения с базой данных.

```
<#h name=first,last,step>
```

- передаются параметры управления количеством одновременно отображаемых записей таблицы

Тег <#r>

Тег `<#r name="FAMILY" gid="идентификатор группы" [form="FAILURE_FORM"] [encode=0|1]>` контролирует доступ к html- документу на основе переданных параметров и содержимого таблиц базы данных.

Иногда нужно разделить посетителей своего web- сайта на категории, например, на продавцов и покупателей. Вы можете использовать разные возможности, в том числе предоставляемые web-сервером для аутентификации и разграничения пользователей. Вы можете использовать также специальный тег `<#r>` для управления того, кому показывать или не показывать данную форму.

По умолчанию encode=1 (пароль хранится в базе в зашифрованном виде)

Предположим, что Вы ведете простую таблицу продавцов guid, состоящую из трех столбцов: guid, ggid, gukey. В первом столбце- идентификатор продавца, во втором- признак принадлежности к группе продавцов, в третьем- пароль продавца.

Вам нужно будет сделать такой CheckEmp.sql файл:

```
select count(ggid) from company.guid g
where (g.guid = ':uid') and (g.ggid=':gid') and (g.gukey=':ukey')
```

и поместить его в папку вместе с шаблонами.

Затем в вызывающей html форме введите теги `<input type=text name=uid>` и `<input type=password name=ukey>`. Вы можете передать и по другому, важно что в вызываемую защищаемую форму были переданы параметры uid и upwd. Имена этих параметров могут быть и другими. В параметре uid передается идентификатор продавца, в параметре ukey- его пароль.

В вызываемой защищаемой форме вставьте тег `<#r name=CheckEmp gid="идентификатор группы продавцов" form=/failformaccess.htm encode=0>`

Параметр form тега `<#r>` можно опустить. Этот параметр задает форму, которая будет выдана в случае, если у пользователя нет в списке продавцов или он ввел неверный пароль. По умолчанию имя такой формы берется из реестра (параметр AccessDeniedForm), а если он не задан, то выдается текстовое сообщение.

В форме сообщения о невозможности доступ к странице можно использовать только тег `<#metas>`. Все другие теги не разбираются и остаются в теле формы без изменений. На месте этого тега вставляется сообщение от сервера базы данных с описанием ошибки соединения.

Тег контроля прав доступа к форме `<#r>` используется в любом месте html формы. Если задано несколько тегов, действует первый из них.

Глава 4. Теги в шаблонах .htt

Когда шлюз, разбирая переданный в обязательном параметре form, файл HTML, находит в нем тег `<#table> (#t)`, он загружает соответствующий шаблон .htt и выполняет заданное в файле .sql выражение SELECT. Если оператор SELECT возвращает непустой набор данных, шлюз форматирует полученный набор данных в соответствии с частью тела шаблона .htt и вставляет записи в выходной документ HTML. Количество записей может быть ограничено параметрами first и last, если нет, то будут вставлены все найденные записи.

Для указания полей возвращаемого набора данных (таблицы базы данных) в шаблоне .htt

используется тег `<#f>` с дополнительными опциями форматирования.

Тег `<#field>`. Вставка полей из таблиц БД

Тег `<#field>` (`<#f>`) выглядит следующим образом:

```
<#f name="ИМЯПОЛЯ" [fmt=currency|money|hash|datetime specifier="mmmm ууу"]  
[replace=emptysame] [http=encode][coru=ИмяПараметра]>
```

Этот тег используется в .htt между тегами `<#@>` и `<#/@>`, то есть в теле шаблона. Подставляет значение поля возвращаемого набора данных, преобразованное для передачи http.

`replace=emptysame`

Опция `replace`, равная `emptysame`. Назначение этой опции- в том, чтобы не помещать на странице повторяющиеся значения. Значение поля, отличающееся от предыдущего, будет помещено на страницу. Значение поля, не отличающееся от предыдущего, не будет помещено на страницу. Эта опция удобна для того, чтобы разгрузить html- страницу от повторяющихся полей.

`http=encode`

Опция `http`, равная `encode`, вызывает `HTTPEncode` для обеспечения прозрачности http. Опцию `http=encode` необходимо указывать только в ссылках типа:

```
<a href="http://localhost/cgi-bin/is2sql.dll/show?parameter=<#f name=field_name  
http=encode>
```

чтобы возможные символы были правильно преобразованы (пробелы- в знаки '+', управляющие и символы из расширенного набора - к виду %XX)

`fmt=currency`

Если не указан параметр `fmt=currency`, значение поля будет приводиться к строковому представлению, используя свойство `AsString`. Это касается также полей дат, времени и денежных полей. Если задан параметр `fmt=currency`, целая часть числа будет преобразована в рубли, а дробная- в копейки и результат будет написан прописью (в рублях и копейках). При этом часть числа, меньшая 1 копейки, будет округлена функцией `round()`, то есть большее или равное в точности полкопейки будет округлять до следующего числа целого числа копеек, а меньшая- часть копейки будет отброшена.

`fmt=money`

`fmt=money` представляет денежный формат в соответствии с настройками системы, то есть 100,00 p. Но, если в реестре задан параметр `CurrencySimple=0x1 (True)`, то вместо формата из настроек системы будет использован простой формат 100=00

`fmt=datetime`

Опция `fmt=datetime specifier=xxx` форматирует строку даты и(или) времени. Параметр `specifier` задает формат отображения даты и времени. Например, формат "mmmm уууу" будет отображать полное название месяца и четыре цифры года, дата и время не будут показаны.

`fmt=hash`

Опция `fmt=hash` вычисляет пароль по заданному полю, только когда в параметре передан правильный пароль администратора.

Опция `coru` предназначена для того, чтобы создать заново или изменить старое значение параметра формы. Может использоваться для передачи в следующую форму параметра значения из поля таблицы.

Специальные поля

Если поле не найдено, тег `<#f>` пробует найти специальные поля, какие перечислены в описании тега `<#a name=sys|ip>`. Если нет специальных полей, в html код будет вставлена строка `FIELD_<Имя_несуществующего_поля>_DOESNT_EXISTS`

Внешние форматирующие функции

```
<#f name="ИМЯ_BLOB_ПОЛЯ" fmt=external dll="" func="" [specifier=""]>
```

Специальный модификатор `fmt=external` имеет дополнительные параметры, которые отсутствуют в случаях других модификаторов формата `fmt`:

`dll`- имя динамически загружаемой библиотеки. В пути имени можно указывать алиасы web-сервера.

`func`- имя функции

`specifier`- необязательный строковый параметр, передаваемый в функцию динамически загружаемой библиотеки.

Этот тег используется для вызова функции из динамически загружаемой библиотеки для форматирования поля таблицы.

Тег <#X>

Тег <#X formParameter formatString> обеспечивает безопасное редактирование в редакторе Microsoft FrontPage и другими html- редакторами, которые не допускают вложения тегов.

Редактор html может не позволить вставить в шаблоне внутри тега псевдотег, например, <INPUT value=<#P name=chbank.pbankno>>. Символы '<' и '>' будут заменены на < и >. Тогда is2sql.dll не сможет найти теги и подставить значения. Чтобы избежать это, используется тег <#X> внутри которого можно поместить теги, используя знаки > и <. formParameter соответствует <#P name=family.par>. Если первый параметр в апострофах, и начинается с символа 'a', вместо параметра формы вставляется параметр sys или ip.

Нужно указать два дополнительных параметра тега - replace2fld= и tag=. На всякий случай, эти теги должны быть указаны первыми. Первый параметр задает имя поля таблицы или специальные поля ip и sys. Второй параметр задает имя создаваемого тега. Например,

```
<#X replace2fld="chbank.pbankno" tag="input" type="text" name="chbank.pbankno"
size="20" value="***">
```

В передаваемом браузеру тексте будет следующее:

```
<input type="text" name="chbank.pbankno" size="20" value='12345'>, где 12345-
значение параметра формы chbank.pbankno.
```

Также можно использовать следующий синтаксис:

```
<#x replace2fld="TODAY" tag=input name="delnotes.pmdate" value="***">
```

В передаваемом браузеру тексте будет следующее:

```
<input name="delnotes.pmdate" value="31.12.2000">
```

Другие теги

В зависимости от реализации, в теле шаблона .htt допускается использовать теги:

<#a>- как описано в предыдущей главе

<#p>- как описано в предыдущей главе

Тег копирования <#с> не используется, вместо него используйте в случае необходимости модификатор сору=Имя_Нового_Параметра тега <#f>

Тег <#x> не используется.

Замечание: Сейчас можно указать только один параметр тега с одинаковым именем. Если задано несколько параметров тега с одинаковым именем, используется первый, остальные игнорируются.

Глава 5. SQL запросы

Вместе с шаблонами .htt вы создаете файлы запросов .sql. Имена полей, в возвращаемом наборе данных, указываются в тегах <#f>, как показано в предыдущей главе. В sql выражения передаются все полученные HTML- формой параметры, а также копируемые и создаваемые тегом <#с> параметры.

Для указания параметра поставьте перед именем параметра знак двоеточия: «:», например:

```
SELECT * FROM GADGET
WHERE :GADGET_ID+0 > 123
```

или

```
SELECT * FROM :WHEREFROM
WHERE :GADGET_ID+0 > 123
```

Обратите внимание, что вы можете подставлять любые выражения, как в примере выше, или как здесь:

```
SELECT * FROM GADGET
WHERE :GADGET_NAME_TYPE:GADGET_NAME_NUMBER = 'ВСТАВКА1234'
```

Параметры :\$. в sql запросах

В sql выражениях можно подставлять значения, которые возвращают теги <#a name=sys> и <#a name=ip>. Для этого такой параметр должен начинаться символа доллара "\$". Например,

```
SELECT ' :$UserAgent' UA ...
```

вернет строку описания браузера в виде списка совместимости.

Глава 6. Параметры форм

Параметры форм создаются элементами управления. Если несколько элементов имеют одинаковое

имя и передают значения (как, например, может быть с элементами управления CHECKBOX), они все будут доступны через тег <#P> и, возможно, другие теги. При этом значения будут разделены между собой запятой и пробелом. Таким образом, полученный список может использоваться в SQL конструкции SELECT ... WHERE FIELD IN:

```
..IN(:LISTPAR1)
..IN (1, 2, ..N) ..
```

Как быть, если Вы хотите передать из нескольких элементов управления значения, но не разделять их запятой и пробелом? Для этого нужно поставить символ "+" в конце значения:

```
<INPUT TYPE=HIDDEN NAME=VALUE1 VALUE="F:\src\+">
<INPUT TYPE=HIDDEN NAME=VALUE1 VALUE="mytext.txt">
```

В результате тег <#P name=VaLuE1> вернет:

```
F:\src\mytext.txt
```

В примере 5 следующего раздела подробно рассматривается один из вариантов использования сшивающего знака «+»

Если нужно поставить знак плюс в конце строки, поставьте перед плюсом знак "\":

```
a+b+c\+ -> a+b+c+
```

При этом, если в форме будет присутствовать другой элемент с тем же именем, конкатенация параметров будет производиться как список, разделенный запятой с пробелом.

Если параметры в списке будут представлять собой не слова, а целые предложения, каждый элемент НЕ будет заключаться в одиночные кавычки:

```
'1 параметр', 2, '3, 4', 'Конец' - не так!
```

Проверка на то, является ли параметр строкой и, соответственно, нуждается в кавычках, не делается. Это повышает требования к аккуратности написания sql выражений и созданию параметров- Вы должны делать это сами. Зато отключение возможности такой проверки позволяет Вам написать примерно такой код:

```
..value="1" ... value="2, 3, 4"
```

тогда результат слияния будет:

```
1, 2, 3, 4
```

Если включить (только при перекомпиляции) возможность такого закавычивания, то получилось бы:

```
1, '2, 3, 4'
```

Если параметр затем передается в sql выражение, то в sql выражении проверяется, есть ли одиночная или двойная кавычка непосредственно перед употреблением параметра. Если есть, то кавычки не ставятся, а если они есть- то они удаляются. Если нет, то тогда предложение закавычивается, а одиночное слово или число - нет. Поэтому для строковых параметров ставьте кавычки в sql предложениях.

Часть 2.

Примеры

Глава 1- Простые примеры (DBDEMOS)

Глава 2- Простые примеры (IBLOCAL)

Глава 1. Очень простые примеры (DBDEMOS)

В этой главе приведено несколько очень простых примеров, демонстрирующих основные идеи построения web- приложения. В примерах используются готовые таблицы Paradox, входящие в поставку Delphi (алиас DBDEMOS).

Для проверки примеров в действии запустите web- сервер на своем компьютере.

1. Отображение html файла

Введите в строке Адрес браузера:

```
http://localhost/scripts/is2sql.dll/s?dbs=DBDemos&form=/index.htm
```

Вы увидите начальную страницу вашего web- сервера, и если это не так- то есть появились всякие сообщения об ошибках, тогда вернитесь назад к чтению предыдущей части и повторите настройки. Смысл указания после имени DLL «/s» рассмотрим позже, в замечании к примеру 6.

То, что это начальная страница, следует из параметра form=/index.htm. Знак слеша «/» - это алиас web- сервера, указывающий на его корневой каталог. Эта страница не содержит никаких данных из таблиц базы данных DBDemos, зачем же тогда он указан, и можно ли опустить параметр dbs?

Посмотрим, что получится, если это сделать:

```
Unknown database. Alias: db_is2sql Invalid user name (), password or database alias ()
```

Это происходит потому, что шлюзу обязательно нужно указать параметры базы данных, даже если страница /index.htm не содержит никаких данных из нее, и на то есть свои причины. Все же существует путь, как сделать так, чтобы не указывать параметры базы данных, такие, как dbs, но давайте рассмотрим его позднее и перейдем к следующему простому примеру.

2. Отображение таблицы

Теперь сделаем маленький файл ex2.htm и поместим его в корневой каталог web- сервера:

```
<HTML><BODY><#t name=/ex2></BODY></HTML>
```

В строке Адрес браузера введем:

```
http://localhost/scripts/old/is2sql.dll/s?dbs=DBDemos&form=/ex2.htm
```

и получим иконку с сообщением об ошибке:

```
sql error : select sql "/ex2" not found
```

Почему это произошло, ответ простой- в теге <#t name=/ex2> указано семейство ex2, и его нужно создать. Создадим файл sql запроса ex2.sql:

```
SELECT * FROM ANIMALS
```

и файл шаблона htt ex2.htm

```
<p>Name: <#f name=NAME> Size: <#f name=SIZE> Weight: <#f name=WEIGHT> Area: <#f name=AREA>
</p>
```

Оба файла также поместим в корневой каталог. Обновив страницу в браузере, получим:

```
ame: Angel Fish Size: 2 Weight: 2 Area: Computer Aquariums
ame: Boa Size: 10 Weight: 8 Area: South America
ame: Critters Size: 30 Weight: 20 Area: Screen Savers
...
```

Теперь попробуем такой адрес в браузере:

```
http://localhost/ex2.sql
```

и, конечно, получим текст sql- запроса.

Как вы уже поняли, и html файл, и sql файл и файл шаблона оказались доступны для чтения потому, что все они размещены в корневом каталоге web- сервера, что действительно неразумно и позднее мы эту умышленную неаккуратность будем исправлять в главе «Безопасность»

3. Более красивая таблица

В предыдущем примере поля не разделяются тегами таблицы <table>, строк <tr>, колонок <td>. Если мы используем их, то тем самым улучшим внешний вид готовой html страницы. Для этого исправим файл шаблона /ex2.htm примерно так:

```
<tr>
<td><#f name=NAME></td><td><#f name=SIZE></td>
<td><#f name=WEIGHT></td><td><#f name=AREA></td>
</tr>
```

Придется исправить и /ex2.htm, вставив шапку таблицы:

```
<HTML>
<BODY>
<table><tr><td>Name</td>
<td>Size</td>
<td>Weight</td>
<td>Area</td></tr>
<#t name=/ex2>
</table>
</BODY>
</HTML>
```

Создадим файл /ex1.htm:

```
<HTML><head><meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
<title>Пример</title></head>
<body><form method="GET" action="/scripts/is2sql.dll/s">
<input type=hidden name=dbs value=DBGDemos><input type=hidden name=form value="/
ex2.htm">
<input name=ANIMAL_NAME value="Boa"><input type=submit value="Запрос">
</form></body></html>
```

Откройте в браузере:

<http://localhost/ex1.htm>

Нажмите кнопку «Запрос» и получите результат, как в предыдущих примерах. Попробуйте ввести в окне ввода вместо Воа другие слова, например, Parrot.

Теперь таблица действительно выглядит как таблица:

Name	Size	Weight	Area
Angel Fish	2	2	Computer Aquariums
Boa	10	8	South America
Critters	30	20	Screen Savers
House Cat	10	5	New Orleans
Ocelot	40	35	Africa and Asia
Parrot	5	5	South America
Tetras	2	2	Fish Bowls

4. Таблица, не содержащая данных

Может оказаться так, что sql запрос не возвратил ни одной строки. В примере 3 в готовой html странице от таблицы останется только шапка. Можно сделать так, чтобы вместо пустой таблицы, не содержащей ни одной строки, кроме шапки, было выведено предупреждающее сообщение. Для этого нужно использовать в шаблоне специальный разделитель <#!@>. Подробнее о разделителях будет сказано в примере 1 следующей главы, пока же исправим шаблон примера 3:

```
<tr>
<td><#f name=NAME></td><td><#f name=SIZE></td>
<td><#f name=WEIGHT></td><td><#f name=AREA></td>
</tr>
<#!@>
<td colspan=4>Нет ни одного животного</td>
```

В строке Адрес браузера введем:

<http://localhost/scripts/old/is2sql.dll/s?db=DBGDemos&form=/ex2.htm>

и получим тот же результат, что в предыдущем примере. Нужно исправить файл /ex2.sql так, чтобы ни одна запись таблицы animals не удовлетворяла условию поиска:

```
SELECT *
FROM ANIMALS
WHERE NAME=':ANIMAL_NAME'
```

попробовав теперь, получим результат:

Name	Size	Weight	Area
Нет ни одного животного			

Обратите внимание на взятый в кавычки параметр :ANIMAL_NAME символ «:», его обсудим в

следующем примере.

5. Параметры в sql выражениях

В предыдущем примере уже был использован параметр sql выражения, который по аналогии с другими программами, в том числе Delphi, отмечает параметр знаком двоеточия «:» перед именем параметра. Чтобы посмотреть, как происходит передача параметра из html формы в sql выражение, в строке Адрес браузера введем:

```
http://localhost/scripts/is2sql.dll/s?dbs=DBDemos&form=/ex2.htm&ANIMAL_NAME=Boa
```

В результате получим одну строку из набора данных:

Name	Size	Weight	Area
Boa	10	8	South America

Что произойдет, если опустить параметр ANIMAL_NAME, было видно в примере 4. Как вы догадались, не переданный параметр был представлен пустой строкой, а таблица animal не содержит записей с значением поля name, равным пустой строке.

Как вы уже поняли, введенные в строке браузера после знака вопроса параметры (dbs, form и другие, разделяемые знаком «&» есть не что иное, как строка GET запроса CGI. В следующем примере сделаем завершающий штрих- создадим html- форму, из которой и передадим все эти параметры вместо того, чтобы писать каждый раз в строке браузера.

Пример 6. HTML- форма

Создадим такую форму ex1-1.htm:

```
<html><head><title>Вызов формы</title></head><body>
<form method="GET" action="/scripts/is2sql.dll/s">
  <input type="hidden" name="dbs" value="DBDemos">
  <input type="hidden" name="form" value="/ex1.htm">
  <input name="ANIMAL_NAME" value="">
  <input type="submit" value="Запрос">
</form>
</body></html>
```

Откройте в браузере эту форму, введите искомое имя и нажмите кнопку. Обратите внимание на строку браузера после выполнения запроса.

Замечание к примеру 6

Обратите внимание, что строка браузера выглядит почти, но не совсем так, как это мы делали это вручную в примере 5:

```
http://localhost/scripts/is2sql.dll/s?dbs=DBDemos&form=%2Fex2.htm&ANIMAL_NAME=Boa
```

Разница в том, что символы «/» заменились на %2F, также заменяются некоторые другие символы в соответствии с требованиями CGI, просто не обращайте на это внимания. Что касается буквы /s перед знаком «?», то это сокращение от /show, одного из «путей» ISAPI DLL. ISAPI/NSAPI также позволяет опустить /s, так как /show, или /s- путь по умолчанию.

Пример 7. Улучшение результата

Как можно улучшить полученный результат? Можно просто добавить ссылку на страницу формы /ex1.htm из страницы результата. Но можно сделать еще лучше, изменив /ex2.htm так, чтобы он тоже был формой, похожей на ex1.htm:

```
<HTML><head><meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
<title>Улучшенный ответ</title></head><body>
<form method="POST" action="/scripts/is2sql.dll/s">
<input type="hidden" name="dbs" value="DBDemos"><input type="hidden" name="form" value="/
ex2.htm">
<input name="ANIMAL_NAME" value="<#p name="ANIMAL_NAME">"><input type="submit
value="Запрос"></form>
<table><tr>
<td>Name</td><td>Size</td><td>Weight</td><td>Area</td>
</tr><#t name=/ex2></table></BODY></HTML>
```

Попробуйте теперь. Обратите внимание на две вещи:

- при каждом нажатии кнопки «Запрос» вызывается одна и та же форма /ex2.htm;
- метод GET заменен на метод POST, и в строке Адрес браузера не видны передаваемые параметры.
- тег <input name=ANIMAL_NAME..> имеет атрибут value, равный одноименному параметру <#p name=ANIMAL_NAME>. Тег <#p> вставляет значение переданного в форму параметра.

Пример 8. Совместимость с FrontPage

В примере 7 есть особенность, на которую следует обратить внимание- а именно внутри тега `<input>` вставлен тег `<#p>`. Это было бы все равно, если только не загружать в `/ex2.htm` в «умный» HTML- редактор. Вложения тегов в HTML недопустимы, поэтому тег `<input>` можно исправить так:

```
<#x replace2fld=ANIMAL_NAME tag=input name=ANIMAL_NAME value="%s">
```

Атрибут тега `replace2fld` указывает переданный в форму параметр, а `tag`- тег, который будет создан. Если вы используете FrontPage или иной html- редактор, следящий за правильностью тегов, используйте тег `<#x>`, если нет, то можете отсавить пример 7 как есть.

Резюме

Тег `<#t>` вставляет записи из таблицы БД. Буква «t»- сокращение от table, и вместо `<#t>` можно писать `<#table>`.

Файл `.sql` содержит запросы, параметры, отмечаемые символом «:», передаются в него из формы HTML. Некоторые параметры HTML зарезервированы для использования шлюзом `is2sql`. В этой главе были рассмотрены два таких параметра: `dbs` и `form`

Глава 2. Простые примеры (IBLOCAL)

В этой главе будут рассмотрены примеры, использующие таблицы SQL сервера Interbase, входящие в поставку Delphi (алиас IBLOCAL).

Для проверки примеров в действии нужно запустить сервер БД Interbase.

Пример 1. Разделители заголовка и окончания `<#@>` и `<#/@>`

Две таблицы: Employee и Job содержат информацию о том, какую работу и как она оплачивается. Создадим файл соответствующего sql- запроса /ex3.sql:

```
SELECT * FROM EMPLOYEE E, JOB J
WHERE E.JOB_CODE = J.JOB_CODE AND E.JOB_GRADE = J.JOB_GRADE
ORDER BY E.FIRST_NAME, E.LAST_NAME
```

В обеих таблицах есть связующие эти две таблицы два поля: JOB_CODE и JOB_GRADE.

Создадим файл шаблона html /ex3.htm:

```
<table><tr><td>FIRST NAME</td><td>LAST NAME</td><td>JOB TITLE</td><td>SALARY</td><td>MIN SALARY</td><td>MAX SALARY</td></tr>
<#@><tr>
<td><#f name=FIRST_NAME></td><td><#f name=LAST_NAME></td>
<td><#f name=JOB TITLE></td><td><#f name=SALARY></td>
<td><#f name=MIN SALARY></td><td><#f name=MAX SALARY></td>
</tr>
<#!@><td colspan=6>Нет ни одного сотрудника</td>
<#/@>
</table>
```

Обратите внимание на разделители `<#@>` и `<#/@>`. Разделитель `<#@>` отделяет заголовок от тела шаблона. Все, что перед ним, считается заголовком. Аналогично, после разделителя `<#/@>` следует окончание шаблона.

В предыдущих примерах эти разделители не рассматривались и появились только в этом примере. Почему? Потому что их использование необязательно. Расположенное между тегами `<#@>` и `<#/@>` тело шаблона повторяется для каждой записи.

Необязательный разделитель `<#!@>` был рассмотрен в примере 4 предыдущей главы и отмечает начало шаблона, которое будет использовано, если sql- запрос не вернул ни одной записи.

Создадим два html- файла, один для запроса /ex3.htm:

```
<HTML><head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Пример</title></head>
<body><form method="POST" action="/scripts/old/is2sql.dll/s">
<input type=hidden name=dbs value=IBLocal>
<input name=user value="SYSDBA"><input type=password name=key value="masterkey">
<input type=hidden name=form value="/ex3-1.htm"><input type=submit
value="Обновить">
</form></BODY></HTML>
```

От предыдущих примеров форма отличается присутствием тега параметра имени пользователя базы данных сервера `<input name=user...>` и его пароля `<input name=key ...>`

Второй для ответа /ex3-1.htm, уже без тега `<table>`, `</table>`:

```
<HTML>
<head><meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Пример</title></head>
<body>
<#t name=/ex3>
</BODY></HTML>
```

При желании вы можете исправить /ex3-1.htm в духе примера 8 предыдущей главы. Откройте в браузере страницу:

<http://localhost/ex3.htm>

И нажмите кнопку на форме. В результате получится:

```
FIRST NAME LAST NAME JOB TITLE SALARY MIN SALARY MAX SALARY
Ann Bennet Administrative Assistant 22935 20000 40000
...
```

В запросе не учтено географическое расположение, влияющее на оплату труда. Таблица достаточно большая и, чтобы посмотреть, сколько в ней записей, добавим в файл /ex3-1.htm строку:

```
Records: <#a name=sys value=qty>
```

сразу за `<#t name=/ex3>`, и обновим страницу. В конце страницы увидим:

```
...
Yuki Ichida Engineer 6000000 50000 90000
```

Records: 105

Строку «Records: <#a name=sys value=qty>» можно было бы вставить и в шаблон /ex3.htm за разделителем <#/@>.

Пример 2. Разделение больших таблиц на страницы

Улучшим предыдущий пример, разделив большую страницу на несколько страниц. На одной странице разместим по 10 записей, для этого в форме /ex3.htm добавим строку:

```
<input name=last value="10">
```

Если хотите, можете добавить:

```
<input name=first value="1">
```

Параметры first и last указывают, какие записи показывать на странице. Обратите внимание, что первая запись имеет номер 1, а не ноль.

Теперь, когда число записей ограничено, нужно добавить ссылки на другие страницы. Для этого вставим следующую строку после разделителя <#/@> в шаблоне /ex3.htm:

```
<td colspan=6><#a name=sys value=pageline></td>
```

Аналогичную вставку можно было бы сделать и в /ex3-1.htm.

Обновим страницу, в ее конце увидим:

```
Yuki      Ichida      Engineer      6000000      50000 90000  
[1-10 | 11-20 | 21-30 | 31-40 | 41-50 | 51-60 | 61-70 | 71-80 | 81-90 | 91-100 |]  
Records: 105
```

Диапазон записей 1-10 не является ссылкой, потому что мы уже находимся на странице, в которой присутствуют указанные записи. Ткните мышью в ссылки, чтобы проверить правильность.

Обратите на строку Адрес браузера. Как вы видите, листание страниц происходит путем указания параметров first и last:

```
http://localhost/scripts/is2sql.dll/  
s?dbs=IBLocal&user=SYSDBA&key=masterkey&form=%2Fex3-1.htm&ex3.first=11&ex3.last=20
```

Мы видим переданные параметры в строке Адрес браузера потому, что тег <#a name=sys value=pageline> формирует эти ссылки. Перед параметрами first и last задан квалификатор семейства ex3. Квалификатор указывает, что он действует только на указанное семейство /ex3.sql и /ex3.htm, при этом знак «/» и вообще весь путь опускается.

Квалификаторы семейств и их назначение

В предыдущем примере были упомянуты квалификаторы семейств. Для чего они используются, ведь мы видели, что в предыдущих примерах параметры ими не снабжались.

Квалификаторы ограничивают воздействие на выбранное семейство. Семейство состоит из файлов .sql, .htm и .par (последний файл будет рассмотрен позднее) и называется по имени, заданном в атрибуте name тега <#t>, без указания пути.

Например, если в html- форме есть несколько тегов <#t name=family...>, и если они содержат некоторые одинаковые параметры, то мы можем разделить параметры между ними, указав квалификатор.

Напротив, достаточно передать один параметр без указания квалификатора, чтобы он мог быть использован в нескольких sql запросах.

Пример 3. Добавление записи в таблицу

Для выполнения sql- запросов, которые не возвращают записи, используется тег <#e>. Также, как и тег <#t>, атрибут name указывается семейство, но так как записи не возвращаются, то отпадает необходимость в шаблоне .htm, достаточно одного файла .sql с запросами DML: INSERT, UPDATE, DELETE и другие. Буква «e»- сокращение от execute, и можно писать <#execute>. Сперва создадим /ex4.sql:

```
INSERT INTO EMPLOYEE(EMP_NO, FIRST_NAME, LAST_NAME, DEPT_NO, JOB_CODE, JOB_GRADE,  
SALARY, JOB_COUNTRY)  
VALUES (:NO, ':I', ':F', 110, 'Eng', 5, 32000, 'USA')
```

Создадим файл /ex4.htm:

```
<HTML><head><meta http-equiv="Content-Type" content="text/html; charset=windows-  
1251">  
<title>Пример</title></head>  
<body><form method="GET" action="/scripts/is2sql.dll/s">  
<input type=hidden name=dbs value=IBLocal><input name=user value="SYSDBA">  
<input type=password name=key value="masterkey">  
<p>№:<input name=no>  
<p>Имя:<input name=i>  
<p>Фамилия:<input name=f>
```

```



```

Использованы три параметра: No, F и O- номер нового сотрудника, его имя и фамилия. Так как добавление в таблицу не допускается без указания отделения, оплаты труда и некоторых других, то вписаны в /ex4.sql случайные значения.

Наконец, создадим результирующий html /ex4-1.htm:

```

<HTML><head>
<title>Запись создана</title></head>
<body><#execute name=/ex3>Новый сотрудник: <#p name=i> <#p name=f></BODY>
</HTML>

```

Попробуйте загрузить форму /ex4.htm. Установите значение No 150 и введите имя и фамилию нового сотрудника, нажмите кнопку «Записать». Запись добавлена.

Что будет, если ввести неверное значение No? Во всяком случае запись добавлена не будет. Попробуйте заведомо неверные значения (например, No = 1000 приведет к ошибке, так как таблица EMPLOYEE не допускает ввода такого числа в поле EMP_NO). В этом случае будет выведена иконка с сообщением об ошибке. Вы можете изменить html- документ, который будет использован для сообщения об ошибках, но это будет обсуждено позже.

Замечания о хранимых процедурах

Если is2sql скомпилирован для работы с Interbase, то нет разницы между <#t> и <#e> в том, как они выполняются. В случае других- sql серверов и настольных БД в теге <#t> допускаются только выражения SELECT. Хранимые процедуры (stored procedure) в случае Intebase можно вызывать из обоих тегов (в случае <#t>- оператором SELECT, <#e>- EXECUTE).

В случае Borland Database Engine следует использовать тег <#e> с атрибутом stored=true и описать ее в .sql, как описано в предыдущей части.

В случае использования драйвера BDE для доступа к РСУБД Oracle обратите внимание на особенности вызова процедур, как описано в в предыдущей части.

Пример 4. Передача параметров из формы в форму

Некоторые параметры должны передаваться из формы в форму, такие, как dbs, user, key. В вашем web- приложении также могут быть такие параметры.

Для передачи параметров можно было бы писать:

```



```

Эти записи укорачиваются следующим образом:

```

<#h name="dbs,user,key,i"><#hidden name="f">

```

Тег <#h> (сокращение от слова hidden) следует размещать внутри каждого парного тега <form></form>, где требуется передача перечисленных параметров. Следующий пример демонстрирует возможную ошибку:

```

<form...>...<#hidden name="f,i">...<#hidden name="i">...</form>

```

Вероятная причина ошибки- повторение параметра i (в предыдущем примере- имени), результирующий html- файл будет иметь два поля формы i:

```

...<input type="hidden" name="i" value="Yuki">
...<input type="hidden" name="i" value="Yuki">

```

Пример 5. Случай нескольких одинаковых параметров

Что происходит, если несколько параметров формы имеют одинаковое имя? Шлюз в этом случае сшивает все такие параметры в одно. Приведенные в примере 4 параметры i будут соединены в один параметр:

```

Yuki, Yuki

```

Это имеет как свои плюсы, так и минусы. Пример 4 демонстрирует отрицательное свойство сшивания значений одинаковых параметров как следствие неаккуратности. Теперь рассмотрим пример, демонстрирующий положительную сторону сшивания.

Создадим шаблон для выбора сотрудников из списка (drop- down menu) .htt:

```

<select name="empno" size=1 multiple><#@>
<option value=<#f name=emp_no>><#f name=last_name>, <#f name=first_name></option>
<#!@><option value="-1">Нет сотрудников</option><#/@></select>

```

Указание `multiple` позволяет выбрать одно, несколько или ни одного значения из меню. Что будет, если ни одного значения не будет выбрано? Иногда в таком случае хорошо добавить значение, например, `-1`.

В `html` файла укажем это сторожевое значение:

```
<input type=hidden name=empno value="-1">
```

В таком случае будет гарантировано, что значение параметра `empno` не будет пустым. Это позволяет писать такие `.sql`:

```
... WHERE EMP_NO IN (:EMPNO)...
```

без боязни того, что пустое значение `:EMPNO` приведет к ошибке.

Пример 6. Форма начала теста (экзамена)

Рассмотрим достаточно сложный пример, форму, с которой начинается тестирование в приложении «Тест», которое рассматривалось выше. Вот его `html` файл `/quiz/ex/exam.htm` (слева номер строк для ссылок):

```
1 <html><body>
2 <#r name="/tquiz/1/stright" form="/quiz/st/failidentify.htm" encode="0">
3 <#e name="/tquiz/next/mkquestions" stored=true> <#e name="/tquiz/next/
add_st2ex" stored=true>
4 <#r name="/tquiz/next/allowexam" form="/quiz/ex/examnoallow.htm" encode="0">
5 <#e name="/tquiz/next/rm_st2part"> <#e name="/tquiz/next/add_st2part">
6 <form name="f1" method="post" action="/scripts/is2sql.dll">
7 <input type="hidden" name="form" value="/quiz/ex/np.htm+">
8 <input type="hidden" name="qorder" value="0"><#h
name="dbs,user,key,sno,uid,ukey,exno">
9 <#t name="/tquiz/next/partnext">
10 <h2><#t name="/tquiz/1/exname"></h2>
11 <p>Результаты предыдущих попыток: <#t name="/tquiz/1/prevexamgrade"></p>
12 <p>Время: <#t name="/tquiz/1/examtime"> </p>
13 <input type="submit" value="Приступить к тесту" name="B1"></form></body></html>
```

Строка 2 проверяет переданные параметры `ukey` (пароль студента) и `uid` (email студента), следующим образом (вот файл `/tquiz/1/stright`):

```
SELECT COUNT(SNO) FROM STUDENT S
WHERE (S.SPWD=':UKEY')
AND (S.SEMAIL=':UID')
```

Строка 3 вызывает две хранимые процедуры, генерирующие таблицы вопросов и добавляющую запись о том, что студент приступил к сдаче теста. Если переданные в форму идентификатор и пароль студента неверные, то эти процедуры выполняться не будут, так как разбор `html` будет сразу прерван.

Строка 4 проверяет возможность сдачи теста, исходя из свойств теста (его стоимости), вот содержимое файла `/tquiz/next/allowexam`:

```
SELECT COUNT(E.EXNO) FROM EXAM E, ST2EX SE
WHERE SE.SNO=:SNO AND SE.SEXNO=:EXNO AND SE.SEXNO=E.EXNO
AND (E.EXPRICE IS NULL OR E.EXPRICE<=0.0 OR SE.SPAID>0.0)
AND (E.EXREQUESTPRICE IS NULL OR E.EXREQUESTPRICE<=0.0 OR SE.SPAIDREQUEST>0.0)
```

Строка 5 требует пояснений. Каждый тест может состоять из нескольких независимых частей. Эта строка указывает, что студент приступает к сдаче первой части теста. Строки с 6 по 13 задают форму и выводят свойства теста. Строки 6, 7 и 8 задают форму и передаваемые дальше параметры. Строка 7 указывает обязательный параметр `form` с суффиксом «+».

Так как параметр `form` более нигде не указывается, то в следующую форму будет передано значение `/quiz/ex/np.htm`, без сшивания с другим значением.

Строка 9 предоставляет это второе значение параметра `form`, когда все части экзамена будут завершены. Файл `sql /tquiz/next/partnext.sql` выглядит так:

```
SELECT PNO, PNAME, PDESC, PQCOUNT FROM PART P WHERE P.PNO =
(SELECT MIN(SPNO) FROM ST2PART SP WHERE SP.SSNO = :SNO+0
AND (SP.SDONE IS NULL OR SP.SDONE<'1'))
```

В этом `sql` выражении в таблице `ST2PART` в поле `SDONE` указывается, завершена сдача части теста, или нет. Так как требуется только одна запись части экзамена, а не все, то используется функция `MIN`. Сведения о части берутся из таблицы `PART`. Эти две таблицы связываются полями `PNO` и `SPNO` - уникальным номером части экзамена.

Шаблон `/tquiz/next/partnext.htm` использует значения полей `PNO` и `PQCOUNT` из этого `sql` запроса:

```
<#&>
<input type=hidden name=pno value=<#f name=pno>>
<input type=hidden name=qcnt value=<#f name=pqcount>>
<#!&>
<input type=hidden name=form value=".stop.htm">
<#/&>
```

В случае, когда `sql` запрос возвращает пустой набор данных, шаблон устанавливает второе значение параметра `form=".stop.htm"`.

Итак, когда все части экзамена завершены, в выходной html форме задаются два значения form:

```
/quiz/ex/np.htm+  
.stop.htm
```

Сшивающий знак «+» дает в результате одно значение обязательного параметра

```
/quiz/ex/np.htm.stop.htm
```

Этот файл завершает тест, в нем выводятся результаты теста. Файл /quiz/ex/np.htm.stop.htm не содержит сшивающих параметров:

```
1 <html><body>  
2 <#r name="/tquiz/1/stright" form="/quiz/st/failidentify.htm" encode="0">  
3 <#e name="/tquiz/next/calcscoreex" stored=true>  
4 <form name="f1" method="post" action="/scripts/is2sql.dll">  
5 <input type="hidden" name="qno_prev"><#h name="dbs,user,key,sno,uid,ukey,exno">  
6 <input type="hidden" name="form" value="/quiz/ex/exls/exls.htm">  
7 <#t name="/tquiz/1/exname"><p>Тест полностью завершен. Ваши результаты: <#t  
name="/tquiz/next/exscore"><p>Части: <#t name="/tquiz/next/partsscore">  
8 <input type="submit" value="Список тестов" name="B1"></form></body></html>
```

После того, как студент посмотрит результат, страница вернется к списку доступных тестов /quiz/ex/exls.htm (строка 6).

Перейдем теперь к вопросу о том, каким образом перебирается в экзамене часть за частью? Очевидно, это должно делаться в ссылаемых формах. Первая из них - /quiz/ex/np.htm. Рассмотрим содержимое его файла:

```
1 <html><head><body>  
2 <#r name="/tquiz/1/stright" form="/quiz/st/failidentify.htm" encode="0">  
3 <form name="f1" method="post" action="/scripts/is2sql.dll">  
4 <input type="hidden" name="form" value="/quiz/ex/nq.htm+">  
5 <#h name="dbs,user,key,sno,uid,ukey,exno,pno,qcnt">  
6 <#t name="/tquiz/next/questionnext"><#t name="/tquiz/next/partname"><#e name="/  
tquiz/next/parttimeon"><#t name="/tquiz/next/parttime">  
7 <input type="submit" value="Готово" name="B1"></form></body></html>
```

Строка 2 - проверка прав, точно такая же как в /quiz/ex/exam.htm. В строке 4 указывается обязательный параметр form, тоже со знаком плюс «+». Ссылаемая форма /quiz/ex/nq.htm выводит вопросы части теста. В строке 5 передаются наследуемые параметры, exno и pno- номера сдаваемого теста и текущей части теста, qcnt- число вопросов в части теста для информации посетителю web- страницы. Строка 6 выводит сведения о текущей части теста и помечает, когда студент приступил к сдаче части теста. По аналогии с предыдущим примером, как вы уже догадались, в <#t name="/tquiz/next/questionnext"> будет вставлен дополнительное значение параметра form /quiz/ex/nq.htm.stop.htm. SQL выражение выглядит так:

```
SELECT MQNO, MORDER FROM MKQUESTION WHERE MSNO = :SNO+0  
AND MPNO = :PNO+0 AND MORDER = :QORDER + 1
```

Шаблон также имеет часть <#!@> с указанием второго значения параметра form:

```
<#e>  
<input type=hidden name=qno value=<#f name=mqno>>  
<input type=hidden name=qorder value=<#f name=morder>>  
<#!@>  
<input type=hidden name=form value=".stop.htm">  
<#/@>
```

Когда все вопросы в части завершаются, значение form после сшивания будет /quiz/ex/nq.htm.stop.htm:

```
1 <html><body>  
2 <#r name="/tquiz/1/stright" form="/quiz/st/failidentify.htm" encode="0">  
3 <#e name="/tquiz/next/doanswerlast"><#e name="/tquiz/next/calcscore"  
stored=true>  
4 <form name="f1" method="post" action="/scripts/is2sql.dll">  
5 <#h name="dbs,user,key,sno,uid,ukey,exno">  
6 <input type="hidden" name="form" value="/quiz/ex/np.htm+">  
7 <#t name="/tquiz/next/partnext"><#t name="/tquiz/next/partname"><p>Часть  
завершена. Всего было вопросов: <#p name="qcnt">.  
8 </form></body></html>
```

В строке 7 /tquiz/next/partnext точно закрывает «цикл» по частям теста, выражение /tquiz/next/partnext.sql уже видели выше. Вспомните, что происходит, если все части были завершены.

Замечание: Почему в форме части теста использовался тег <#t name="/tquiz/next/questionnext">, выбирающий следующий вопрос? Отчасти для того, чтобы при отсутствии вопросов в части теста сразу перейти к следующей части. Тоже самое можно сказать и о тесте- при отсутствии частей он сразу завершается.

В завершение посмотрим на форму /quiz/ex/nq.htm, которая выводит вопрос и принимает значение ответа на предыдущий вопрос:

```

1 <html><head><body>
2 <#r name="/tquiz/1/stright" form="/quiz/st/failidentify.htm" encode="0">
3 <#e name="/tquiz/next/doanswer">
4 <form name="f1" method="post" action="/scripts/is2sql.dll">
5 <input type="hidden" name="form" value="/quiz/ex/nq.htm">
6 <#h name="db, user, key, sno, uid, ukey, exno, pno, qcmt, pqtme">
7 <#t name="/tquiz/next/questionnext">
8 <#t name="/tquiz/next/questiondesc"><p>Ответ: <#t name="/tquiz/next/answer">
9 <input type="submit" value="Далее" name="B1"></form></body></html>

```

Строка 8 идентична началу строки 6 в /quiz/ex/np.htm и выбирает следующий вопрос, а если они закончились, добавляет второе значение form, после сшивания form будет равен /quiz/ex/nq.htm.stop.htm, рассмотренный выше.

В строке 3 запоминается ответ на предыдущий вопрос, sql выражение:

```

UPDATE MKQUESTION SET MANO=:ANO+0 WHERE MSNO = :SNO
AND MPNO = :PNO AND MORDER = :QORDER-1 AND MANO < 1

```

В предыдущем примере (/quiz/ex/nq.htm.stop.htm) в строке 3 указано примерно тоже самое: <#e name="/tquiz/next/doanswerlast">.

```

UPDATE MKQUESTION SET MANO=:ANO+0 WHERE MSNO = :SNO AND MPNO = :PNO AND MANO < 1
AND MORDER = (SELECT max(morder) from mkquestion where mpno=:PNO and msno=:SNO)

```

Пример 6. Изменение типа контента

Иногда нужно изменить тип контента, передаваемый браузеру. Например, если нет нужды использовать разметку HTML, можно указать браузеру, что сгенерированный шлюзом контент является простым текстом, включив следующий тег в любом месте файла HTML шаблона:

```
<#a name=sys value=Content-Type newvalue="text/plain">
```

Встретив этот тег, шлюз применит тип text/plain к генерируемой странице, установив в заголовке значение Content-Type в заданное значение. Другими применимыми типами могут быть:

```

image/gif
audio/wav
application/pdf

```

для индикации того, что шлюз возвращает картинку в формате GIF, звуковой файл и PDF документ соответственно. По умолчанию каждая страница, генерируемая шлюзом, имеет тип контента:

```
text/html
```

Путь /record

Например, вставьте ссылку

```

<head>
<bgsound src="/scripts/is2sql.dll/r?qry=/getwav&fld=wavblob&wavname=chimes.wav&
content-type=audio/wav" loop="-1">...

```

где wavname- параметр, передаваемый sql getwav. SQL выражение getwav возвращает BLOB поле wavblob.

Параметр content-type указывает браузеру тип контента audio/wav. Этот параметр нужно указывать в данном случае обязательно, так как по умолчанию тип контента будет text/html, и браузер откажется воспроизводить звук. Подробнее о пути /r смотрите в части «Пути и теги», глава «Пути».

Замечание: Ошибка BDE при работе с TTable с BLOB- полями, выражающаяся в добавлении дополнительного байта в некоторых случаях на работу шлюза не действует, так как компонент TTable не используется.

Выше был рассмотрен путь /record, теперь рассмотрим пример пути /s.

Путь /show

Для указания типа контента в пути /s нужно использовать тег <#a name=sys value=Content-Type newvalue="..."> в шаблоне .htt. Вот файл шаблона loadgif.htt:

```
<#@><#f name name=GIFBLOB><#!@>Нет ничего<#/@>
```

Файл шаблона img.html:

```
<#a name=sys value=Content-Type newvalue="image/gif"><#t name=loadgif>
```

Файл sql:

```

SELECT GIFBLOB FROM GIFIMAGES I
WHERE I.INAME=:imgidx

```

Как вы видите, он должен возвращать только одно значение.

При загрузке файла шаблона html шлюзом:

```
http://localhost/scripts/is2sql.dll/s?...form=/img.html&imgidx="sun.gif"
```

браузеру будет возвращена картинка gif из таблицы gifimages. Многоточие указывает другие параметры, требуемые для подключения к базе данных, если это требуется.

Обратите внимание, что в обоих приведенных примерах файлов не допускаются лишние пробелы и другие символы. Проследите, что ваш текстовый редактор не вставил дополнительных символов,

например, последовательность CR LF в конце файлов. Это нужно потому, что данный пример возвращает содержимое BLOB- поля таблицы базы данных.

Глава 3. Безопасность

Следующие примеры взяты из приложения «Тесты». Полные тексты скриптов для создания БД Interbase или Oracle вы можете найти в дистрибутиве или на моей web- страничке. Для полноты восприятия перед чтением последующих примеров ознакомьтесь со структурой таблицы STUDENT приложения «Тесты»:

```
CREATE DOMAIN IID AS INTEGER;
CREATE DOMAIN ISO AS VARCHAR(8) COLLATE WIN1251;
...
CREATE DOMAIN IDESC AS VARCHAR(128) COLLATE WIN1251;
CREATE DOMAIN IDT AS DATE;
```

Эти операторы DDL Interbase создают типы, применяемые в приложении. Домены ISO..IS3, IDESC также создают символьные типы с нужной сортировкой полей. Таблица STUDENT содержит поле SNO с уникальным неизменяемым идентификатором студента, используемом внутри приложения. Поле SEMAIL содержит e-mail студента. Так как e-mail студента уникален и легко запоминается, то он используется для его идентификации:

```
CREATE TABLE STUDENT (
  SNO      IID,
  ...
  SPWD     IS1,
  SEMAIL   IDESC,
  ...
  SREMOVED IDT);
```

Поле SPWD содержит пароль студента, который он вводит во время входа в web- приложение. Идея заключается в том, что другие студенты, не зная паролей, не смогут получить несанкционированный доступ к чужим данным.

Пример 1. Проверка прав доступа

На html- страницах, используемых шлюзом, в первых строках вписан тег <#r>:

```
<#r name="/tquiz/1/stright" form="/quiz/login/faillogin.htm" encode="0"> ...
<#h name="dbs,user,key,sno,uid,ukey,exno">
```

Параметр uid содержит идентификатор пользователя (то есть его e-mail), параметр ukey содержит пароль пользователя. Эти параметры, также как параметры соединения с БД: dbs, user, key и идентификатор записи студента sno (он используется в других sql выражениях), передаются дальше (тег <#h>).

Указанный stright.sql выглядит так:

```
SELECT COUNT(SNO) FROM STUDENT S
WHERE (S.SPWD=:UKEY)
AND (S.SEMAIL=:UID)
```

В теге <#r> указан параметр encode=0, иначе в таблице STUDENT поле SPWD должно быть зашифровано. Если в таблице STUDENT есть запись с e-mail и паролем, запрос вернет значение, большее 0, и тег <#r> разрешит загрузку и показ пользователю html- страницы. Если пароль указан неверно, то тег <#r> прекратит загрузку страницы и вернет пользователю другую html-страницу, а именно faillogin.htm.

Если не задан form, используется значение настроек AccessDeniedForm. В заданной форме отказа можно использовать только тег <#metas>

Имена и пароли СУБД

Обычно ваш сервер БД недоступен из внешних сетей, поэтому удаленный пользователь не сможет воспользоваться логинами БД. Однако в любом случае рекомендуется в вашем web приложении скрывать логин назначением параметров dbs, user и key по умолчанию. Для этого используйте настройки одноименных параметров.

Часть 3. Ресурсы

В этой части рассматриваются возможные настройки работы шлюза. Часть настроек можно сделать с использованием пароля администратора из браузера, указав путь шлюза /setup. Однако большинство настроек доступно только через реестр.

Оптимизация работы

Переустановка соединения

Для восстановления соединений с различными БД нужно настроить реакцию на типы ошибок, после которых нужно восстановить соединение с БД. В реестре перечислите НАЧАЛА строк сообщений об ошибках, после которых нужно восстановить соединение с БД.

Шлюз is2sql при обнаружении ошибки БД проверяет ее описание на совпадение с перечисленными в разделе ReconnectError ошибками и прерывает соединение. Есть и второй путь, значение параметра Each=1 заставляет прерывать соединение в случае любой ошибки, возвращаемой СУБД. Параметр Count задает последний номер в списке (начиная с 1).

Число потоков (threads)

По умолчанию число потоков - 8. Вы можете значительно сэкономить число сессий (соединений) с БД, указав нужное число потоков, особенно если Вы ограничены числом сессий лицензией.

Шлюз не управляет тем, в каком потоке выполнять конкретный запрос. Это значит, что если через шлюз соединение делается к одной БД от имени одного пользователя, время реакции будет почти моментальным, так как поток сохраняет соединение с заданной БД от имени заданного пользователя.

Напротив, даже если соединение ранее в потоке было установлено, но заданные параметры dbs, user, key отличаются от бывших в прежнем обращении в потоке, шлюз закроет соединение потока с сервером БД и откроет его заново.

Другими важными настройками являются алиасы web- сервера и шлюза, алиасы баз данных в случае Interbase- версии, форматы дат, времени и валют. Эти и другие настройки рассматриваются в первой главе.

Алиасы баз данных

В зависимости от того, скомпилировали вы is2sql.dll с использованием BDE, или применили компоненты Interbase Express или NC OCI 8, алиас базы данных трактуется различным образом. *Замечание:* эти компоненты требуют установленного клиента Interbase или Oracle, но не требуют установки BDE.

Алиасы BDE

Алиасы BDE указываются утилитой *BDE Administrator*. Никаких дополнительных настроек не требуется, так как информация из раздела реестра DbAlias не используется в случае BDE.

Алиасы для баз данных прямого доступа к SQL серверу

Алиасы баз данных для Interbase Express и NC OCI 8 задаются в разделе реестра DbAlias, однако отличаются по формату.

Задание алиаса для Interbase Express

Параметр выглядит так:

<Алиас>=<Файл базы данных>[,<Кодовая страница соединения>]

Файл базы данных задается в синтаксисе, который описан в руководстве Interbase. В этом параметре указывается имя файла базы данных, и, если он размещается на локальном сервере, то указывается имя сервера Interbase в сети и сетевой протокол.

Например, следующий строковый параметр

equiz=62.76.127.26:C:\SRC\equiz\db\equiz.gdb,WIN1251

задает использование базы данных Interbase C:\SRC\equiz\db\equiz.gdb, расположенной на машине с IP адресом 62.76.127.26 по алиасу equiz. Для доступа к серверу баз данных будет использован протокол TCP.

При этом указывается кодовая страница SET NAMES=WIN1251 для соединений.

Параметр <Кодовая страница соединения> не является обязательным, однако рекомендуется использовать приведенное значение, если вы используете кириллицу в полях базы данных. По умолчанию используется кодовая страница, принятая в Interbase.

Задание алиаса для NC OCI 8

Параметр выглядит так:

<Алиас>=<Строка связи>[, <необязательный параметр>]

Строка связи указывается такой, какая установлена в TNS.

Замечание: *Oracle использует схемы. Для доступа к таблице в схеме, отличной от имени пользователя БД, которым вы соединяетесь с сервером, нужно указывать эту схему.*

В этом случае удобно использовать задание схемы в переменных по умолчанию вместо явного указания схемы.

Глава 1. Реестр

Настройки записываются в реестре: HKEY_LOCAL_MACHINE\Software\IbSoftware\is2sql\1.0

MaxConnections

MaxConnections указывает максимальное число потоков (по умолчанию 8). Число 8 более чем достаточно для целей разработки и отладки. Для работы рекомендуется 32.

NoCacheConnections

NoCacheConnections=True (0x1)- потоки будут уничтожаться при закрытии сразу после выполнения потока, что может повысить надежность работы, но снизит производительность. По умолчанию неактивный поток остается доступным для следующего запроса (кэшируется). Не ставьте этот параметр для SQL серверов равным нулю, так как скорость обращения к базе данных из-за постоянных подключений и отключений пользователей резко снизится. Вместо этого нужно аккуратно настроить параметры в разделе ReconnectError.

Во время работы переключить из состояния кэширования потоков в состояние выгрузки из памяти после завершения выполнения потока можно с помощью параметра CACHECONNECTIONS=1|0 (CHECKED,) (True, False). Посмотреть состояние- с помощью тега <#a name=sys value=>CacheConnections>>

Banner

заголовок для вывода коротких диагностических сообщений.

Db

имя алиаса БД по умолчанию. По умолчанию- пустая строка.

User

пользователь БД по умолчанию. По умолчанию- - пустая строка. Используется для скрытия account'a.

Key

Пароль пользователя БД по умолчанию. По умолчанию- пустая строка. Используется для скрытия account'a

db, user и key используются в случае, если из передающей формы не были переданные одноименные параметры.

Замечание: в разделе Variables реестра также задаются переменные, которые могут использоваться внутри sql выражений. Поскольку db, user и key не относятся к разбору sql выражений, не рекомендуется перечислять db, user и key в разделе Variables.

Вы можете поставить аккаунт очень ограниченного по возможностям пользователя базы данных, я предпочитаю указывать аккаунт в параметрах db, user, key в каждой форме, обращающейся к is2sql.dll

PatternPath

Задается каталог по умолчанию. Если в реестре не задан каталог по умолчанию, то таким считается каталог, из которого запущен .DLL. (Наверное, это scripts или cgi-bin)

Раздел DbAlias. Алиасы баз данных (для не BDE)

Алиасы баз данных Oracle или Interbase, к которым обеспечен доступ через компоненты прямого доступа, должны быть перечислены в разделе реестра DbAlias.

TagPrefix

Префикс разбираемых тегов

TagPrefix, строковый параметр, по умолчанию “#”

SQLCollection

Коллекция файлов запросов и шаблонов. Если задан, то шлюз использует указанный файл (в имени файла коллекции можно указывать только абсолютный или относительный путь) для хранения файлов шаблонов, .sql и .rag- файлов.

SQLCollection задает имя файла коллекции sql запросов и шаблонов (кроме файлов, задаваемых параметром form). Формат файла текстовый. Можно указать путь, начиная с алиаса. Относительный путь берется из каталога, из которого запущен .DLL. По умолчанию имя файла sql.txt. Нельзя указывать имя каталога. Если файл sql.txt не найден, то тогда файлы читаются не из коллекции, а из каталога, заданного в параметре PatternPath. Не рекомендуется использовать этот параметр.

Раздел ResponseCustomHeader. Задаваемые заголовки ответов

Заголовки ответов перечисляются в разделе ResponseCustomHeader. Эти заголовки с указанными значениями будут добавляться в ответ, отправляемый web- сервером клиенту. В заголовке передаются также сообщения об ошибках в строке Error=, если они были обнаружены. Если ошибок не было, эта строка в заголовок ответа не помещается.

раздел Variables. Параметры по умолчанию

В разделе реестра Variables можно перечислить переменные и их значения в строковом представлении, которые могут использоваться внутри sql выражений по умолчанию.

Это значит, что наиболее часто используемые переменные можно не передавать из формы в форму, а указать их в реестре.

Примером использования такой переменной может быть имя схемы oracle.

Например, в файле myquery.sql

```
SELECT * FROM :SCHEMA.MYTABLE
```

в реестре в разделе Variables укажите вашу схему:

```
SCHEMA=MYORACLEUSERNAME
```

иначе в форму, содержащую тег <#t name=myquery> нужно было бы передавать параметр Schema, если в реестре не была указана эта переменная.

Замечание: Поскольку зарезервированные параметры dbs, user, key, sp и form не относятся к разбору sql выражений, не рекомендуется перечислять dbs, user и key и другие зарезервированные параметры в разделе Variables.

DoTransaction. Управление транзакциями

DoTransaction - строковый параметр, принимает значения "1" или "0".

Параметр DoTransaction='1' включает транзакции sql запросов, выполняемых из тегов <#t> и <#e>. По умолчанию DoTransaction='1'

Параметр DoTransaction='0' отключает транзакции sql выражений.

Однако рекомендуется устанавливать в любом случае DoTransaction='1'.

В случае прямого доступа к Interbase не имеет значения, так как транзакция подтверждается для всех sql выражений, указанных на одной странице одним COMMIT'ом независимо от значения этого параметра реестра.

В случае прямого доступа к Oracle NCOCI транзакции должны подтверждаться, и это значение должно быть равно "1".

В случае использования BDE для настольных баз данных рекомендуется указывать DoTransaction='1'.

Замечание: для Interbase и NC OCI нужно перекомпилировать is2sql со соответствующим символом условной компиляции)

DbTransisolation. Выбор типа транзакций

Тип транзакций выбирается строковым параметром DbTransisolation. Допустимые значения описаны ниже в зависимости от того, с какими символами условной компиляции USE_xxx скомпилирован is2sql

В случае BDE драйверов настольных баз данных значение этого параметра должно быть равно 0.

Замечание: Выбор типа транзакций влияет только в том случае, если разрешено выполнять sql-выражения в транзакции DoTransaction='1'.

Замечание: В случае прямого доступа Interbase значение этого параметра ни на что не влияет. В других случаях- BDE и NC OCI значение этого параметра можно оставить по умолчанию.

Значение	Мнемоника	BDE	Описание
----------	-----------	-----	----------

0	tiDirtyRead		Разрешает чтение изменений, делаемых в других транзакциях и еще не назакоммиченных .
---	-------------	--	--

1 **tiReadCommitted** Разрешает чтение только закоммиченных изменений, сделанных в других транзакциях. Это значение установлено по умолчанию.

2 **tiRepeatableRead** Запрещает внесение изменений в записи со стороны других транзакций, пока выполняется их чтение.

Обратите внимание, что в случае BDE значение по умолчанию - 1 (**tiReadCommitted**).
В случае настольных систем нужно устанавливать значение 0 (**tiDirtyRead**), иначе BDE вернет сообщение об ошибке:

The transaction isolation level must be dirty read for local databases
то есть для настольных систем нужно или устанавливать **DoTransaction=0** или **DbTransisolation=0**.
Компонент доступа к Oracle 8.x NC OCI

Значение	Мнемоника NC OCI	Описание
0	tmDefault	По умолчанию
1	tmReadWrite	
2	tmSerializable	
3	tmReadOnly	
4	tmDiscrete	

раздел **ReconnectError**. Обработка ошибок SQL.

Раздел **ReconnectError** содержит параметры:

DbErrorPrefix

DbErrorPrefix- Префикс ошибок. Если параметр **DbErrorPrefix** не указан, то он принимается равным "ORA-". Присвойте префикс ошибок для вашего SQL сервера или пустую строку. Этот параметр действует следующим образом- ищется первое вхождение указанной строки (чувствительно к регистру букв).

Коды ошибок

Начиная с этого места, далее анализируются коды или сообщения ошибок, вся строка до этой строки игнорируется. Если, однако, префикс не найден, будет проанализирована все сообщение об ошибки, начиная с первого символа.

Count - число ошибок. Каждая ошибка в реестре нумеруется, начиная с 1. 0- нет ошибок, для которых нужно делать прерывание соединения с сервером.

Each - True (0x1) -прерывать соединение с сервером в случае возникновения любой ошибки. Не рекомендуется в случае отладки приложения, так как вызывает дополнительные затраты на переустановление соединений с сервером.

1, 2,... - строковые параметры, содержащие части строк сообщений об ошибках, идентифицирующие случаи утери соединения с сервером SQL

Рекомендуется установить следующие параметры в разделе **ReconnectError**, для случаев сбоев сервера SQL Oracle 8:

Значение параметра	реестра	сообщение сервера
ORA-01041		internal error. hostdef extension doesn't exist
ORA-03313		
ORA-03314		
TNS-		

Рекомендуется установить следующие параметры в разделе **ReconnectError**, для случаев сбоев сервера SQL Interbase:

Значение параметра	реестра	сообщение сервера
Cannot perform operation - DB is not open		сервер баз данных недоступен
connection lost		связь с сервером утеряна

Для серверов, отличных от указанных, следует устанавливать значения параметров в соответствии с документацией.

*Замечание: После отладки приложение рекомендуется значение параметра **Each** сделать равным **True** (0x1)*

*Замечание: В заголовке ответа, передаваемому клиенту, передаются также сообщения об ошибках в строке **Error=**, если они были обнаружены. Если ошибок не было, эта строка в заголовок ответа не помещается. В строке может быть описание ошибки, или любой код, индицирующий обнаружение ошибок.*

DBConnectFailForm. Форма извещения об ошибке соединения с базой данных по умолчанию

DBConnectFailForm задает страницу html, которая будет выводиться в случае, если пользователь указал неправильный пароль или имя пользователя. Эта форма появится и в том случае, если пользователь ввел неверный пароль в параметрах **pwd** и **hash**. По умолчанию этот параметр

реестра - пустая строка,. В этом случае форма не загружается и выдается сообщение примерно такого содержания:

database login failed...

Этот параметр прекрывается использованием параметра **forminvalid**. То есть, если в вызывающей форме, например, был тег `<input type=hidden name=forminvalid value=/mypages/faildb.html>`, то в случае, когда невозможно присоединиться к заданной базе данных с указанным именем и паролем, вместо правильной формы будет загружен `/mypages/faildb.html`. Если этот параметр был опущен, то `is2sql.dll` попытается найти в реестре `DBConnectFailForm` и загрузить его в качестве оповещения о сбое. Если и это не удалось, будет выдано строковое предупреждение.

В имени файла формы допускается и рекомендуется использование алиасов web- сервера.

ErrorIcon. Иконка ошибки

Иконка ошибки содержит в alt описание возникшей ошибки. По умолчанию изображение иконки - `/icons/err.gif`. Размер иконки- 16x16 пиксел. Строка `ErrorIcon` в реестре задает иконку, если этот параметр в реестре существует и пуст, иконка не будет отображаться, а в страницу будет вставлен текст описания ошибки.

AccessDeniedForm. Форма извещения недостаточности прав для просмотра формы по умолчанию

`AccessDeniedForm` задает форму, которая будет выдаваться пользователю в случае, если у него недостаточно прав для просмотра заданной формы. Права контролируются тегом `<#r>`. В теге `<#r>` форма извещения о недостаточности прав задается в необязательном параметре тега `form`. Если он опущен, то тогда используется форма, заданная в реестре по умолчанию. Если `AccessDeniedForm` в реестре отсутствует, равен пустой строке или указывает на несуществующую форму или файл его нельзя по каким- то причинам загрузить, выдается простое текстовое сообщение о невозможности показать форму.

eccFileName, xlatname. Перекодирующие таблицы

`eccFileName` имя файла коллекции `xlat` таблиц. По умолчанию- `usr.epc` (файл `usr.epc` в каталоге, задаваемом `PatternPath` или в каталоге, в котором находится `is2sql.dll`).

`xlatname` имя или номер(целое десятичное) `xlat` таблицы. По умолчанию "0". Этот параметр зарезервирован для последующего использования.

Заданный таким образом файл перекодировки загружается в таблицы в память. Если файл не найден, или не может быть считан, то возможность перекодирования отключается.

Перекодирование служит для того, чтобы передать пользователю html код в заданной кодировке. Допустим, на сервере html страницы, sql запросы, файлы шаблонов и результаты выполнения sql - запросов все в кодировке Windows CP 1251. Вы можете указать тег

`<meta http-equiv=»Content-Type» content=»text/html; charset=windows-1251">`

в заголовке html страницы, или, если возникает такая потребность, перекодировать содержимое передаваемых через `is2sql` html- страниц в другую кодировку, например, `koi-8r`

Тогда из вызывающей формы нужно будет передать параметр `cr`:

`...&cr=win-koi...`

Тогда входные параметры будут переведены в кодировку `koi-8r`, а результирующий html код также будет приведен из кодировки windows CP 1251 в кодировку `koi-8r`

Значение параметра `cr` в случае использования файла `usr.epc` могут принимать строго определенные значения

pagelineXXX. Линейки страниц

Линейка страниц представляет собой примерно следующее:

| 1-10 | 11-20 |

внизу таблиц. Ссылки задаются параметрами `first` и `last`

Формат строки листания страницы `PAGELINE`

Формат строки листания страницы `PAGELINE` задается 7 параметрами в реестре.

`PageLineInactive` строка формата не текущей страницы. По умолчанию `"%d-%d"`

`PageLineActive` строка формата текущей страницы. По умолчанию `'<input type=hidden value=»%s»>%d-%d'`

`PageLineStart` открывающая скобка. По умолчанию `" ("`

`PageLineFinish` закрывающая скобка. По умолчанию `") "`

`PageLineDelimiter` разделитель между ссылками страниц. По умолчанию `" | "`.

`PageLineStep` шаг (число записей на странице). По умолчанию 10.

`PageLineLimit` число ссылок на странице. По умолчанию 10.

Замечание: *Линейка листания страниц `<#A name=sys value=pageline>` не понуждает*

подсчитывать число записей.

Поэтому, для аккуратного отображения перед этим тегом вставьте тег `<#A name=sys value=qty>`. Тег `qty` подсчитает число записей, и линейка будет правильно отображена. Иначе будет показана длинная линейка, (но не длиннее 10 ссылок, или как задано в настройках).

В случае локальной БД Вы можете захотеть пренебречь затратами на подсчет записей в запросе. Как это сделать, смотрите раздел «Замечания по компиляции»

MkNull. Отладка SQL

MkNull если True (0x01)- отсутствующие или пустые значения параметров будут заменены на значение *NULL*, как этого требует синтаксис SQL. По умолчанию опция отключена для удобства отладки, так как тогда можно проследить отсутствующие или неверно назначенные параметры. Временно назначить значение параметра MkNull можно так:

```
http://ensen/cgi-bin/is2sql.dll/setup?mknull=true
```

Получить текущее значение параметра можно в теге `<#A name=sys value=OPTMKNULL>`. Тег возвращает значение CHECKED или пустую строку.

В случае пустой строки, взятой в кавычки- “”, можно сказать, что будет делаться подстановка NULL в кавычках: “NULL”

Чтобы избежать это, лучше поступить следующим образом:

- отключить MkNull=0x0
- все строковые и параметры дат передавать в двойных кавычках
- числовые значения передавать так:
...FIELD1 = :PARAMETER1+0

В этом случае, если пользователь не ввел значение в *PARAMETER1*, то будет передан ноль.

ValidateQuoteChar

ValidateQuoteChar Если True (0x1) - принуждает проверять передаваемые параметры форм на предмет наличия кавычек. Числовые и однословные параметры в кавычки не ставятся. ValidateQuoteChar пропускает проверку параметров, у которых уже установлены кавычки одинарные или двойные кавычки. По умолчанию False.

OptimizeDoc

OptimizeDoc True(0x1)-все управляющие символы будут удалены перед отправкой браузеру. Может незначительно увеличить скорость загрузки страниц с большим числом пробелов (выровненных лесенкой), однако искажает содержимое тегов `<PRE>` и аналогичных ему.

Не рекомендуется использовать эту возможность, а оптимизировать собственно сами html страницы и шаблоны .htt

Раздел ResponseCustomHeader

Раздел ResponseCustomHeader содержит дополнительные строки для передачи web- браузеру.

LogFile. LogFunc. LogFuncStart. Журнал активности

LogFile is2sql.dll ведет журнал активности в случае, если установлено правильное значение параметра реестра LogFile. При запуске is2sql проверяет, существует ли этот параметр и задано ли его значение. Затем is2sql.dll пытается открыть указанный файл, чтобы добавлять новые записи. Если файл не существует, is2sql пытается создать файл с указанным именем. Если это невозможно, журнал не ведется. Если файл открывается, запись ведется в следующем формате:

```
TLogStruc = record
remoteIP: String[4*4]; { IP адрес клиента }
t0, { начало по времени }
dt: TDateTime; { продолжительность }
len: Integer; { длина ответа, отправляемого клиенту }
empno: Integer; { номер сотрудника }
lst: String[255]; { хост, алиас, имя, пароль БД, форма, семейство, действие }
end;
```

Замечание: структура *TLogStruc* может быть изменена (см. *isutil.pas*).

Относительный путь имени файла журнала берется по отношению к задаваемому в параметре *PatternPath*.

В зависимости от реализации журнал может вестись в текстовом файле или в базе данных. Во втором случае параметр указывает имя таблицы в БД, которая должна иметь соответствующую структуру. Ведет журнал islog.dll. Если islog отсутствует в каталоге, задаваемом *PatternPath* или в каталоге, где размещен is2sql.dll, он ищется в каталогах, задаваемых системной переменной *PATN*. Если не удастся загрузить функции из islog.dll, журналирование отключается (при этом значение параметра LogFile в реестре не сбрасывается).

LogDLL- задает DLL журналирования (по умолчанию islog.dll). Можно указать полный путь,

тогда не будет делаться поиск в других каталогах.

LogFunc- задает функцию журналирования (по умолчанию logfunc)

LogFuncStart- задает функцию, включающую журнал журналирования (по умолчанию logstartfunc)

DDEEnable True(0x1)- разрешить оповещение приложений используя DDE (запустить сервер DDE)

topic : ddelog, logevt. В текущей версии не применяется.

Формат даты и времени

По умолчанию формат даты и времени используется тот, который установлен в Windows на машине, где запущен is2sql.dll. Если нужно изменить формат представления даты и времени, нужно установить непустые значения параметров:

ShortDateFormat (представление дат)

LongTimeFormat (представление времени)

Также можно изменить LongDateFormat и ShortTimeFormat, но это не будет иметь значения. Для получения сведений о формате, обратитесь к справке Delphi, функции DateToStr, TimeToStr, DateTimeToStr. Для LongTimeFormat рекомендуется HH:mm:ss, для ShortDateFormat - dd.MM.yyyy. DecimalSeparator 1 символ параметра DecimalSeparator задает разделитель дробной части в числах с запятой. Если параметр отсутствует или пустой, по умолчанию берется разделитель из настроек Windows на машине, как запущено is2sql.dll. Рекомендуется установить значение «.» (точка). Если параметр длиннее 1 символа, берется первый символ.

Время актуальности страницы

ExpiresMinutes=<время в минутах>, по умолчанию 0. Строковый параметр, если задано -1, то поле Expires и Last-Modified опускаются в заголовке страницы. Если задан иной параметр, или пустое значение, то в заголовок страницы помещаются два поля. Например, если ExpiresMinutes=5, то поля Last-Modified и Expires в заголовке страницы будут выглядеть примерно так:

```
Last-Modified: Tue, 27 Jun 2000 15:00:32 GMT
```

```
Expires: Tue, 27 Jun 2000 15:05:32 GMT
```

Как видно, ExpiresMinutes задал период актуальности данных равным пяти минутам.

Рекомендуется использовать значение ExpiresMinutes=-1.

CurrencySimple. Формат простого представления валюты

CurrencySimple=True (0x1) - представлять денежный формат (fmt=money) как **100=00**. Этот параметр бинарный, не строковый.

CurrencySimple=False (0x0) - представлять денежный формат (fmt=money) как в соответствии с настройками системы, то есть

100,00 р.

Этот параметр бинарный, не строковый.

shUniqueID. Ключ шифрации

HashUniqueID Ключ шифрации HashUniqueID используется в качестве префикса для генерации шифров. По умолчанию- “;”. Изменение этого параметра может привести к тому, что старые шифры нельзя будет использовать. Если этот параметр отсутствует в реестре, или равен пустой строке, то ключ шифрации будет “;”

RootPwd Пароль администратора

RootPwd Параметр RootPwd содержит пароль администратора. Пароль администратора позволяет при указании его в параметре dll pwd (вместо пароля клиента) получать значения пароля для клиентов в теге <#a name=sys value=_pwd_>.

Пароль администратора указывается в путях /setup и /reload.

Область действия

Настройки действуют на is2sql.dll и частично (параметры локализации= форматы времени и других данных)- на is2sqllog.dll.

Часть 3.

Замечания

ПО

КОМПИЛЯЦИИ

Соглашение

Вы можете свободно распространять это ПО **без изменений** и добавлений в том виде, как Вы его получили, и вносить изменения в код программ для использования в собственных целях при соблюдении следующих небольших условий:

- оставляйте в исходных кодах указание на авторство в том виде и месте, где они находятся в исходных кодах.

Компилятор

Для компиляции Вам потребуется Delphi 5, Delphi 4 или Delphi 3, компоненты сторонних производителей и библиотеки не нужны с одним но: для компиляции приложения без использования BDE (а сейчас доступна только версия для Interbase) нужны компоненты InterBase Express (IBX), включенные в Delphi5 и отсутствующие в ранних версиях.

Некоторые компоненты VCL переписаны, однако Вам не требуется переустанавливать компоненты VCL, так как нужный код уже содержится в исходных кодах.

Рекомендуется не использовать dll журналирования, так как он дублирует журнал web- сервера. Код dll журналирования в отдельном zip файле, возможно, вам потребуется внести изменения в коде.

Первый запуск

Поместите is2sql.dll в каталог /cgi-bin или /scripts. Запустите is2sql.dll, обратившись к нему:

<http://localhost/cgi-bin/is2sql.dll/info?show=list>

из браузера. Если в ответ будет что-то вразумительное, все хорошо. Если нет, то одно из трех:

- web сервер не установлен на компьютере (в указанном примере это ваш компьютер) или вы опечатались при обращении, проверьте url.

- заданный каталог (или виртуальный каталог) не имеет прав на выполнение ISAPI приложений (либо web сервер не настроен или не может в принципе работать с isapi приложениями).

- пользователь web (то есть пользователь, которым запущен web- сервер) не имеет достаточных прав на какие- то действия. Если это так, попробуйте поэкспериментировать с назначением разных прав. В случае Microsoft IIS имя пользователя обычно IUSR_<имя хоста>. Конечно, это не описывает всех возможных ситуаций, но в случае широко известных web серверов проблем не должно возникать. При выборе web сервера обратите внимание, что не все бесплатные web- серверы поддерживают как метод GET, так и метод POST. Иногда метод POST включается при регистрации копии web- сервера.

Файл usr.erc содержит файл кодировки. По умолчанию is2sql.dll ищет его (также и шаблоны, если не указан путь или алиас web- сервера) в своей папке. Редактор кодировок, если надо, скачайте с моей странички из раздела «Софт для пейджинговых операторов».

Настройте BDE- введите нужные вам алиасы БД. Напишите простые шаблоны и попробуйте их. Все настройки делаются в реестре windows.

Рекомендуется не использовать dll журналирования, так как он дублирует журнал web- сервера. Код dll журналирования в отдельном zip файле, возможно, вам потребуется внести изменения в коде, так как код функции журналирования весьма груб.

Возможно, вы пожелаете хранить файлы шаблонов в одном файле. Для этого нужно создать этот файл. Воспользуйтесь простой утилитой etxtcoll. Преимущество этого способа в том, что все необходимые шаблоны единожды загружаются в память, в результате чего сокращается время,

затрачиваемое на доступ к ресурсу web- сервера. Для отладки используйте тег `<#metas>` в файлах шаблонов.

Коллекции шаблонов

Возможно, вы пожелаете хранить файлы шаблонов в одном файле. Для этого нужно создать этот файл. Воспользуйтесь простой утилитой `etxtcoll` (исходники, выполнимый файл). Преимущество этого способа в том, что все необходимые шаблоны единожды загружаются в память, в результате чего сокращается время, затрачиваемое на доступ к ресурсу web- сервера. Минус в том, что внесение изменений в страницы невозможно делать на лету- приходится останавливать сервис и запускать его заново. В общем, такая возможность предусмотрена но не рекомендуется- выигрыш незаметен.

Формат файла коллекции

Файл коллекции представляет набор файлов:

```
<## ИМЯ  Файла 1><CR><LF>
... - содержимое файла 1
<## ИМЯ  Файла 2><CR><LF>
... - содержимое файла 2
<##>
```

`<##>` - признак конца файла

Для создания и редактирования файлов коллекций используйте утилиту `etxtcoll` (описание в разделе «Утилиты»).

Отладка

Для отладки используйте тег `<#metas>` в файлах шаблонов. В него помещается всякая информация о переданных параметрах.

Символы условной компиляции

Прямой доступ к базам данных

Найдите в начале файла `fis2sql.pas` строки

```
{.$DEFINE USE_BDE} // BDE
{$DEFINE USE_IB} // Interbase Express components
{.$DEFINE USE_NCOI} // Oracle 8.x direct access. Requires files you can download
from http://www.torry.ru for free
```

и определите, будете ли вы использовать BDE или прямой доступ к Interbase через Interbase Express, или к Oracle 8.x через компонент NC OCI 8 Дмитрия Арефьева, <http://www.ncom.ru/diman/>

Все эти символы нельзя использовать одновременно. Хотя бы один символ не должен быть «закомментирован» (в примере для этого используется точка перед символом \$)

В зависимости от версии, возможно этого будет недостаточно, тогда задайте соответствующий символ условной компиляции на закладке

Замечание: компонент Interbase Express присутствует в палитре компонент Delphi 5. В случае Oracle нужно установить компонент доступа к oci- функциям NC OCI 8, который вы можете скачать с <http://www.torry.ru> (ссылка DbWare).

Для обеспечения доступа к различным базам данных используйте BDE.

В случае использования компонент прямого доступа к SQL серверам Interbase и Oracle вместо алиасов BDE

CGI вместо ISAPI/NSAPI

Возможно, по каким- то причинам вы захотите отказаться от использования преимуществ ISAPI/ NSAPI. Так как условная компиляция не действует на объявления `library` и `program`, вам нужно изменить их в тексте программы.

Для компиляции CGI .EXE скрипта:

- измените «`library`» на «`program`» в файле проекта `is2sql.dpr`
- выберите в меню `Project|Option` и выставьте в поле «`conditions define`» символ условной компиляции `MKCGI`
- перекомпилируйте весь проект (`Project|Build`)

Возможно, в зависимости от версии Delphi или, точнее говоря, версии VCL, надо изменить тип единственного модуля `TWebModule` на `TDataModule`

Для компиляции ISAPI/NSAPI DLL:

- измените «program» на «library» в файле проекта is2sql.dpr
- выберите в меню Project|Option и удалите символ условной компиляции MKCGI
- перекомпилируйте весь проект (Project|Build)

Дополнительную информацию вы можете найти в комментариях в файле is2sql.dpr.

Глава 1. Утилита etxtcoll

Утилита etxtcoll служит для просмотра, вставки и удаления файлов sql - выражений, параметров и шаблонов из коллекции файлов, используемой как кэш в is2sql. По умолчанию коллекции файлов не используются, и их использование задается в реестре Windows.

Наиболее полезными функциями являются загрузка и выгрузка каталогов и замена слов во всех файлах в коллекции. Эти функции можно использовать для замены во всех sql выражениях имени схемы (для Oracle), имен таблиц и других объектов базы данных.

Для этого нужно создать новый файл коллекции, добавить полностью весь каталог с файлами sql- запросов и шаблонов htt, выполнить команду заменить слово и обратно выгрузить всю коллекцию.

Меню

Операции с файлом коллекции файлов

Файл

Новая коллекция	Создать новый файл с пустой коллекцией
Загрузить коллекцию	Открыть файл коллекции
Сохранить	Сохранить изменения в файле коллекции
Выход	Завершить работу

Замечание: Команды Новая коллекция и Загрузить коллекцию не предлагают сохранить изменения.

Редактировать

Переименовать	Переименовать текущий файл
Удалить файл	Удалить текущий файл из коллекции
Удалить все файлы, кроме	Удалить все файлы, кроме заданных по маске файлов
Заменить слова	Заменить слова на другие во всех файлах в коллекции
(файлы ограничиваются маской)	
Дополнительное окно	Открывает еще одно окно утилиты
Подтвердить изменения	Сохраняет изменения в настройках

Операции

Новый файл	Вставить в коллекцию новый пустой файл
Добавить файл(ы)	Добавить один или несколько файлов в коллекцию
Добавить каталог	Добавить все файлы, включая подкаталоги с файлами,
удовлетворяющие маске файлов. В имени файлов будет записано имя подкаталога, если	
файл лежал внутри подкаталога.	
Выгрузить файл	Выгрузить текущий файл
Выгрузить все	Выгрузить все файлы по маске, включая подкаталоги

Справка

Справка	Справка в HTML (запускает браузер)
О программе	Ссылка на домашнюю страничку утилиты

Замечание: Маска файлов действует не только на имена файлов, но и на имена подкаталогов.

Глава 2. Редактор файлов коллекций перекодировочных таблиц

Описание утилиты

Утилита предназначена для редактирования таблиц перекодировок. usf.epc содержит таблицы: doskoi, doswin, dosiso, koidos, koiwin,... для перекодирования КОИ-8(p), ISO8859-5, Microsoft Windows CP1251.

Скомпилируйте проект Exlat1.dpr, откройте прилагаемые файлы .epc и посмотрите его содержимое. Обратите внимание на то, что один файл содержит несколько таблиц и галочки, отмечающие наличие тех или иных групп символов.

Пример перекодировки символов есть в файле EPGDefs.epc (для различных пейджеров).

Смотрите следующий параграф и файл epcoll.pas для получения информации о структуре файла

.EPS.

Формат файла кодировок

Файл с расширением EPS содержит описания кодировок пейджеров и имеет простой формат:

```
<Заголовок>
<Количество записей N>
<Запись описания пейджера 1>
..
< Запись описания пейджера N>
```

<Заголовок>.

Всегда равно сигнатуре "EP". Двухбайтная сигнатура файла используется для проверки файла. Символы "EP" указывают на порядок следования байт в слове (старший байт - по младшему адресу). Иной порядок задается сигнатурой "MP" (не используйте ее).

<Количество записей N>

Указывает количество записей в файле. Беззнаковое двухбайтное слово. Допустимые значения- 0, FFFFh.

Затем следуют друг за другом записи описания пейджера. Одна запись описывается структурой:

```
TCPDefinition = record
  Name: String[31];           // Название пейджера
  Flags: set of pgt_used..pgt_cyrl; // битовое поле
  BaseCP: Word;               // исходная кодовая страница
  CP: Word;                   // целевая кодовая страница
  Reserved: Byte;             // не используется, всегда 0
  InUse: set of Byte;         // битовое поле
  XLAT: array [0..255] of Byte; // таблица перекодировки символов
  Reserved2: Word;            // не используется, всегда 0
end;                          // 328 байт
```

Примечание: это описание упрощено, для использования в Ваших приложениях Вам нужно будет его немного видоизменить.

Пояснения:

тип String[31] - последовательность, состоящая из символов (длиной не более 31 знака) и предвещающим ее байтом ее длины (строка в стиле Mac). Рекомендуются все терминирующие байты обнулять.

Тип set of pgt_used..pgt_cyrl - битовое поле (один байт длины), в котором заданы общие характеристики кодовой страницы пейджера:

Бит	Бит 0	Бит 1	Бит 2	Бит 3	Бит 4	Бит 5
Мнемоника	Pgt_used	Pgt_num	pgt_latU	pgt_latl	pgt_cyrU	pgt_cyrl
	Загружать	цифры	заглавные	строчные	КИР	кир

Биты 6 и 7 всегда не выставлены (0).

Name- уникальное название перекодировочной таблицы. Вы можете использовать любые символы, знаки кириллицы должны быть из кодовой страницы CP1251. Не рекомендуется использовать знаки препинания и пробелы. Для первого символа рекомендуется использовать только буквы. Предполагается, что пейджера чувствительно к регистру.

Примечание: Если первый символ "*" (звезда), запись может содержать сведения, отличные от описываемого.

Flags- битовое поле описано выше

CP- номер кодовой страницы. Это поле всегда равно 0 или 1251. Вы можете произвольно присваивать номера для группирования описаний пейджеров в группы по каким- либо иным признакам, используя это поле.

InUse- битовое поле из 256 бит, каждый из которых соответствует одному из 256 символов кодовой страницы. Выставленный бит означает, что соответствующий ему символ используется.

XLAT- массив кодов символов, например, 33-й элемент массива (с номером 32) в кодовой странице 1251 должен содержать код пробела 32. Если задан код 0, то считается, что символ не подлежит передаче.

Reserved и Reserved2- зарезервированы, всегда ноль.

Примечание: Даже если в битовом поле InUse определено то, что какая- то буква не используется,

рекомендуется в массиве XLAT задать этой букве код заменяющего его символа. Это не касается управляющих символов (и вообще всех не алфавитно-цифровых символов), для которых можно задавать код 0.

Глава 1. Использование кириллицы в Interbase

Во первых, нужно создать базу данных, указав правильную кодовую страницу для базы данных в целом, и задав ее же для использования в сортировках.

Установите сессию

Запустите `wisql` или `isql` и установите кодовую страницу:

```
SET NAMES WIN1251;
```

Создание базы данных

Создайте файл базы данных с использованием выражения `DEFAULT CHARACTER SET WIN1251` для определения кодовой страницы по умолчанию:

```
CREATE DATABASE "/src/equiz/db/equiz.gdb"  
PAGE_SIZE 8192  
USER "SYSDBA"  
PASSWORD "masterkey"  
DEFAULT CHARACTER SET WIN1251;
```

Создание доменов

Использование доменов для задания типов, на мой взгляд, значительно упрощает на стадии определения используемых типов в полях таблиц, особенно в случае необходимости указания `COLLATE SEQUENCE`.

Например, следующие декларации

```
CREATE DOMAIN ICHAR  
AS VARCHAR(1) COLLATE WIN1251;
```

```
CREATE DOMAIN ISO  
AS VARCHAR(8) COLLATE WIN1251;
```

```
CREATE DOMAIN IS1  
AS VARCHAR(32) COLLATE WIN1251
```

создают домены `ICHAR`, `ISO`, `IS1` как строковые типы с заданной кодовой страницей, используемой для сортировки.

Теперь можно создавать таблицы с указанием этих доменов:

```
CREATE TABLE CERTIFICATE(  
  CSERIAL IS1,  
  CORDER IS1,  
  CSIGNED ICHAR  
  ...
```

Алфавитный указатель

Символы

<#!@>

Описание, Часть ELSE шаблона 8

Пример 3 24

<#/@>

Описание, Окончание шаблона 8

<#@>

Описание, тело шаблона 8

<#a>

Установка новых значений 17

И

Изменение значения CONTENT-TYPE 17

Изменение типа контента 32. *См. также* Content-Type: Изменение типа возвращаемой страницы

К

Каталог по умолчанию

Тег <#a value=sys value=patternpath newvalue=> 17

Квалификаторы семейств 28

Коллекции шаблонов 42

Компрессия

zLib 12. *См. также* Путь: /гес или /г: decode=z

Константы

булевы значения 13

Копирование параметров

Тег <#f> 20

Н

Настройка

CacheConnections 13

CurrencySimple 20

DBConnectFailForm 12

MaxThreads 13

Настройки

ср

Указание файла перекодировок, ECCNAME 17

О

Отладка

метод POST

тег <#metas> 16

П

Параметр

ср 12

dbb 12

Пример 3 24

form

Пример 3 24

key 12, 13

key пароль администратора 12. *См. также* Путь: /setup

смена пароля 12. *См. также* Путь: /setup

user 12

Параметры форм 21

Передача параметров из формы в форму 29

Путь

/info 12

/res или /r 12

decode=z 12

Пример фонового звука 32

/reload 13

/setup 12

mknulл форсирование NULL 13

/show или /s 10

С

Совместимость

Borland Database Engine (BDE)

теги <#e> и <#t> 29

FrontPage 26

тег <#x> 21

Interbase

теги <#e> и <#t> 16, 29

Сшивающий знак «+» 31

Как поместить знак "+" в конце значения параметра 22

Описание, использование для операторов sql 22

Т

Тег

<#a name=ip> 18

переменные web- сервера 18

<#a>

<#a name=sys value=pageline> 28

<#a name=sys value=qty> 27

вычисление хеш- функции 17

информация о шлюзе 16

отправка почтовых сообщений 17

<#c>

Инкремент параметров 18

Копирование параметров 18

Создание параметров 18

<#e>

Пример 3 28

<#f> 20

Описание, тег поля 8

Пример 2 23

Пример 3 24

<#h> 19

Пример 4 29

<#metas> 12

<#p>

Описание 15

Описание, параметр 8

<#t>

Пример 5 31

<#t>

Пример 2 23

Пример 3 24

<#x>

Описание, тег для совместимости с FrontPage 26. См. также Совместимость: FrontPage

Ф

Формат

внешняя форматирующая функция

Тег <#f> 20

Дата/время

Тег <#f> 20

денежный

Тег <#f> 20

тег <#p> 15

Ш

Шаблон htt 8. *См. также* <#!@> Часть ELSE шаблона; <#/@> Окончание шаблона; <#@> Тело шаблона

Иностранные термины

BDE Administrator 34

CGI 42

Content-Type

Изменение типа возвращаемой страницы 17

ср, параметр 11

db, параметр 10

first, параметр 11

form, параметр 10

forminvalid, параметр 11

hash, параметр 11

ISAPI 42

key, параметр 10

last, параметр 11

push, параметр 11

pwd, параметр 11

step, параметр 11

user, параметр 10

