

Ну что ж, приступим. Для начала определимся с той информацией которую нам необходимо представить в самих новостях. Это могут быть: заголовок, краткий анонс, тело новости (то, что будет отображаться на экране после перехода по ссылке на новость, чтобы не загромождать экранное пространство большими текстами), дата, автор.

Итак, создаем базу в Access. Назовем ее Articles.mdb. В ней мы должны создать таблицу articles, которая будет содержать следующие поля:

Title	Memo
Anons	Memo
Body	Memo
Author	Memo
Date	Date/Time
Log	AutoNumber

Теперь нам необходимо создать файл, с помощью которого мы сможем получить доступ к базе articles (хранящейся на сервере) через обыкновенный браузер.

Любой язык программирования имеет свои инструменты доступа к данным. ASP — не исключение. Остановимся на этом поподробнее.

Вот основная процедура доступа к данным, описывающая необходимые переменные, их параметры, а также тип данных:

```
<%
Function Incdb_GetConnection( sConnString )
Dim oConn
Set oConn = Server.CreateObject("ADODB.Connection")
oConn.Open sConnString
Set Incdb_GetConnection =
oConn
End Function
%>
```

В нашем примере файл articles.asp имеет следующие строчки кода:

```
accessdb="articles"
cn="driver={Microsoft Access Driver (*.mdb)};"
cn=cn & "dbq=" & server.mappath(accessdb)
Set rs = Server.CreateObject("ADODB.Recordset")
rs.Open sql, cn
```

Важное замечание!

Скрипты, написанные средствами ASP, — это, как и документы HTML, простые текстовые файлы, имеющие расширение ASP.

В теле файла должны содержаться операторы <% %>, показывающие, где интерпретатор должен начать обработку процедур, а где должен ее закончить. В других местах документы могут содержать обыкновенные тэги языка HTML, которые отвечают за вывод, форматирование, графическое оформление страниц и т. д.

Итак, мы получили доступ к базе через web, теперь мы сможем совершать с ней любые операции: чтение, запись, удаление, добавление и т. д. Давайте инициализируем переменные, которые будут соответствовать полям нашей базы данных, чтобы после открытия использовать переменные полей в структуре нашей программы.

```
u_action=request.querystring("u_action")
u_title=replace(trim(request.form("u_title")))
u_anons=replace(trim(request.form("u_anons")))
u_body=replace(trim(request.form("u_body")))
u_author=replace(trim(request.form("u_author")))
u_log=request.querystring("u_log")
```

Теперь, когда переменные определены и база данных открыта, можно создать SQL-запрос на выборку из базы данных. Если вам приходилось работать с приложениями, созданными в MS Access, то вы уже знаете основные принципы построения SQL-запросов. В нашем примере запрос может иметь следующий вид:

```
sql = "insert into articles (title,anons,body,author)"
sql = sql & " values( '" & u_title & "', '" & u_anons & "', '" & u_body & "', '" & u_author & "' )"
```

Результаты SQL-запроса необходимо отобразить на экране, для этого необходимо «сгенерить» HTML-документ, содержанием которого будут созданные нами переменные.

```
'Начало вывода HTML
<html>
<%
'Показываем текст, сообщающий о том, что новости были добавлены (эта процедура понадобится нам немного позже)
if u_action="insert" then %>
'Ваши новости были добавлены<br>
<% else
'Выводим содержимое базы данных %>
<% do while not rs.eof and counter < 6
'Устанавливаем количество показываемых новостей на экране
counter=counter+1 %>
<% if u_action="display" then %>
```

Важное замечание!

Важность процедуры открытия базы данных сложно переоценить. В случае если вы допустили ошибку, сервер вернет сообщение и приложение не будет работать. Более того, если в структуре дизайна вашей страницы вы используете скрипты ASP как небольшую часть (в нашем примере — новостная лента), из-за ошибки процедуры открытия базы пользователь может не увидеть и остальных частей документа! Во избежание подобных неприятностей приложения необходимо тщательно тестировать.

Тип	Случаи использования
System DSN	<ul style="list-style-type: none"> ▶ Все используемые базы данных устанавливаются как системные и регистрируются Start -> Setting -> Control Panel -> Administrative tools -> Data Source (ODBC) -> System DSN Данный вид подключения помогает получить абсолютные пути к данным, но в случае когда вы не имеете своего web-сервера, а пользуетесь услугами хостинга, эта функция использоваться не может, поскольку ISP не дают пользователям доступа к datasource из соображений безопасности
DSN-less ODBC	<ul style="list-style-type: none"> ▶ При отсутствии доступа к System DSN ▶ При отсутствии уверенности, что на сервере установлен компонент ADO 2.0 или более старшей версии
DSN-less OLEDB	<ul style="list-style-type: none"> ▶ При отсутствии доступа к System DSN ▶ Если вы уверены, что на сервере установлен компонент ADO 2.0 или более старшей версии ▶ Если вы хотите иметь максимальные возможности в управлении данными

▲ Табл. 1. Способы установки связи с базами данных

```

» <a href="<%= script_name %>?u_action=display&u_log=<%=
rs("log")%>" target="_blank"><%= rs("title") %></a> <br>
<%= rs("anons") %><br>
<% else %>
<%= rs("title") %><br>
...
<% end if %>

```

Результат работы данного кода представит на экране новостную ленту из шести новостей, содержащих заголовки новости (title (u_title)) и анонс (anons (u_anons)). Заголовок новости будет иметь вид гиперссылки, при нажатии на которую откроется новое окно, содержащее тот же заголовок новости и ее полный текст (body (u_body)). Если вы внимательно изучите приведенный листинг, то заметите, что очень просто видоизменить как оформление вывода новостей, так и сам вывод, добавив, например, автора новости — или вместе с заголовком, или после анонса.

На разработку подобного скрипта мы потратили достаточно мало времени, но получили работоспособное и динамическое приложение (занимающее всего 2 Кбайт) с высокой скоростью обработки. Теперь, если администратор базы данных захочет добавить новости, он сможет это сделать в таблице базы данных Access. Новости будут автоматически отображаться на новостной странице сервера, причем старые новости будут «самостоятельно» убираться с экрана при появлении новых записей. Поскольку HTML-текст вывода новостей — это простой текст, то данный скрипт способен легко вписаться в любой дизайн в том месте, где в тело страницы вы вставите тэг `<!--#include="articles.asp"-->`.

Создавая динамические приложения, логично предусмотреть возможность администрирования ресурсов через web без необходимости работы непосредственно на сервере. Для этого современные web-мастера создают инструменты удаленного администрирования. Создадим и мы возможность добавлять новости в нашу базу данных через web-форму. Уверен, что работа с формами в документах HTML не представляет для вас никакого труда. В нашем примере это выглядит так:



▲ Apache в действии

Важное замечание!

Любые страницы, в которых есть операторы ASP (например, такие как include) должны иметь расширение asp, для того чтобы интерпретатор понимал, что в документе есть процедуры ASP, которые необходимо обработать.

```

<% if u_action="add" then
    'Это пользовательская функция, отвечающая за добавление
    новостей через экранную форму, созданная с использованием
    JavaScript %>
<script Language="JavaScript">
<!--
function Blank_TextField_Validator(form)
{
if (form.u_title.value == "")
    'Создаем поле для заголовка новости
    {
    alert("Please fill in the title field.");
    'Выдаем ошибку, если поле осталось незаполненным (для про-
    верки заполнения формы мы и воспользовались JavaScript)
    form.u_title.focus();
    'Возвращаем пользователя в поле до тех пор, пока он не впи-
    шет в поле текст, потом осуществляем переход на другую за-
    писку
    return (false);
    }
...
    'И так по всем полям формы
    //-->
</script>
<form method="post" action="<%= script_name %>>
    'Выбираем метод отправки новостей на сервер
    'Строим форму на странице (обыкновенные средства HTML)
    'Используем тэги <form>, </form>, <input type> и <textarea>
    'Далее проверяем, хочет ли пользователь добавить еще одну
    новость
    'Или вернуться к просмотру новостей на странице
    'Показываем сообщение, что новости были добавлены (об
    этом мы говорили выше)
    'Показываем на экране навигационные ссылки
<p><a href="<%= script_name %>">View Recent Articles</a><br>
<a href="<%= script_name %?u_action=add">Add Your
Article</a>
<% end if %>

```

» На этом документ `article.asp` заканчивается. Подведем итоги. В одном файле мы имеем: средства работы с базой данных MS Access, средства просмотра (и встраивание скрипта) содержимого базы данных, шаблон внешнего вида документа, средства добавления новостей через web-интерфейс непосредственно в базу данных на сервере и средства навигации по нашему скрипту. И все это буквально в одном файле размером 3 Кбайт со всеми необходимыми комментариями в тексте!

Полный текст документа (файл `article.asp`), связывающий все описанные модули воедино, вы найдете на Chip CD в разделе Bonus / «Листинги и материалы».

Теперь хотелось бы сказать несколько важных слов относительно типов данных, используемых при создании скриптов на языке ASP. Прежде всего необходимо остановиться на самом типе данных ODBC (Open DataBase Connectivity). Это открытый стандарт данных, позволяющий пользователю получать доступ к данным фактически без каких-либо знаний о типе используемой базы данных. Поэтому, например, скрипты ASP имеют возможность работы с практически любыми базами данных (это свойство языка позволит вам без особого труда «дать вторую жизнь» вашим «древним» базам данных и обеспечить управление ими через ваш сервер в сети). Вы имеете возможность использовать всю мощь функций при работе с ODBC: буферизацию, возвращение типов данных, получение сведений о источниках данных, перенаправление вызова, паролирование и разграничение доступа.

Есть три основных способа установки связи в ваших базах данных (табл. 1).

Более подробную информацию вы можете получить, например, на сайте <http://docs.rinet.ru> или из документации по использованию ASP.

В нашем примере мы использовали самый простой тип данных, при котором приложение будет работать везде, где есть поддержка ASP на сервере. Наш скрипт позволяет любому пользователю добавить любую новость в вашу базу данных. Никаких особых прав администратора не требуется. В случае построения

более сложных приложений это, конечно, недопустимо, поскольку содержанием должен управлять только администратор или пользователи, имеющие соответствующие привилегии. В системах UNIX-серверов можно разграничить доступ к различным частям своего сервера (файлам, директориям, процедурам) на уровне permissions, что достаточно проблематично при использовании MS Windows Server (устанавливать permissions на каждый файл позволяет функция `xcac1` входящая в Resource Kit). Это, например, затрудняет работу на Windows серверах некоторых Perl-скриптов.

С помощью ASP можно создавать привилегированный доступ к страницам и разделам вашего сервера. Вам достаточно создать базу данных следующего содержания:

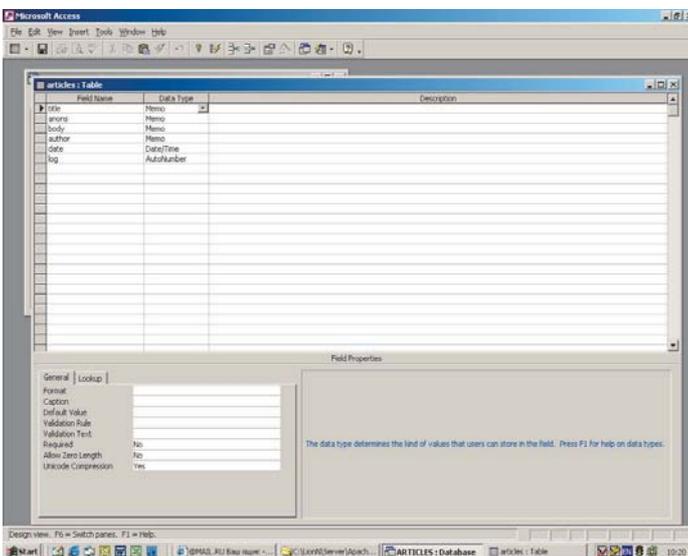
MemberID	AutoNumber
LoginName	Text
LoginPassword	Text

А также процедуры, обеспечивающие обработку данных (файл `login.now.asp`):

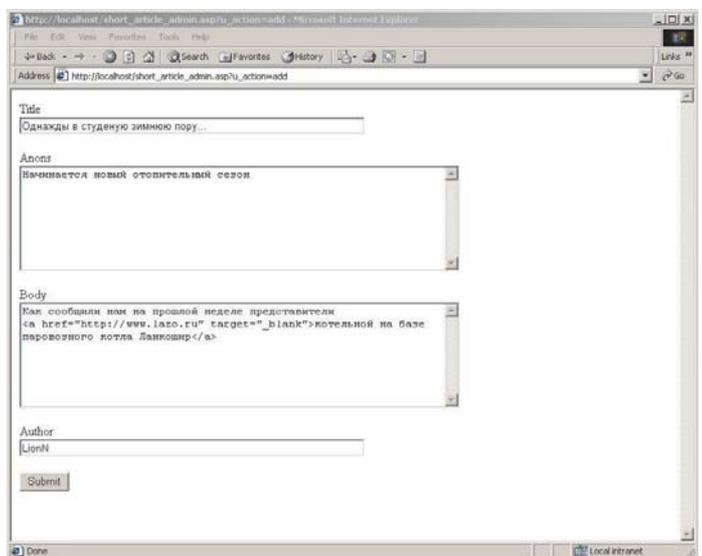
```
<%
Dim strSQL           'Структура языка запросов
Dim objConn          'Подключение к базе данных
Dim objRs            'Запись
Dim strLoginName     'Имя пользователя
Dim strLoginPassword 'Пароль
Dim lngMemberID      'Уникальные номер пользователя

'Получение данных о имени пользователя и пароле с экранной
формы
strLoginName = Request.Form("LoginName")
strLoginPassword = Request.Form("LoginPassword")

'Открытие базы данных
Set objConn = Server.CreateObject("ADODB.Connection")
Set objRs = Server.CreateObject("ADODB.Recordset")
objConn.Open "DRIVER=Microsoft Access Driver (*.mdb);DBQ=" &
```



▲ База данных Articles



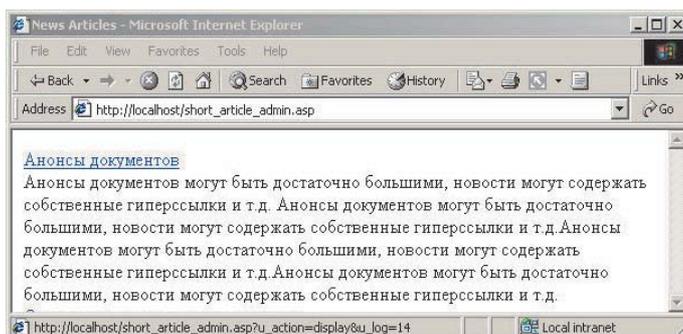
▲ Новость добавляется просто — одним нажатием кнопки

```

» Server.MapPath("Members.mdb")
   'Поиск пользователя в базе
strSQL = "SELECT MemberID FROM Members WHERE LoginName = '" &
strLoginName & "' AND LoginPassword = '" & strLoginPassword & "'"
Set objRs = objConn.Execute(strSQL)
   'Если пользователя с таким именем и паролем в базе не содер-
   жится, выводится сообщение об ошибке
If objRs.EOF Then
   Response.Write("Login Failed")
Else
   lngMemberID = objRs(0)
   Response.Write("Login Succeeded, MemberID = " & lngMembe-
   rID)
End If
   'Если пользователь найден – выводим его имя и ID
...
   'Тут описываем процедуры перенаправлении пользователя на за-
   щитенные страницы
...
Set objRs = Nothing
objConn.Close
Set objConn = Nothing
%>

```

В результате работы данного скрипта вы сможете попасть в администраторскую зону для управления своим ресурсом. Теперь осталось каким-либо образом соединить два скрипта в единое приложение, и мы получим полноценный менеджер новостей для вашего сайта.



▲ Отображение новостей в IE

Одним из вариантов может быть следующий путь. Создаем копию скрипта article.asp с именем admin_article.asp и копируем его в папку admin, устанавливая на папку соответствующие permissions. Туда же помещаем базы данных, для предотвращения несанкционированного доступа.

Внимание! Не забывайте о пути к базе данных в своих скриптах, если базы не зарегистрированы как System DSN.

В файле article.asp убираем все строчки, относящиеся к добавлению информации в базу – этот скрипт теперь будет отвечать лишь за показ новостей на экране. Заносим в базу members.mdb данные о пользователях и помещаем в файл login.now.asp процедуру перехода к файлу admin_article.asp:

```
<!--#include="admin/admin_articles.asp"-->
```

Менеджер новостей готов!

■ ■ ■ Леонид Николаев

Support

Необходимые замечания

► Где взять/установить/настроить ASP? Сразу оговоримся, что наш сервер работает под управлением MS Windows, поэтому вероятнее всего установлен IIS (Internet Information Server) – серверное решение от Microsoft. В данном случае ASP входит в комплект поставки и его можно установить/настроить в момент установки IIS или с помощью утилиты Server Manager. Если в качестве web-сервера вы используете Apache (собственно, как и сам автор этих строк), вы имеете возможность выбора. Первый путь: вы ищете в Сети динамические библиотеки apache-asp и редактируете свой файл http.conf, так, чтобы ASP загружался как сервис. Второй путь: вы используете пакет ChilliSoft ASP, который самостоятельно установит все необходимое для работы ASP под Apache.

► Создание динамических страниц с использованием ASP – это процесс написания

несложных скриптов, главными преимуществами которых являются: использование компоненты ADO (ActiveX Data Object) для доступа к данным и возможность использования сценариев написанных на других языках (например Perl) внутри одного документа ASP.

► В Сети есть огромное количество бесплатных и условно бесплатных скриптов которые реализуют возможность публикации новостей на сайтах без редактирования HTML вручную. Обычно скрипты post news пишут на PHP или Perl. Настройка и работа этих интерпретаторов на серверной платформе MS Windows не представляет собой сложности, но все же, думается, использование «родных» (однородных) технологий – более правильный путь.

► Работа новостных скриптов подразумевает хранение информации в базах данных (иногда информация хранится в простом

текстовом файле с каким-либо символом разделителя). В случае с PHP и Perl это будет скорее всего MySQL или PostgreSQL. Администрирование этих баз данных довольно хлопотливое занятие, особенно для новичков в интернет-программировании. Облегчают этот труд те же PHP- или Perl-скрипты, которые тоже необходимо найти в Сети и установить. При работе с ASP вы сможете использовать и перечисленные выше базы данных, и базу данных созданную средствами офисного пакета MS Access, начиная с версии 97 (Возможно ранние версии этого продукта тоже подходят для этой цели, но мною они протестированы не были. Более того, с 97 версии Access формы можно сохранять как ASP-файлы). Безусловно, ASP имеет все необходимое для работы и с текстовыми файлами, хотя мы остановились на примерах, работающих с базами данных.