VB5DB Object Reference

Database object from the VB5-CGI Objects collection. Copyright © 1997, 1998 EazyWare.

Reference

⊞ **Search**
⊞ **AddNew**
⊞ **General**

VB5DB Example

VB5CGI Object
VB5HTML Object
VB5-CGI Objects Overview

VB5DB Object Reference

Database object from the VB5-CGI Objects collection.

Reference

⊟ **Search**

- AddSearchField
- AddSearchKeyNumeric
- AddSearchKeyString
- CreateSearchTable
- GetRecordsetRows
- GetSearchQuery
- GetSearchTable
- IsSearchRestricted
- OpenSearch
- **AddNew**
- **General**

VB5DB Example

VB5CGI Object
VB5HTML Object
VB5-CGI Objects Overview

VB5DB Object Reference

Database object from the VB5-CGI Objects collection. Copyright © 1997, 1998 EazyWare.

Reference

- **Search**
- **AddNew**
- AddNewField
- OpenAddNew
- Update
- **General**

VB5DB Example

VB5CGI Object
VB5HTML Object
VB5-CGI Objects Overview

VB5DB Object Reference

Database object from the VB5-CGI Objects collection. Copyright © 1997, 1998 EazyWare.

Reference

- **Search**
- **AddNew**
- **General**
- CloseAll
- DatabasePath

VB5DB Example

VB5CGI Object
VB5HTML Object
VB5-CGI Objects Overview

AddSearchField

Adds an existing database table field name to the SQL search query string, which will be shown in the HTML search table.

Syntax

Sub AddSearchField(ByVal FieldName As String)

Parameters

FieldName : Existing database table field name.

Example

```
Dim search  As String  'SQL search query
Dim totrecs As Long    'Number of total found records
Dim maxrecs As Long    'Number of maxium shown table rows
(...)
DB.AddSearchKeyString "KeyName", "FieldName", True, True
DB.AddSearchKeyNumeric "KeyAge", "FieldAge", ">=", True
DB.AddSearchField "FieldRemarks"
search = DB.GetSearchQuery("TableName", "FieldName")
totrecs = DB.OpenSearch("C:\WebDB\Database.mdb", search)
If totrecs > 0 Then
    maxrecs = DB.CreateSearchTable(20)
    HTML.SubmitPage "Test Search", DB.GetSearchTable()
Else
    HTML.SubmitPage "Test Search", "No records found!"
End If
DB.CloseAll
```

This example searches in a database and returns the matching records in a HTML table:
Two form input fields with the name "KeyName" and "KeyAge" related to the database fields "FieldName" and "FieldAge" and one database field **"FieldRemarks"** are added for the following GetSearchQuery method, which returns the SQL search query.
The OpenSearch function opens the database "C:\WebDB\Database.mdb" with the previous created search query and returns the number of found records. If the search query was successful and found at least one matching record the resulting search table will be created and submitted. Finally the database has to be closed.

AddSearchKeyNumeric

Adds a numeric query string parameter key value to the SQL search query string.

Syntax

Sub AddSearchKeyNumeric(ByVal KeyName As String, [ByVal FieldName As String], [ByVal CompareOperator As String = "="], [ByVal ShowInTable As Boolean = True])

Parameters

KeyName : Name of the parameter key name from the query string (KeyName=KeyValue), case insensitive.
FieldName : Name of the existing database table field name. If omitted, the parameter KeyName will be used as field name.
CompareOperator : Defines how the query compares the KeyValue with the FieldName value (=, <=, >=, <, >, <>), default="=".
ShowInTable : If True, this database field value will be shown in the HTML table, default=True.

Example

```
Dim search  As String  'SQL search query
Dim totrecs As Long    'Number of total found records
Dim maxrecs As Long    'Number of maxium shown table rows
(...)
DB.AddSearchKeyString "KeyName", "FieldName", True, True
DB.AddSearchKeyNumeric "KeyAge", "FieldAge", ">=", True
DB.AddSearchField "FieldRemarks"
search = DB.GetSearchQuery("TableName", "FieldName")
totrecs = DB.OpenSearch("C:\WebDB\Database.mdb", search)
If totrecs > 0 Then
    maxrecs = DB.CreateSearchTable(20)
    HTML.SubmitPage "Test Search", DB.GetSearchTable()
Else
    HTML.SubmitPage "Test Search", "No records found!"
End If
DB.CloseAll
```

This example searches in a database and returns the matching records in a HTML table:
Two form input fields with the name "KeyName" and **"KeyAge"** related to the database fields "FieldName" and **"FieldAge"** and one database field "FieldRemarks" are added for the following GetSearchQuery method, which returns the SQL search query.
The OpenSearch function opens the database "C:\WebDB\Database.mdb" with the previous created search query and returns the number of found records. If the search query was successful and found at least one matching record the resulting search table will be created and submitted. Finally the database has to be closed.

AddSearchKeyString

Adds a text query string parameter key value to the SQL search query string.

Syntax

Sub AddSearchKeyString(ByVal KeyName As String, [ByVal FieldName As String], [ByVal UseLike As Boolean = True], [ByVal ShowInTable As Boolean = True])

Parameters

KeyName : Name of the parameter key name from the query string (KeyName=KeyValue), case insensitive.
FieldName : Name of the existing database table field name. If omitted, the parameter KeyName will be used as field name.
UseLike : If True, the SQL query will use the Like operator, which allows to search with wildcards (see the VB5 helpfile). If False, the corresponding KeyName value has to match exactly the field value. Default=True.
ShowInTable : If True, this database field value will be shown in the HTML table, default=True.

Example

```
Dim search  As String  'SQL search query
Dim totrecs As Long    'Number of total found records
Dim maxrecs As Long    'Number of maxium shown table rows
(...)
DB.AddSearchKeyString "KeyName", "FieldName", True, True
DB.AddSearchKeyNumeric "KeyAge", "FieldAge", ">=", True
DB.AddSearchField "FieldRemarks"
search = DB.GetSearchQuery("TableName", "FieldName")
totrecs = DB.OpenSearch("C:\WebDB\Database.mdb", search)
If totrecs > 0 Then
    maxrecs = DB.CreateSearchTable(20)
    HTML.SubmitPage "Test Search", DB.GetSearchTable()
Else
    HTML.SubmitPage "Test Search", "No records found!"
End If
DB.CloseAll
```

This example searches in a database and returns the matching records in a HTML table:
Two form input fields with the name **"KeyName"** and "KeyAge" related to the database fields **"FieldName"** and "FieldAge" and one database field "FieldRemarks" are added for the following GetSearchQuery method, which returns the SQL search query.
The OpenSearch function opens the database "C:\WebDB\Database.mdb" with the previous created search query and returns the number of found records. If the search query was successful and found at least one matching record the resulting search table will be created and submitted. Finally the database has to be closed.

CreateSearchTable

Creates the HTML search table and returns the positive number of returned rows or the negative number of containing rows.

Syntax

Function CreateSearchTable([ByVal FirstRecord As Long = 1], [ByVal LastRecord As Long], [ByVal ShowRecordNumber As Boolean = True], [ByVal ShowTableHeader As Boolean = True], [ByVal ShowBorder As Boolean = True], [ByVal HTMLTable As String], [ByVal HTMLTableHeader As String], [ByVal HTMLTableData As String]) As Long

Parameters

FirstRecord : First record to show in table. Default=1.
LastRecord : Last record to show in table. Default=containing records in recordset.
ShowRecordNumber : If True, the first table column will be the row number, starting from 1. If the table header is visible, the heading will be "Nr.". Default=True.
ShowTableHeader : If True, the table header will be visible. The heading for each column is the database field name. Default=True.
ShowBorder : If True a standard border will be used for the table. Default=True.
HTMLTable : Additional table HTML tags (e.g. "BGCOLOR=RED"). Default=none.
HTMLTableHeader : Additional table header HTML tags (e.g. "BGCOLOR=RED"). Default=none.
HTMLTableData : Additional table data HTML tags (e.g. "BGCOLOR=RED"). Default=none.

Return Value

If the return value is positive: Number of visible table rows (FirstRecord to LastRecord, if they were specified).
The absolute value of a negative return value describes the number of found records, which are outside of the specified range.

Example

```
Dim search  As String  'SQL search query
Dim totrecs As Long    'Number of total found records
Dim maxrecs As Long    'Number of maxium shown table rows
(...)
DB.AddSearchKeyString "KeyName", "FieldName", True, True
DB.AddSearchKeyNumeric "KeyAge", "FieldAge", ">=", True
DB.AddSearchField "FieldRemarks"
search = DB.GetSearchQuery("TableName", "FieldName")
totrecs = DB.OpenSearch("C:\WebDB\Database.mdb", search)
If totrecs > 0 Then
    maxrecs = DB.CreateSearchTable(20)
    HTML.SubmitPage "Test Search", DB.GetSearchTable()
Else
    HTML.SubmitPage "Test Search", "No records found!"
End If
DB.CloseAll
```

This example searches in a database and returns the matching records in a HTML table:

Two form input fields with the name "KeyName" and "KeyAge" related to the database fields "FieldName" and "FieldAge" and one database field "FieldRemarks" are added for the following GetSearchQuery method, which returns the SQL search query.
The OpenSearch function opens the database "C:\WebDB\Database.mdb" with the previous created search query and returns the number of found records. If the search query was successful and found at least one matching record **the resulting search table with max. 20 rows will be created** and submitted. Finally the database has to be closed.

GetRecordsetRows

Returns the actual recordset as a two-dimensional array -> varRecords(intField, intRecord).

Syntax

Public Function GetRecordsetRows([ByVal FirstRecord As Long], [ByVal LastRecord As Long])

Parameters

FirstRecord : First needed record in returned array. Default=0.
LastRecord : Last needed record in returned array. Default=max. contained record in recordset.
Note: The lower bound of the returned array (both dimensions: field and record) always starts with 0. See also the GetRows method from the VB documentation.

Return Value

Two-dimensional array. The first subscript identifies the field and the second the row or record number: varRecords(intField, intRecord)).

Example

```
Dim search  As String  'SQL search query
Dim totrecs As Long    'Number of total found records
Dim maxrecs As Long    'Number of maxium shown table rows

Dim records As Variant 'Contains the two-dimensional recordset array
Dim tempstr As String  'Temporary string variable
Dim row     As Integer 'Row or record counter
Dim col     As Integer 'Colon or field counter
(...)
DB.AddSearchKeyString "KeyName", "FieldName", True, True
DB.AddSearchKeyNumeric "KeyAge", "FieldAge", ">=", True
DB.AddSearchField "FieldRemarks"
search = DB.GetSearchQuery("TableName", "FieldName")
totrecs = DB.OpenSearch("C:\WebDB\Database.mdb", search)
If totrecs > 0 Then
    records = DB.GetRecordsetRows(5, 15)
    For row = 0 To UBound(records, 2)
        For col = 0 To UBound(records, 1)
            tempstr = tempstr & rows(col, row) & ", "
        Next col
        tempstr = tempstr & HTML.GetBreak
    Next row
    HTML.SubmitPage "Test Search", tempstr
Else
    HTML.SubmitPage "Test Search", "No records found!"
End If
DB.CloseAll
```

This example searches in a database and returns eleven rows (row 5 to 15) from the created recordset with coma separated field values.

Two form input fields with the name "KeyName" and "KeyAge" related to the database fields "FieldName" and "FieldAge" and one database field "FieldRemarks" are added for the following GetSearchQuery method (database table name="TableName", sorted by the field "FieldName"), which returns the SQL search query.
The OpenSearch function opens the database "C:\WebDB\Database.mdb" with the previous created search query and returns the number of found records. If the search query was successful and found at least one matching record, **record 5 to 15 will be retrieved (if they exist) with the "GetRecordsetRows" method**.
After that, the array with the containing field gets HTML formatted and submitted. Finally the database has to be closed.

GetSearchQuery

Returns the database SQL search query string.

Syntax

Function GetSearchQuery(ByVal TableName As String, [ByVal OrderFieldName As String]) As String

Parameters

TableName : Name of the database table.
OrderFieldName : Name of the database table field, by which the resulting HTML table will be sorted. Default=no specific order will be used.

Return Value

Text of the SQL database query.

Example

```
Dim search  As String  'SQL search query
Dim totrecs As Long    'Number of total found records
Dim maxrecs As Long    'Number of maxium shown table rows
(...)
DB.AddSearchKeyString "KeyName", "FieldName", True, True
DB.AddSearchKeyNumeric "KeyAge", "FieldAge", ">=", True
DB.AddSearchField "FieldRemarks"
search = DB.GetSearchQuery("TableName", "FieldName")
totrecs = DB.OpenSearch("C:\WebDB\Database.mdb", search)
If totrecs > 0 Then
    maxrecs = DB.CreateSearchTable(10, 20)
    HTML.SubmitPage "Test Search", DB.GetSearchTable()
Else
    HTML.SubmitPage "Test Search", "No records found!"
End If
DB.CloseAll
```

This example searches in a database and returns a range of matching records (row 10 to 20) in a HTML table:
Two form input fields with the name "KeyName" and "KeyAge" related to the database fields "FieldName" and "FieldAge" and one database field "FieldRemarks" are added for the following **GetSearchQuery method (database table name="TableName", sorted by the field "FieldName"), which returns the SQL search query**.
The OpenSearch function opens the database "C:\WebDB\Database.mdb" with the previous created search query and returns the number of found records. If the search query was successful and found at least one matching record the resulting search table will be created and submitted. Finally the database has to be closed.

GetSearchTable

Returns the created HTML table string from the last search query.

Syntax

Function GetSearchTable() As String

Return Value

HTML table text.

Example

```
Dim search  As String  'SQL search query
Dim totrecs As Long    'Number of total found records
Dim maxrecs As Long    'Number of maxium shown table rows
(...)
DB.AddSearchKeyString "KeyName", "FieldName", True, True
DB.AddSearchKeyNumeric "KeyAge", "FieldAge", ">=", True
DB.AddSearchField "FieldRemarks"
search = DB.GetSearchQuery("TableName", "FieldName")
totrecs = DB.OpenSearch("C:\WebDB\Database.mdb", search) Then
If totrecs > 0 Then
    maxrecs = DB.CreateSearchTable(20)
    HTML.SubmitPage "Test Search", DB.GetSearchTable()
Else
    HTML.SubmitPage "Test Search", "No records found!"
End If
DB.CloseAll
```

This example searches in a database and returns the matching records in a HTML table:
Two form input fields with the name "KeyName" and "KeyAge" related to the database fields "FieldName" and "FieldAge" and one database field "FieldRemarks" are added for the following GetSearchQuery method, which returns the SQL search query.
The OpenSearch function opens the database "C:\WebDB\Database.mdb" with the previous created search query and returns the number of found records. If the search query was successful and found at least one matching record the resulting **search table will be created, returned and submitted**.
Finally the database has to be closed.

IsSearchRestricted

Returns True, if the search query is restricted (the WHERE-clause contains data).

Syntax

Function IsSearchRestricted() As Boolean

Return Value

True, if the created SQL query contains a restriction. False, if no restriction was used, therefore all database records will be returned.

Example

```
Dim search  As String  'SQL search query
Dim totrecs As Long    'Number of total found records
Dim maxrecs As Long    'Number of maxium shown table rows
(...)
DB.AddSearchKeyString "KeyName", "FieldName", True, True
(...)
If DB.IsSearchRestricted() Then
    search = DB.GetSearchQuery("TableName", "FieldName")
    (...)
Else
    HTML.SubmitPage "Test Search", "No query was specified"
End If
```

This example examines, if at least one query form field was used to restrict the database query. Instead of returning all database records, the HTML page with the message "No query was specified" will be shown.

OpenSearch

Opens a database file and creates a search query recordset. Returns the number of found records.

Syntax

Function OpenSearch(ByVal DatabaseFileName As String, ByVal SearchQuery As String, [ByVal Username As String = "admin"], [ByVal Password As String]) As Long

Parameters

DatabaseFileName : Name of the database file (.mdb). If the name starts with a "\", the relative path will be taken from the property DatabasePath.
SearchQuery : The SQL query text. Usually, this query comes from the method GetSearch .
Username : Username, if the database is secure, default="admin".
Password : Password, if the database is secure, default="".

Return Value

Number of found records.

Example

```
Dim search  As String  'SQL search query
Dim totrecs As Long    'Number of total found records
Dim maxrecs As Long    'Number of maxium shown table rows
(...)
DB.AddSearchKeyString "KeyName", "FieldName", True, True
DB.AddSearchKeyNumeric "KeyAge", "FieldAge", ">=", True
DB.AddSearchField "FieldRemarks"
search = DB.GetSearchQuery("TableName", "FieldName")
totrecs = DB.OpenSearch("C:\WebDB\Database.mdb", search)
If totrecs > 0 Then
    maxrecs = DB.CreateSearchTable(20)
    HTML.SubmitPage "Test Search", DB.GetSearchTable()
Else
    HTML.SubmitPage "Test Search", "No records found!"
End If
DB.CloseAll
```

This example searches in a database and returns the matching records in a HTML table:
Two form input fields with the name "KeyName" and "KeyAge" related to the database fields "FieldName" and "FieldAge" and one database field "FieldRemarks" are added for the following GetSearchQuery method, which returns the SQL search query.
The OpenSearch function opens the database "C:\WebDB\Database.mdb" with the previous created search query and returns the number of found records. If the search query was successful and found at least one matching record the resulting search table will be created and submitted. Finally the database has to be closed.

AddNewField

Adds a value to a database table field (adds a new record, when first time used).

Syntax

Sub AddNewField(ByVal FieldName As String, ByVal FieldValue As Variant)

Parameters

FieldName : Existing database table field name.
FieldValue : Type compatible field value.

Example

```
If DB.OpenAddNew("\MyData.mdb", "MyTable") Then
    DB.AddNewField "MyFieldName", "My Name"
    DB.CloseAll()
End If
```

This example opens the database table "MyTable" from the database "\MyData.mdb". If successful, **it adds a new record to the table "MyTable" and enters the string value "My Name" to the field "MyFieldName"**.
After adding new records, it is important to update the changed table. In this example, we use the CloseAll method, which updates the table "MyTable" and closes the database.

OpenAddNew

Opens a database to add a new record. Returns True, if successful.

Syntax

Function OpenAddNew(ByVal DatabaseFileName As String, ByVal TableName As String, [ByVal Username As String = "admin"], [ByVal Password As String]) As Boolean

Parameters

DatabaseFileName : Name of the database file (.mdb). If the name starts with a "\", the relative path will be taken from the property DatabasePath.
TableName : Name of the database table, to which a record will be added.
Username :   Username, if the database is secure, default="admin".
Password : Password, if the database is secure, default="".

Return Value

True, if the database and table could be opened successfully.

Example

```
If DB.OpenAddNew("\MyData.mdb", "MyTable") Then
    DB.AddNewField "MyFieldName", "My Name"
    DB.CloseAll()
End If
```

This example **opens the database table "MyTable" from the database "\MyData.mdb"**. If successful, it adds a new record to the table "MyTable" and enters the string value "My Name" to the field "MyFieldName".
After adding new records, it is important to update the changed table. In this example, we use the CloseAll method, which updates the table "MyTable" and closes the database.

Update

Updates a changed database record.
This method is only needed, when within a script, multiple times records get written to the database, without closing them before.

Syntax

Sub Update([ByVal RefreshCache = True])

Parameters

RefreshCache : If True, the changed record will be immediately written to the disk, default=True.

Example

```
If DB.OpenAddNew("\MyData.mdb", "MyTable") Then
    DB.AddNewField "MyFieldName", "John"
    DB.Update()
    DB.AddNewField "MyFieldName", "Mark"
    DB.CloseAll()
End If
```

This example opens the database table "MyTable" from the database "\MyData.mdb". If successful, it adds a new record to the table "MyTable" and enters the string value "John" to the field "MyFieldName".
After adding new records, it is important to update the changed table, if new records will be added. Finally, we use the CloseAll method, which updates the table "MyTable" and closes the database.

CloseAll

Updates and closes all related database objects.

Syntax

Sub CloseAll([ByVal RefreshCache = True])

Parameters

RefreshCache : If True, the changed record will be immediately written to the disk, default=True.

Example

```
If DB.OpenAddNew("\MyData.mdb", "MyTable") Then
    DB.AddNewField "MyFieldName", "My Name"
    DB.CloseAll()
End If
```

After adding new records, it is important to update the changed table. In this example, we use the CloseAll method, which updates the table "MyTable" and closes the database and also all related database objects.

DatabasePath

Returns/Sets the default database path and will be used, if a path starts with "\".

Syntax

Property DatabasePath As String

Example

```
DB.DatabasePath = "C:\Web\MyDatabase"
```

This example sets the default database path to "C:\Web\MyDatabase".

VB5DB Example

The following Visual Basic source code (demoDB.bas) shows a CGI script, which mainly uses the VB5DB object. This example was tested with the Microsoft Personal Web Server version 1.0a and 2.0 and the IIS3.
If no query string was submitted, (e.g. http://localhost/scripts/demoDB.exe), an error page will be shown.
Otherwise, all required form fields and additional database fields are added to build the SQL query. If the user entered at least one query field, this search query gets created and used to open the database, which returns the number of found records. After creating the HTML search table, a page shows the corresponding information with a hit counter.

```
Private CGI     As New VB5CGI.clsCGI    'Instance the VB5CGI Object (needs
VB5CGI.DLL)
Private HTML    As New VB5HTML.clsHTML  'Instance the VB5HTML Object
(needs VB5HTML.DLL)
Private DB      As New VB5DB.clsDB      'Instance the VB5DB Object (needs
VB5DB.DLL)

Private search  As String               'Search query (SQL string)
Private totrecs As Long                 'Number of total matching records
found in the database
Private maxrecs As Long                 'Number of maximum matching
records shown in the table

Sub Main()
    With HTML
        'Add an additional text to an eventual error message
        .ErrorSubText = "This could be your error text - " & _
                        .GetTextLink("EasyWare",
"mailto:tools@eazyware.com")
        'This is the default database file path, please change it
accordingly
        DB.DatabasePath = "C:\Program Files\VB5-CGI Objects\Samples"
        If .InitQueryString() Then                          'Did we get a
query string?
            With DB
                'Build the search query, and if the checkboxes on the form
are checked,
                'add them in the following order to the resulting table
                .AddSearchKeyString "State", , ,
HTML.HasKeyValue("ShowState")
                .AddSearchKeyString "City", , ,
HTML.HasKeyValue("ShowCity")
                .AddSearchKeyString "ZipCode", , ,
HTML.HasKeyValue("ShowZipCode")
                .AddSearchKeyString "HotelName", , ,
HTML.HasKeyValue("ShowHotelName")
                .AddSearchKeyNumeric "Cost", ,
HTML.GetKeyString("CostOperator"), _

HTML.HasKeyValue("ShowCost")
                If .IsSearchRestricted() Then
                   'Add some additional database fields in the following
order to the table
                    .AddSearchField "Address"
                    .AddSearchField "Tel"
                    .AddSearchField "Fax"
                    .AddSearchField "Email"
                    .AddSearchField "Remarks"
                    'Create the SQL search query from the database table
"tblHotels" by order
                    search = .GetSearchQuery("tblHotels",
HTML.GetKeyString("Order"))
                    'Open the database file with the SQL search query
                    totrecs = .OpenSearch("\hotel.mdb", search)
                    'Create the search result table and specify max.
records to return
                    maxrecs
```

```
= .CreateSearchTable(HTML.GetKeyInteger("FirstRow"), _
                    HTML.GetKeyInteger("LastRow"), , , False,
"bgcolor=#FFCC99", "bgcolor=#99FFCC")
                    .CloseAll                               'Close all
database objects
                Else
                    'Tell the user, that no query field was specified
                    HTML.SubmitPage "No query was specified", _
                                    "Please, enter at least one query
field!", True
                    End
                End If
            End With
            'Create the HTML page with the result table
            .PageBegin "VB5-CGI Objects demonstration [" &
CGI.GetScriptName(True) & "]", "FFFFCC"
            If maxrecs > 0 Then                             'Did the
database query return any records?
                If totrecs > maxrecs Then
                    .BodyTextB "The following " & maxrecs & " of total " &
totrecs & " record(s) matched your query:"
                Else
                    .BodyTextB "The following " & maxrecs & " record(s)
matched your query:"
                End If
                .BodyHTML DB.GetSearchTable()               'Return the
HTML table
                .BodyTextBI "This table has been created " &
CGI.GetHitCounterInc(App.EXEName) & " times!"
            ElseIf maxrecs < 0 Then
                .BodyTextB Abs(maxrecs) & " records outside the specified
range found, please try again"
            Else
                .BodyTextB "No matching record was found, please try
again"
            End If
            .PageEnd                                        'End and
submit the page
        Else
            HTML.ErrorPage "No query string received!"      'No query
string received
        End If
    End With
End Sub
```

HTML Example Source

HTML Example Source

The following HTML text was used (demoDB.htm) to invoke the script:

```
<HTML>
<HEAD>
  <TITLE>VB5-CGI Objects demo: VB5DB object</TITLE>
</HEAD>

<BODY BGCOLOR="#FFFFFF">
<H2>VB5-CGI Objects demo: VB5DB object</H2>
<B>CGI script: demoDB.exe</B>
<HR>
<P><B>To find a Hotel, please enter a search query below:</B></P>
<FORM METHOD="POST" ACTION="http://localhost/Scripts/demoDB.exe">
  <TABLE BORDER="0" HEIGHT="250" CELLPADDING="2" WIDTH="621">
    <TR>
      <TD WIDTH="162" HEIGHT="25"></TD>
      <TD WIDTH="282" HEIGHT="25"></TD>
      <TD ALIGN="center" HEIGHT="25" WIDTH="73">Show</TD>
      <TD ALIGN="center" HEIGHT="25" NOWRAP WIDTH="80">Order by</TD>
    </TR>
    <TR>
      <TD WIDTH="162" HEIGHT="25">State</TD>
      <TD WIDTH="282" HEIGHT="25"><SELECT NAME="state" SIZE="1">
        <OPTION> </OPTION>
        <OPTION VALUE="AL">AL</OPTION>
        <OPTION VALUE="AK">AK</OPTION>
        <OPTION VALUE="AZ">AZ</OPTION>
        <OPTION VALUE="AR">AR</OPTION>
        <OPTION VALUE="CA">CA</OPTION>
        <OPTION VALUE="CO">CO</OPTION>
        <OPTION VALUE="CT">CT</OPTION>
        <OPTION VALUE="DE">DE</OPTION>
        <OPTION VALUE="DC">DC</OPTION>
        <OPTION VALUE="FL">FL</OPTION>
        <OPTION VALUE="GA">GA</OPTION>
        <OPTION VALUE="HI">HI</OPTION>
        <OPTION VALUE="ID">ID</OPTION>
        <OPTION VALUE="IL">IL</OPTION>
        <OPTION VALUE="IN">IN</OPTION>
        <OPTION VALUE="IA">IA</OPTION>
        <OPTION VALUE="KS">KS</OPTION>
        <OPTION VALUE="KY">KY</OPTION>
        <OPTION VALUE="LA">LA</OPTION>
        <OPTION VALUE="MA">MA</OPTION>
        <OPTION VALUE="ME">ME</OPTION>
        <OPTION VALUE="MD">MD</OPTION>
        <OPTION VALUE="MI">MI</OPTION>
        <OPTION VALUE="MN">MN</OPTION>
        <OPTION VALUE="MS">MS</OPTION>
        <OPTION VALUE="MO">MO</OPTION>
        <OPTION VALUE="MT">MT</OPTION>
        <OPTION VALUE="NE">NE</OPTION>
        <OPTION VALUE="NV">NV</OPTION>
        <OPTION VALUE="NH">NH</OPTION>
        <OPTION VALUE="NJ">NJ</OPTION>
        <OPTION VALUE="NM">NM</OPTION>
        <OPTION VALUE="NY">NY</OPTION>
        <OPTION VALUE="NC">NC</OPTION>
        <OPTION VALUE="ND">ND</OPTION>
        <OPTION VALUE="OH">OH</OPTION>
        <OPTION VALUE="OK">OK</OPTION>
        <OPTION VALUE="OR">OR</OPTION>
        <OPTION VALUE="PA">PA</OPTION>
        <OPTION VALUE="RI">RI</OPTION>
        <OPTION VALUE="SC">SC</OPTION>
        <OPTION VALUE="SD">SD</OPTION>
        <OPTION VALUE="TN">TN</OPTION>
        <OPTION VALUE="TX">TX</OPTION>
        <OPTION VALUE="UT">UT</OPTION>
        <OPTION VALUE="VT">VT</OPTION>
        <OPTION VALUE="VA">VA</OPTION>
        <OPTION VALUE="WA">WA</OPTION>
        <OPTION VALUE="WV">WV</OPTION>
        <OPTION VALUE="WI">WI</OPTION>
        <OPTION VALUE="WY">WY</OPTION>
      </SELECT></TD>
      <TD ALIGN="center" HEIGHT="25" WIDTH="73"><INPUT TYPE="checkbox"
NAME="ShowState"
      VALUE="ON" CHECKED></TD>
      <TD ALIGN="center" HEIGHT="25" NOWRAP WIDTH="80"><INPUT TYPE="radio"
VALUE="State"
      NAME="Order"></TD>
    </TR>
    <TR>
      <TD WIDTH="162" HEIGHT="25">City</TD>
      <TD WIDTH="282" HEIGHT="25"><INPUT TYPE="text" NAME="City"
SIZE="30"></TD>
      <TD ALIGN="center" HEIGHT="25" WIDTH="73"><INPUT TYPE="checkbox"
NAME="ShowCity"
      VALUE="ON" CHECKED></TD>
      <TD ALIGN="center" HEIGHT="25" NOWRAP WIDTH="80"><INPUT TYPE="radio"
VALUE="City"
      NAME="Order"></TD>
    </TR>
    <TR>
      <TD WIDTH="162" HEIGHT="25">Zip Code</TD>
      <TD WIDTH="282" HEIGHT="25"><INPUT TYPE="text" NAME="ZipCode"
SIZE="30"></TD>
      <TD ALIGN="center" HEIGHT="25" WIDTH="73"><INPUT TYPE="checkbox"
NAME="ShowZipCode"
      VALUE="ON" CHECKED></TD>
      <TD ALIGN="center" HEIGHT="25" NOWRAP WIDTH="80"><INPUT TYPE="radio"
VALUE="ZipCode"
      NAME="Order"></TD>
    </TR>
    <TR>
      <TD WIDTH="162" HEIGHT="25">Hotel Name</TD>
      <TD WIDTH="282" HEIGHT="25"><INPUT TYPE="text" NAME="HotelName"
SIZE="30"></TD>
      <TD ALIGN="center" HEIGHT="25" WIDTH="73"><INPUT TYPE="checkbox"
NAME="ShowHotelName"
      VALUE="ON" CHECKED></TD>
      <TD ALIGN="center" HEIGHT="25" NOWRAP WIDTH="80"><INPUT TYPE="radio"
VALUE="HotelName"
      NAME="Order"></TD>
    </TR>
    <TR>
      <TD WIDTH="162" HEIGHT="25">Cost per night</TD>
      <TD WIDTH="282" HEIGHT="25"><SELECT NAME="CostOperator" SIZE="1">
        <OPTION SELECTED VALUE="&lt;=">equals or less than</OPTION>
        <OPTION VALUE="&gt;=">equals or more than</OPTION>
        <OPTION VALUE="=">equals</OPTION>
      </SELECT> <INPUT TYPE="text" NAME="Cost" SIZE="11"></TD>
      <TD ALIGN="center" HEIGHT="25" WIDTH="73"><INPUT TYPE="checkbox"
NAME="ShowCost"
      VALUE="ON" CHECKED></TD>
      <TD ALIGN="center" HEIGHT="25" NOWRAP WIDTH="80"><INPUT TYPE="radio"
VALUE="Cost"
      NAME="Order" CHECKED></TD>
    </TR>
    <TR>
      <TD WIDTH="162" HEIGHT="45">Rows to Return</TD>
      <TD WIDTH="282" HEIGHT="45">From <INPUT TYPE="text" NAME="FirstRow"
SIZE="6" VALUE="1">
      To <INPUT TYPE="text" NAME="LastRow" SIZE="6" VALUE="50"></TD>
      <TD ALIGN="center" HEIGHT="45" WIDTH="73"></TD>
      <TD ALIGN="center" HEIGHT="45" NOWRAP WIDTH="80"></TD>
    </TR>
    <TR>
      <TD WIDTH="162" HEIGHT="45"></TD>
      <TD WIDTH="282" HEIGHT="45">
       <INPUT TYPE="reset" VALUE="Reset" NAME="Reset"> &nbsp;
       <INPUT TYPE="submit" VALUE="Search For Hotels" NAME="Search"></TD>
      <TD ALIGN="center" HEIGHT="45" WIDTH="73"></TD>
      <TD ALIGN="center" HEIGHT="45" NOWRAP WIDTH="80"></TD>
    </TR>
  </TABLE>
</FORM>
</BODY>
</HTML>
```

ActiveX DLL

An in-process Dynamic Link Library, which runs in the same process as the client. It provides the fastest way of accessing objects, because property and method calls don't have to be marshaled across process boundaries.

DLL

A Dynamic Link Library is a program module, with functionality for one or multiple programs.
When loaded in the memory it can be used simultaneously by multiple threads.

EazyWare
Internet and software solutions, Switzerland

VB5-CGI Objects:
Author: Stephan Schmid
Web: http://www.eazyware.com/vb5-cgi
E-mail: tools@eazyware.com

VB5-CGI Objects

A collection of powerful and easy to use Visual Basic 5 objects to develop CGI scripts on any CGI capable web server under Windows 95/NT4.0.

VB5CGI

An object from the VB5-CGI Objects collection, which handles most of the low-level CGI functionality like standard file input/output.

VB5DB

An object from the VB5-CGI Objects collection, which handles the most used database functionality.

VB5HTML

An object from the VB5-CGI Objects collection, which handles most of the high-level HTML functionality like validating query string parameter values and submitting pages.