Active Server Pages (ASP)

Christian Zahler

Active Server Pages sind eine Technologie, mit der Code serverseitig ausgeführt werden kann. ASP ist kein Gestaltungselement! Immer, wenn ein Browser eine Datei mit der Erweiterung ***.ASP** von einem Webserver anfordert, wird dort die ASP-Engine **ASP.DLL** gestartet und die Datei ausgeführt. An den Browser wird eine reine HTML-Datei zurückgeliefert!

Voraussetzungen

- Windows NT Server 4.0 mit Service Pack 3 und Windows NT 4 Option Pack (inkludiert Internet Information Server 4.0); wichtig: ASP-Erweiterungen müssen installiert sein!

  Weniger optimal sind IIS 2.0 oder IIS 3.0 auf NT Server 4.0. Als Entwicklungsumgebung kann auch Windows 95/98 mit dem Personal Web Server verwendet werden.

Werkzeuge zur Erstellung einer ASP-Datei

- Editor (selbst eintippen)
- Microsoft Visual InterDev
- Macromedia Drumbeat

WWW-Adressen zum Thema ASP

`www.wrox.com`

ist ein Diskussionsforum mit vielen (englischsprachigen) Informationen zum Thema Web programmierung und ASP.

ASP-Code ordnet sich prinzipiell dem HTML-Standard unter. Serverseitiger Code ist erkennbar an den speziellen Tag-Begrenzungen

```
<% Code %>
```

Die Programmierung von ASP-Code erfolgt üblicherweise in Visual Basic-Script (VBScript). Die Sprache ist in der 1. Zeile des HTML-Codes anzugeben (es sollte auch am IIS VBScript als Standardsprache eingestellt sein!):

```
<%@ LANGUAGE="VBSCRIPT" %>
```

Beispiel 1: Heutiges Datum

```
<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
<html>
  <head>
    <title>Datum, Zeit, ...</title>
  </head>
    <body>
    Heute ist
    <% =FormatDateTime(now(),vbLongDate) %>
  </body>
</html>
```

Eingebaute Objekte

Fünf Objekte sind in VBScript bereits eingebaut und ersetzen die herkömmliche Ein-/Ausgabesteuerung von Visual Basic:

- `Request`: fordert Informationen vom Browser bzw. enthält Informationen, die von einem HTML-Formular übertragen werden
- `Response`: sendet Informationen zum Browser, vor allem zur direkten Ausgabe von Text aus VBScript heraus
- `Server`: steuert die ASP-Umgebung und dient beispielsweise zur Objekterzeugung und zur Kontrolle von TimeOut-Zeiten
- `Err`: steuert Laufzeitfehler
- `Session`: speichert Informationen über die aktuelle Sitzung (Achtung! Cookies werden verwendet!)
- `Application`: verteilt Informationen zwischen den verschiedenen Nutzern einer Sitzung
- `ObjectContext`: steuert Transaktionen, die vom Microsoft Transaction Server (MTS) verwaltet werden

Beispiel zum Request-Objekt

```
<%@ LANGUAGE="VBSCRIPT" %>
<% option explicit %>
<HTML>
  <HEAD>
    <TITLE>
      ASP: Beispiel 60
    </TITLE>
  </HEAD>
  <BODY>
  <% =Request.ServerVariables("PATH INFO")%>
  </BODY>
</HTML>
```

Beispiel zum Server-Objekt

```
<%@ LANGUAGE="VBSCRIPT" %>
<% option explicit %>
<HTML>
  <HEAD>
    <TITLE>
      ASP: Beispiel 62
    </TITLE>
  </HEAD>
  <BODY>
    <% dim pi %>
    <% pi =
Request.ServerVariables("PATH INFO") %>
    virtueller Pfad: <% =pi %> <BR>
    physisches Verzeichnis am Server: <%
=Server.MapPath(pi) %>
  </BODY>
</HTML>
```

Beispiel zum Response-Objekt: Auslesen der Server-Variablen

```
<%@ LANGUAGE="VBSCRIPT" %>
<% option explicit %>

<HTML>
  <HEAD>
    <TITLE>
      ASP: Beispiel 64
    </TITLE>
  </HEAD>
  <BODY>
    <H1>Server Variables Collection</H1><BR>
    <% dim item %>
    <% dim iloop %>
    <% for each item in
request.servervariables
        for iloop = 1 to
request.servervariables(item).count
          response.write (item & " = " &
request.servervariables(item)(iloop)) %>
          <BR>
        <% next %>
    <% next %>
  </BODY>
</HTML>
```

Es gibt eine Reihe von Environment Variablen, die mit dem Server gestartet werden,

- `QUERY_STRING` Diese Variable enthält den Rückgabestring mit Aufbau wie oben beschrieben.
- `CONTENT_TYPE` Diese Variable gibt den MIME-TYPE an, welche den Typ der Daten angibt (z.B. **`text/plain`**, **`text/html`** oder irgendein Graphiktyp)
- `CONTENT_LENGTH` Gibt die Länge des Datenstrings in Zeichen an.
- `PATH_INFO` Gibt zusätzliche Pfadinformationen an.
- `PATH_TRANSLATED` Gibt zusätzliche Pfadinformationen in einer endgültig verwendbaren Form übersetzt an.
- `REMOTE_ADDR` Gibt die IP-Adresse des Clients an.
- `REMOTE_HOST` Gibt den Namen des Clients an (HOSTNAME).
- `REMOTE_USER` Gibt den authentischen Benutzernamen des Clients an (jeder Benutzer, der mit einem WWW-Server arbeitet, der wird auf diesem Server als der Benutzer „wwwrun" angemeldet – nicht vergessen, entsprechende Berechtigungen für diesen Benutzer zu setzen).
- `REMOTE_IDENT` Gibt den Benutzernamen so an, wie er vom `IDENT_PROTOKOLL` (definiert in RFC 931) vorgegeben wurde.
- `AUTH_TYPE` Art der verwendeten Bestätigung (in welcher Form erfolgt die Autorisierung -> Sicherheit).
- `REQUEST_METHOD` Gibt den Namen der Methode an (GET oder POST).
- `SCRIPT_NAME` Gibt den Namen des CGI Programmes an,
  z.B. „**`/cgi-bin/script.cgi`**" (Angabe also relativ zu „cgi-bin").
- `SERVER_PORT` Standardanschluss (Port 80 für einen WWW-Server)
- `SERVER_PROTOCOL` Name des vom Server verwendeten Protokolls (z.B. HTTP/1.0).

  Diese Variablen bezogen sich auf den Client und dem Weg zwischen Client und Server. Folgende Variablen beziehen sich nur auf den Server:

- `HTTP_USER_AGENT` Name des anfragenden Servers (z.B. Mozilla/1.1M Windows, 16 Bit).
- `HTTP_ACCEPT` Enthält eine Liste von MIME-Typen, die der Client verarbeiten kann.
- `HTTP_REFERER` Enthält die URL der Form/des Dokuments, das die Quelle der Anfrage war
  (z.B. http://server/page.html).

Komponenten

Komponenten sind – ebenso wie die eingebauten Objekte – Sammlungen von Methoden und Eigenschaften. Der Unterschied ist, dass Komponenten auf eine ganz bestimmte Anwendung zugeschnitten sind. Mit IIS 4 werden neun Komponenten geliefert:

- **Werbebanner-Komponente (Ad Rotator)**: wird verwendet, um einen rotierenden Banner zu erzeugen und zu verwalten.
- **Browsereigenschaften-Komponente (Browser Capabilities)**: unterstützt den Programmierer bei der Auswertung der Browsereigenschaften. Damit ist beispielsweise die Anpassung der HTML-Ausgabe an die Möglichkeiten der Browser möglich.
- **Inhaltsverbindungs-Komponente (Content Linking)**: vereinfacht die Navigation zwischen den Seiten.
- **Zähler-Komponente (Counter)**: Damit können Sie sehr einfach Zähler aufbauen, die die Hits auf Ihrem Web zählen.
- **Seitenzähler-Komponente (Page Counter)**: ergänzt die Zähler-Komponente um Funktionen, mit denen Hits auf bestimmte Seiten des Webs registriert werden können.
- **Inhaltsrotations-Komponente (Content Rotator)**: Basis für einen News-Service. Im Gegensatz zur Banner-Komponente wird HTML-Text ausgegeben.
- **Zugriffstest-Komponente (Permission Checker)**: verwaltet geschützte Seiten. Damit lassen sich Links nicht nur schützen, sondern auch verstecken.
- **ActiveX Data Object (ADO)**: spezielle Komponente, die den gesamten Zugriff auf Datenbanken beinhaltet.
- **Dateizugriffs-Komponente (File Access)**: Objekte, mit denen auf die Datei- und Verzeichnisstruktur des Servers zugegriffen werden kann.

Die meisten Komponenten stammen aus der Klasse **MSWC (Microsoft Web Components)** und erweitern die eigentliche Funktionalität von ASP. Um eine Komponente zu benutzen, muss eine Instanz erzeugt werden. Dazu verwendet man die Methode `CreateObject()` des Server-Objekts:

```
    <% set ad =
Server.createobject("MSWC.AdRotator") %>
```

Wichtig: Eine solche Instanz (ad) ist nur innerhalb der Seite verwendbar, die das aufrufende Skript enthält. Wird die Seite beenden, so wird auch das Objekt gelöscht! (Typisch dafür ist die Verwendung der Anweisung SET).

Pagecounter

Achtung! Erfordert die Installation des „Internet Information Server 4 Resource Kit"!

```
<%@ LANGUAGE="VBSCRIPT" %>
<% option explicit %>
<html>
  <head>
    <title> Pagecounter </title>
  </head>
  <body>
    <% Dim sz %>
    <% Set sz =
Server.createobject("IISSAMPLE.PageCounter")
%>
    <% sz.PageHit %>
    Anzahl der Zugriffe auf diese Seite:
    <% =sz.Hits %>
  </body>
</html>
```

Hinweis: Beachten Sie, dass die Page-Counter-Komponente standardmäßig nicht unter der Klasse „MSWC" installiert wird, sondern unter der Klasse „IISSAMPLE"!

Globales Counters-Objekt:

Der Zähler wird hier in der Datei **GLOBAL.ASA** definiert. Die folgende Zeile bewirkt die Erzeugung eines Zähler-Objekts am Server:

```
<OBJECT RUNAT="Server" SCOPE="Application"
ID="zaehler"
PROGID="MSWC.Counters"></OBJECT>
```

Der Code für die Erhöhung des Zählers ist in einem normalen ASP-Skript enthalten. Beachten Sie: Mit der globalen Definition können mehrere Zähler erzeugt und mitgeführt werden. Im folgenden Skript wird ein Zähler „Seiten" erzeugt.

```
<%@ LANGUAGE="VBSCRIPT" %>
<% option explicit %>
<html>
  <head>
    <title> Globales Counters-Objekt
</title>
  </head>
  <body>
    Anzahl der Zugriffe auf diese Seite:
    <% =zaehler.Increment("Seiten") %>
  </body>
</html>
```

Der aktuelle Stand des Zählers wird in der Datei **COUNTERS.TXT** gespeichert, die üblicherweise im Verzeichnis **C:\WINNT\system32\inetsrv** abgelegt wird. Inhalt der Datei **COUNTERS.TXT** für obiges Beispiel:

```
Seiten:3458
```

Beispiel: Ad Rotator

Diese Komponente hat nur eine einzige Methode – `GetAdvertisement` – mit der der nächste Banner zur Anzeige gebracht wird.

```
<%@ LANGUAGE="VBSCRIPT" %>
<% option explicit %>
<HTML>
  <HEAD>
    <TITLE>
      ASP: Beispiel 70
    </TITLE>
  </HEAD>
  <BODY>
    <% dim ad %>
    <% set ad =
Server.createobject("MSWC.AdRotator") %>
    <% ad.clickable = false %>
    <%
=ad.GetAdvertisement("/ilgscript/adrot_70.tx
t") %>
  </BODY>
</HTML>
```

Die Datei **adrot_70.txt** enthält Informationen zu den Werbebilden und die zugehörigen Links:

```
width 460
height 60
border  0
*
//ad_1.gif
http://www.microsoft.com
Microsoft USA
50

//ad_2.gif
http://www.hill-woltron.com
Personalberater HILL Woltron
50
```

1. Zeile: Bannergrafik
2. Zeile: Link zum Werbekunden
3. Zeile: Alternativer Text (so wie bei `<IMG SRC="..:" ALT="...">`
4. Zeile: Relative Häufigkeit der Banneranzeige in Prozent

Beispiel: Ad Rotator Nr. 2

```
<%@ LANGUAGE="VBSCRIPT" %>
<% option explicit %>
<HTML>
  <HEAD>
    <TITLE>
      ASP: Beispiel 72
    </TITLE>
  </HEAD>
  <BODY>
    <% dim ad %>
    <% set ad =
Server.createobject("MSWC.AdRotator") %>
    <% ad.clickable = true %>
    <%
=ad.GetAdvertisement("/ilgscript/adrot_72.tx
t") %>
  </BODY>
</HTML>
```

Die Datei **adrot_72.txt** enthält zusätzlich den Hinweis, auf welche Datei die Ausgabe geschrieben werden soll:

```
REDIRECT /ilgscript/asp_redir_72.asp
 460
 60
  0
*
//ad_1.gif
http://localhost/default.htm
lokaler Server - Homepage Serverroot
50

//ad_2.gif
http://www.hill-woltron.com
Personalberater HILL Woltron
50
```

Die Datei **asp_redir_72.asp** enthält nur eine einzige Anweisung:

```
  <%
Response.Redirect(Request.QueryString("url")
) %>
```

Variablenübergabe an andere Seiten

GET-Methode des Form-Tags in der HTML-Seite/QUERY_STRING-Servervariable: Die Daten werden an eine Servervariable mit dem Namen „`QUERY_STRING`" übergeben, die programmtechnisch (über das `Request`-Objekt) ausgelesen werden kann.

Beispiel

```
http://server/pfad?elementname=wert&ele
mentname=wert1+wert2&
elementname=wert
```

Bedeutung der Zeichen

```
+...ein Blank wird in + umgewandelt
?...Startzeichen für die Liste der
Werte
&...Trennt die Werte der einzelnen
Variablen
%...Sonderzeichen beginnen mit diesem
Zeichen, dann folgt der HEX-Code
    für das Sonderzeichen wie z.B.
        %0A entspricht dem Zeichen
CARRIAGE RETURN (\n)
        %0D entspricht dem Zeichen
LINEFEED (\r)
```

Der Teil „elementname" entspricht dem Namen einer Komponente der Form (z.B. der Name einer Textbox), wert ent-

spricht dem Wert, der für diese Komponente festgelegt wurde (z.B. in eine Textbox „NAME" wurde der Name „Franz Huber" eingegeben, dann steht in dieser Liste NAME=Franz+Huber).

Beispiel: Browser-Fähigkeiten

```
<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
<html>
<head>
<title>Browserfähigkeiten</title>
</head>
<body>
  <% Dim Browser %>
    Browser:
  <%
=Request.Servervariables("http_user_agent")
%>
  <BR>
  <% Set Browser =
Server.CreateObject("MSWC.Browsertype") %>
  <BR>
    <TABLE>
    <TR><TD>Browsertyp: </TD><TD><%
=Browser.Browser %></TD></TR>
    <TR><TD>Version: </TD><TD><%
=Browser.version %></TD></TR>
    <TR><TD>Cookies: </TD><TD><%
=Browser.cookies %></TD></TR>
    <TR><TD>Frames: </TD><TD><%
=Browser.frames %></TD></TR>
    <TR><TD>JavaScript: </TD><TD><%
=Browser.javascript %></TD></TR>
    <TR><TD>VBScript: </TD><TD><%
=Browser.vbscript %></TD></TR>
    <TR><TD>Plattform: </TD><TD><%
=Browser.platform %></TD></TR>
  </TABLE>
</body>
</html>
```

Erklärung: Die in obiger Tabelle angezeigten Werte stammen aus der Datei **BROWSCAP.INI**, die am Server unter **%Systemroot%\System32\inetsrv** zu finden ist. Wenn Sie diese Datei mit einem Texteditor bearbeiten, können Sie auch neuere Browser anzeigen bzw. eigenwillige Ausgaben erzwingen.

Wesentlich ergänzte Versionen der Datei **BROWSCAP.INI** können zum Beispiel von http://www.cyscape.com/browscap/ heruntergeladen werden. Dort sind auch Einträge für neue Browser wie Opera, Netscape Navigator 6.0 und Internet Explorer 5.x vorhanden.

Beispiel:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;       Opera       ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;

[Opera 3.0]
browser=Opera
Version=3.0
majorver=#3
minorver=#0
frames=True
tables=True
cookies=True
backgroundsounds=False
vbscript=False
javascript=True
javaapplets=False
activexcontrols=False
AK=False
SK=False
AOL=False
beta=False
Win16=False
Crawler=False
CDF=False

[Mozilla/3.0 (compatible; Opera/3.0;
windows 3.1)*]
parent=Opera 3.0
platform=Win31

[Mozilla/3.0 (compatible; Opera/3.0;
Windows 95/NT)*]
parent=Opera 3.0
platform=Win32

[Mozilla/3.0 (compatible; Opera/3.0b8;
Windows 95/NT)*]
parent=Opera 3.0
platform=Win32
beta=True
```

Datenbankzugriff mit Active Server Pages mit Hilfe der ActiveX Data Objects

Voraussetzungen

- Windows NT 4.0 Server mit Service Pack 3
- Auf diesem Server ist das NT Option Pack 4 installiert (enthält Internet Information Server 4.0)
- Nach der Installation des Option Packs könnte NT SP 4 auch nachinstalliert werden, aber keinesfalls vorher (sonst Absturz des Internet Information Servers)
- Auf dem IIS ist eine Website installiert
- Die Datei **adovbs.inc** muss aus dem Verzeichnis **C:\Programme\Gemeinsame Dateien\System\ado** in ein bekanntes Verzeichnis innerhalb der Website (z.B. **C:\InetPub\WWWRoot**) kopiert werden. (Diese Datei enthält die ActiveX Data Objects)

Was sind die ActiveX Data Objects?

Diese Objekte stellen eine Sammlung von Klassen – vergleichbar den Data Access Objects (DAO) – dar, die den Zugriff auf ODBC-Datenbanken direkt von VBScript aus ermöglichen.

Folgende sieben **Objekte** (Unterklassen) gehören zur Hauptklasse ADODB (ActiveX-Data-Object-Database):

- `Connection`: Stellt die Verbindung zu einem SQL-Server her
- `RecordSet`: Stellt die Datenschnittstelle zu den Tabellen der Datenbank her
- `Field`: Erlaubt den Zugriff auf ein einzelnes Feld
- `Command`: Sendet einzelne Kommandos an den SQL-Server oder startet gespeicherte

Prozeduren

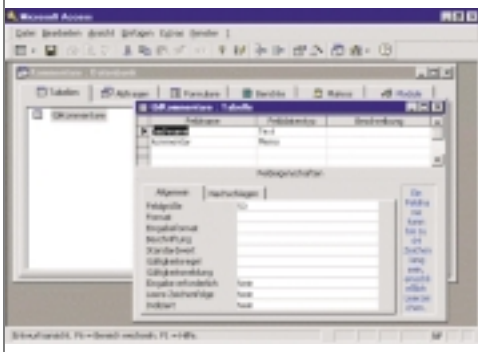
- `Property`: Erlaubt den Zugriff auf Eigenschaften der SQL-Datenbank
- `Error`: Objekt zur Behandlung von Fehlermeldungen

Vorgangsweise

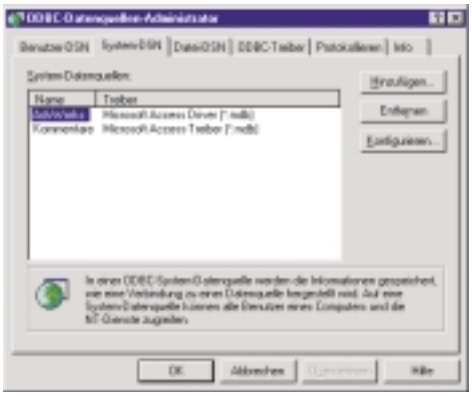
1. Datenbank erstellen
2. ODBC-Eintrag erstellen
3. ASP-Datei programmieren

Zu 1. Datenbank erstellen
Erstellen Sie Ihre Datenbank – wie gewohnt – mit dem Datenbank-Management-System, etwa Access. Speichern Sie die Datenbank in einem Verzeichnis Ihrer Website auf dem Server.

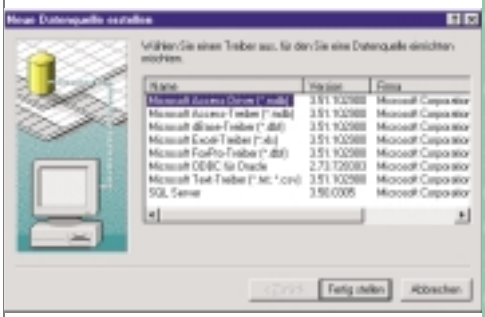
Zu 2. ODBC-Schnittstelle einrichten



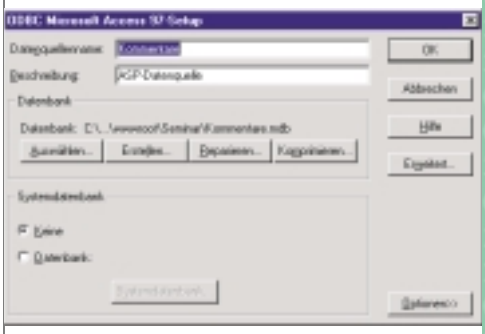
Starten Sie die Systemsteuerung und doppelklicken Sie auf das Icon "***ODBC***". Wählen Sie die Karteikarte "***System-DSN***" und klicken Sie auf "***Hinzufügen***".



Hier wählen Sie zunächst den ODBC-Treiber aus, mit dem Ihre Datenbank angesprochen werden soll (in unserem Fall Access):



Dann vergeben Sie für Ihre Datenbank einen *Data Source Name* (DSN), unter dem die Datenbank angesprochen wird. Sie können auch eine erklärende Beschreibung für Ihre Datenquelle vergeben.



Vergessen Sie nicht, mit der Befehlsschaltfläche „***Auswählen***" den lokalen Pfad sowie den Dateinamen für Ihre Datenbank anzugeben!

zu 3. ASP-Datei programmieren

Beispiel 3: Abfrage einer bestehenden Datenbank

```
<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
<HTML>
  <HEAD>
    <TITLE>Datenbankabfrage</TITLE>
  </HEAD>
  <BODY>
    <% Dim sc, sqlq, rs %>
    <!– #include virtual="/adovbs.inc" –>
    <% Set
sc=Server.CreateObject("ADODB.Connection")
%>
    <% sc.open "Kommentare" %>
    <% sqlq="SELECT tblKommentare.nachname,
tblKommentare.kommentar
       FROM tblKommentare" %>
    <% set rs = sc.execute(sqlq,,adCmdText)
%>
    <% Do While Not rs.EOF %>
    <%    =rs("nachname") & " " %>
      <%    =rs("kommentar") %>
     <% rs.MoveNext %> <BR>
    <% Loop %>
  </BODY>
</HTML>
```

Vorgangsweise

- Server-`Connection`-Objekt erzeugen
- SQL-String festlegen
- `RecordSet`-Objekt erzeugen (dazu nötig: Abfrage-SQL-String)
- Datenbankeinträge anzeigen (Do While-Schleife – Recordset-Inhalt in HTML-Seite anzeigen)

```
Auszug aus der Datei ADOVBS.INC
Die Datei adovbs.inc enthält eine Reihe von
Konstantenvereinbarungen, die ASP-Skripte
weitaus besser lesbar machen.
<%
'--------------------------------------
' Microsoft ADO
'
' (c) 1996 Microsoft Corporation.  All
Rights Reserved.
'
'
'
' ADO constants include file for VBScript
'
'--------------------------------------


'-- CursorTypeEnum Values --
Const adOpenForwardOnly = 0
Const adOpenKeyset = 1
Const adOpenDynamic = 2
Const adOpenStatic = 3

' ..... und so weiter ......
%>
```

Mit der Anweisung

```
<!– #include virtual="adovbs.inc" –>
```

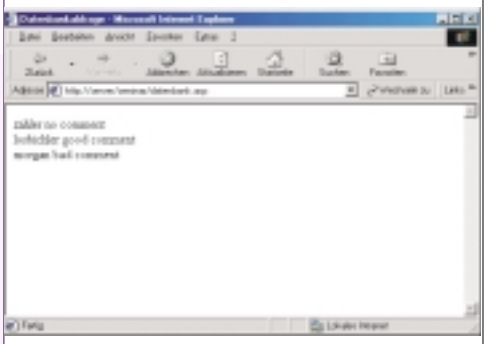
kann die obige Datei so in eine ASP-Datei integriert werden, wie wenn der Code selbst an dieser Stelle eingefügt worden wäre.

Wie sieht obiges Beispiel vom Client aus aus?

Sichtbarer Quellcode am Client:

```
<HTML>
  <HEAD>
    <TITLE>Datenbankabfrage</TITLE>
  </HEAD>
  <BODY>
    zahler no comment <BR>
    losbichler good comment <BR>
    morgan bad comment <BR>
  </BODY>
</HTML>
```

Also: Am Client wird nur eine normale

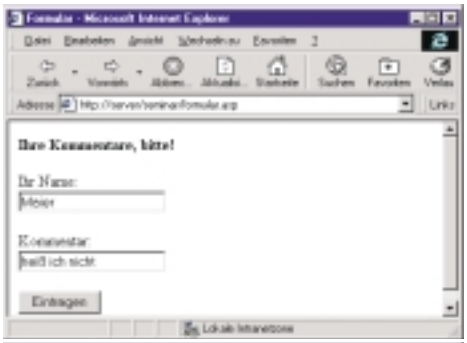


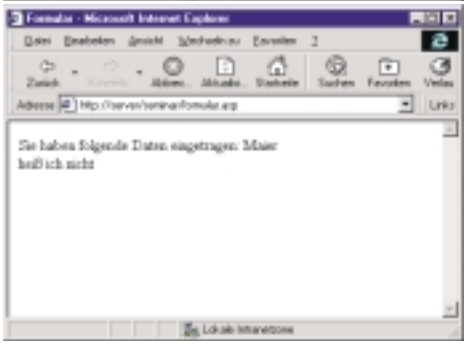
HTML-Seite angezeigt; die serverseitigen Befehle scheinen nicht!

Beispiel 4: Eintrag in eine Datenbank über ein client-seitiges Formular

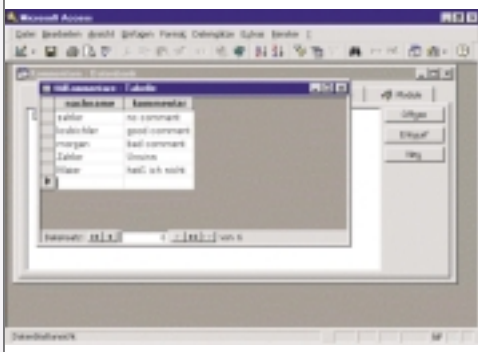
```
<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
<HTML>
  <HEAD>
    <TITLE>Formular</TITLE>
  </HEAD>
  <BODY>
  <% Dim sc, sqlq, rs %>
  <!– #include virtual="adovbs.inc" –>
   <% If Request.Form("eingName")=""
        Or Request.Form("eingKommentar")=""
Then %>
  <H4>Ihre Kommentare, bitte!</H4>
  <FORM METHOD="POST" ACTION="formular.asp">
    <P>Ihr Name:<BR>
    <INPUT TYPE="text" NAME="eingName">
    <P>Kommentar:<BR>
    <INPUT TYPE="text" NAME="eingKommentar">
    <P>
    <INPUT TYPE="submit" VALUE="Eintragen">
   </FORM>
   <% Else %>
      Sie haben folgende Daten eingetragen:
     <% =Request.Form("eingName") %>
     <BR>
     <% =Request.Form("eingKommentar") %>
    <% Set
sc=Server.CreateObject("ADODB.Connection")
%>
    <% Set
rs=Server.CreateObject("ADODB.RecordSet") %>
    <% sc.Open "Kommentare" %>
    <% rs.Open "SELECT * FROM
tblKommentare", _
sc,adOpenDynamic,adLockPessimistic,adCMDText
%>
    <% rs.AddNew %>
    <% rs("nachname") =
Request.Form("eingName") %>
    <% rs("kommentar") =
Request.Form("eingKommentar") %>
    <% rs.Update %>
    <% rs.Close %>
    <% sc.Close %>
   <% End If %>
   </BODY>
</HTML>
```

So sieht diese Datei von der Client-Seite aus:





Der Beweis: Er hat's wirklich eingetragen!!



Dokumentation zu ODBC

Die einführende Dokumentation zum ODBC-System findet man (z.B. für Windows NT) im Ordner **`C:\windows\system32`**. Es sind die Hilfedateien:

`odbc.hlp`	ODBC 3.0 Programmer's Reference
`odbcinst.hlp`	Diese Hilfedatei gibt Aufschluss über die Registerkarten im ODBC-Treiber
`odbcjet.hlp`	ODBC Microsoft Desktop Database Drivers Help
`odbcjtnw.hlp`	ODBC Microsoft Desktop Database Drivers What's New Help

Die aktuelle Versionsnummer des ODBC-Systems ist 4.0.