

Web-Generator

Quasi-dynamisches Web aus der Datenbank, mit VBA programmiert. Dieser Artikel ist eine für die vereinfachte Erklärung reduzierte Form eines Programms, mit dem das PCNEWS-Web generiert wird.

Franz Fiala

<ftp://pcnews.at/pcn/65/fiala/webgen/prg/>

Erforderliche Tools

- MS Access
- Browser

In dieser Übung wird einerseits VBA-Programmierung in der Version MS ACCESS gezeigt, andererseits die Idee der Automatisierung gleichbleibender Formatierungs- und Editierungsschritte verwirklicht. Das Programm wird Schritt für Schritt entwickelt. Den Abschluss jedes Abschnitts bilden Übungsaufgaben.

Es ist nur ein Demonstrationsbeispiel, das für eine reale Anwendung erweitert werden muss.

Vorteile

- Grundsätzliche Änderungen, die das gesamte Web betreffen, können rasch durchgeführt werden.
- Das Aussehen einer Seite wird ähnlich wie bei Druckformatvorlagen eingestellt.
- Fehlerhafte Links können weitgehend vermieden werden

1. Konsistente Webseiten

Eine korrekte Webseite enthält neben dem eigentlichen Inhalt eine Reihe administrativer Elemente, deren Eintragung wichtig ist (wichtig für Suchmaschinen, Aktualität, Professionalität...), wie z.B.:

- Titel
- Bearbeitungsdatum
- Autor
- Deskriptoren
- Hinweise für Suchmaschinen
- Gleichbleibende JavaScript-Kodes

Seiten sind nur dann korrekt und vollständig, wenn alle diese Angaben vorhanden und am jeweils aktuellen Stand sind. Händisches Editieren aller dieser Merkmale ist beim ersten Mal interessant, sonst aber lästiges Beiwerk und wird daher nur allzu leicht vernachlässigt. Damit das nicht passieren kann, wird hier der Weg beschränkt, eine Datenbank zu erstellen, die ein Abbild des Web darstellt. Die Seiten, die der User abrufen, werden nicht händisch editiert, sondern durch ein Programm generiert.

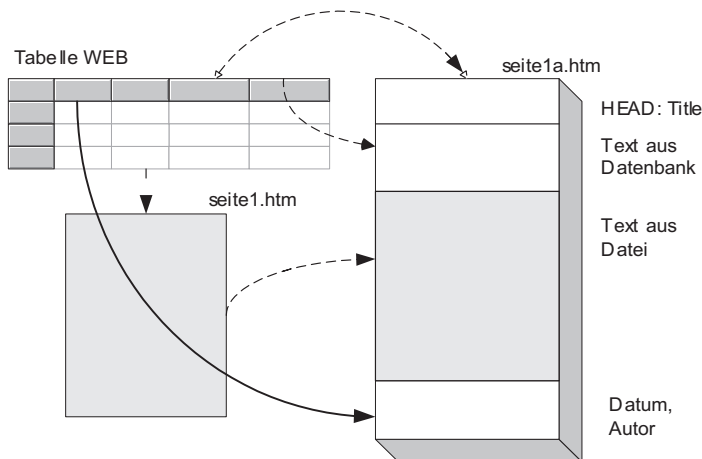
In MS ACCESS eröffnen wir eine Datenbank, z.B. **MYWEB** und in dieser Datenbank eine Tabelle **WEB**.

Tabelle

WEB

Felder

F_SRC	Dateiname einer HTML-Datei	kann auch leer sein
F_DST	zu generierende Datei	muss eingetragen sein, Name muss von F_SRC verschieden sein
H_SRC	HTML-Text	kann auch leer sein
H_TITEL	Text	muss eingetragen sein



Herstellung einer konsistenten HTML-Datei *seite1a.htm* aus einer rohformatierten Datei *seite1.htm* und aus Datenbankelementen

D_DATUM Datum wird nichts eingetragen, wird das aktuelle Datum genommen

Als Primärschlüssel verwenden wir den Namen der zu generierenden Datei. Daraus folgt, dass diese Dateinamen alle verschieden sein müssen.

	Field Name	Data Type	Description
?	N_DST	Text	Name der zu generierenden Datei
▶	F_SRC	Text	Name der zu inkludierenden Datei
	H_SRC	Text	Zu inkludierenden Text
	H_TITEL	Text	Titel
	D_DATUM	Date/Time	Bearbeitungsdatum
	N_STIL	Text	Stilname -> STIL
	L_DOKNEXT	Text	Hyperlink zum nächsten Dokument
	L_DOKPREV	Text	Hyperlink zum vorigen Dokument
	L_KAPNEXT	Text	Hyperlink zum nächsten Kapitel
	L_KAPPREV	Text	Hyperlink zum vorigen Kapitel

Felder der Tabelle WEB. Hier sind auch bereits die später hinzugefügten Felder sichtbar

Hinweis zur Notation

Feldnamen erhalten einen Präfix, um den Typ klarzustellen. Verwendet werden:

F_	Dateiname, Pfad
T_	Textfeld
R_	RGB-Farbwert
H_	HTML-Text
D_	Datumsfeld
N_	Indexfeld oder Primärschlüssel
L_	Link

Für die Generierung der Seite wird angenommen, dass der Inhalt aus zwei Quellen kommen kann:

1. Aus einem Text innerhalb der Datenbank (Feld H_SRC) und
2. Aus einem Text aus einer Datei wobei der Dateiname im Feld F_SRC gespeichert ist.

Dazu muss man sich überlegen, wie - besonders in welcher Reihenfolge - man diese Texte verarbeiten will.

F_SRC	H_SRC	inkludiert wird
leer	leer	nichts
<Name>	leer	Datei <Name>
leer	<Text>	<Text>
<Name>	<Text>	Zuerst <Text>, dann <Name>

Vereinfachungen

Für dieses Übungsbeispiel wird vereinfachend angenommen, dass sich alle Dateien im selben Verzeichnis befinden.

Beispiel

Es gibt eine voreditierte Webseite `seite1.htm`, sowie zwei weitere Seiten, deren Inhalt allein durch die Datenbank bestimmt wird. (Für einfache Seiten muss es daher gar keine voreditierte HTML-Datei geben.) Generiert werden die Seiten `seite1a.htm`, `seite2a.htm` und `seite3a.htm`. Der Name der entstehenden Datei wird als Index-Feld (ohne Wiederholung) benutzt; damit ist sichergestellt, dass nicht versehentlich ein Name zweimal vorkommt und Dateien überschrieben werden.

N_DST	F_SRC	H_SRC	H_TITEL	D_DATUM
seite1a.htm	seite1.htm	Probeseite1	Titel der Probe- seite 1	19.9.99
seite2a.htm		Probeseite2	Titel der Probe- seite 2	19.9.99
seite3a.htm		Probeseite3	Titel der Probe- seite 3	19.9.99

N_DST	F_SRC	H_SRC	H_TITEL	D_DATUM
seite1a.htm	seite1.htm	Probeseite1	Titel der Seite 1	39 08-55:11
seite2a.htm		Probeseite2	Titel der Seite 2	39 08-55:11
seite3a.htm		Probeseite3	Titel der Seite 3	39 08-55:11

Daten in der Tabelle WEB

Nun wird ein Modul erstellt, z.B. mit dem Namen GENERATOR, der alle Funktionen enthält, die zur Generierung des Web erforderlich sind.

Die erste Funktion `GeneriereWeb` erledigt folgende Aufgabe:

Für alle Einträge in der Tabelle WEB werden die Dateien N_DST angelegt. In diesen Dateien werden alle administrativen Daten im korrekten HTML-Format eingetragen und gleichzeitig als Inhalt der Rohtext aus F_SRC und H_SRC eingefügt.

Die Funktion bekommt den Namen `GeneriereWeb` und kann beispielsweise so ausschauen:

1. Öffnen der Datenbank. Da wir mit VBA innerhalb der Applikation ACCESS arbeiten, genügt dazu die Funktion `CurrentDb()`.
2. Öffnen der Tabelle WEB mit `OpenRecordset`
3. Bearbeiten aller Datensätze in einer `While`-Schleife
4. Jeder Datensatz wird mit der zweiten Funktion `GeneriereHTMLDatei` erzeugt

5. Da der Vorgang der Inklusion einer Datei in eine bestehende Datei ein ziemlich grundlegender Vorgang ist, wird dafür die Funktion `InkludiereDatei` geschrieben.
6. Schließen der der Tabelle
7. Schließen der Datenbank

Globale Größen

Das Web muss parametrierbar werden. Insbesondere bedarf es einer Pfad-Angabe, für das Web-Root. Für den ersten Versuch verwenden wir eine globale Variable im Deklarationsteil.

```
Const WebRoot = "D:\Webroot\"
Option Compare Database
```

```
Public Sub GeneriereWeb()
    Dim DB As Database, DS As Recordset
    Set DB = CurrentDb()
    Set DS = DB.OpenRecordset("WEB")
    Do While Not DS.EOF
        GeneriereHTMLDatei DS
        DS.MoveNext
    Loop
    DS.Close
    DB.Close
End Sub
```

```
Public Sub GeneriereHTMLDatei(DS As Recordset)
    Dim n As Integer
    Dim DateiQuelle, DateiZiel As String
    DateiZiel = WebRoot + DS!N_DST
    'Wir müssen bedenken, dass die DateiQuelle auch leer sein kann
    If IsNull(DS!F_SRC) Then
        DateiQuelle = ""
    Else
        DateiQuelle = WebRoot + DS!F_SRC
    End If
    n = FreeFile()
    'Zu generierende HTML-Datei öffnen und Kopf schreiben
    Open DateiZiel For Output As n
    Print #n, "<HTML>"
    Print #n, "<TITLE>" + DS!H_TITEL + "</TITLE>"
    Print #n, "<BODY>"

    'Datenbankfeld H_SRC inkludieren
    Print #n, "<P><I><B>Hier wird der Text " + _
        "aus dem Datenbankfeld H_SRC inkludiert " + _
        "(falls vorhanden):</B></I><BR>"
    If IsNull(DS!H_SRC) Then
        Print #n, "kein Text"
    Else
        Print #n, DS!H_SRC
    End If

    'Datei H DATEI inkludieren
    Print #n, "<P><I><B>Hier wird der Text " + _
        "aus der Datei F_SRC inkludiert " + _
        "(falls vorhanden):</B></I><BR>"
    Print #n, "</P>"
    'Debug.Print Dir(DateiQuelle)
    If DateiQuelle = "" Or Dir(DateiQuelle) = "" Then
        Print #n, "keine Datei"
    Else
        InkludiereDatei n, DateiQuelle
    End If
    Print #n, "</P>"

    'Bearbeitungshinweis
    Print #n, "<P>Bearbeitet am " + Format(DS!D_DATUM) + "<BR>"
    Print #n, "Aktualisiert am " + Format(Now()) + "</P>"
    Print #n, "</BODY>"
    Print #n, "</HTML>"
    Close #n
End Sub
```

```
Public Sub InkludiereDatei(dest As Integer, DateiQuelle As Variant)
    'Merge zweier Dateien.
    'dest Dateinummer jener Datei in die DateiQuelle zu inkludieren ist
    ' diese Datei ist bereits geöffnet,
    ' der Dateizeiger zeigt auf jene Stelle, an der zu inkludieren ist
    'DateiQuelle vollständiger Pfad der Quelldatei
    Dim n As Integer 'Dateinummer für die zu inkludierende Datei
    Dim Zeile As String 'Die Datei wird zeilenweise kopiert

    If Dir(DateiQuelle) = "" Then
        'Datei existiert nicht, nichts wird inkludiert
```

```
Exit Sub
End If

n = FreeFile()
Open DateiQuelle For Input As n
Do While Not EOF(n)
    Line Input #n, Zeile
    Print #dest, Zeile
Loop
Close n
'dest wird nicht geschlossen,
'weil es in dieser Funktion auch nicht geöffnet wurde.
'Die Verantwortung dafür liegt bei der rufenden Funktion
End Sub
```

Hinweis

In der Grundeinstellung von VBA kompiliert dieses Programm nicht. Unter **Tools - References** (Menübezeichnung Office 2000) müssen folgende Bibliotheken angekreuzt sein:

- Visual Basic For Applications
- Microsoft Access 9.0 Object Library
- OLE Automation
- Microsoft DAO 3.6 Object Library
- Microsoft Visual Basic for Applications Extensibility 5.3

Auch die Reihenfolge der Bibliotheken ist wichtig, oft werden Objekte in anderen Bibliotheken anders definiert.

Aufgaben

1. Jede Seite soll eine Autorenkennzeichnung erhalten
2. Auf jeder Seite ist ein Logo in der rechten oberen Ecke zu positionieren
3. Die inkludierte Seite wäre ein ASCII-Text (an der Endung zu erkennen). Diese ASCII-Dateien müssen durch einen Filter laufen, der alle in HTML nicht erlaubten (alle Umlaute, "<>&"...) in HTML-Tags umwandelt. Dieses Filter ist zu schreiben.
4. Wenn die inkludierte Datei die Tags <HTML>, <HEAD>..</HEAD>, <BODY> und </BODY> enthält, so sind diese Tags inklusive Inhalt vorher zu löschen.

2. Seitenstile

Webs bestehen aus vielen Seiten ähnlicher oder identischer Formatierung aber wechselnden Inhalten. Wenn man im Laufe der Arbeiten an einem Web auf die Idee kommt, die Hintergrundfarbe anzupassen, bedeutet das bei konventioneller Editierung eine Menge Handarbeit auf jeder Seite. Je mehr Seiten ein Web umfasst, desto ökonomischer ist es, diese nicht mit einem Editor zu editieren, sondern durch ein Programm editieren zu lassen.

Der Webgenerator erledigt diese Sache auf Knopfdruck.

Wenn man das erste Beispiel genauer betrachtet, gibt es mehrere Stellen, an denen im Programm Formatierungen vorgenommen werden, die man unter Umständen variabel gestalten möchte. (keine Formatierung im BODY-Tag, feste Formatierung des Bearbeitungshinweises)

Es wäre daher wünschenswert, Stilelemente definieren zu können, die dann die betreffenden Text- oder Bildstellen beeinflussen. Da einmal definierte Formatierungen sicher an vielen verschiedenen Seiten zur Anwendung kommen werden, empfiehlt es sich, diese Stile in einer eigenen Tabelle zu definieren und mit der Tabelle WEB mit einer Abfrage zu verbinden.

- Jeder Stil bekommt einen unverwechselbaren Namen (oder eine Nummer)
- Die Tabelle WEB erhält ein zusätzliches Stil-Feld, das diesen Namen ebenso enthält

- Der Einfachheit halber beeinflussen wir im Beispiel nur Attribute im BODY-Tag

Neue Tabelle STIL

N_STIL	Stilbezeichnung
R_TEXT	RGB-Farbe des Textes
R_BACK	RGB-Farbe des Hintergrundes
R_LINK	RGB-Farbe der Links

Field Name	Data Type	Description
N_STIL	Text	eindeutiger Schlüssel für einen gewählten Stil
R_TEXT	Text	RGB-Farbe für den Text
R_BACK	Text	RGB-Farbe für den Hintergrund
R_LINK	Text	RGB-Farbe für die Links

Felder der Tabelle STIL

Neu in Tabelle WEB

N_STIL Stilbezeichnung

Wir definieren in der Tabelle STIL zwei Stile normal und toll:

STIL	R_TEXT	R_BACK	R_LINK
normal	000000	FFFFFF	0000FF
toll	0000FF	CCCCC	FF0000

und ordnen der in der Tabelle WEB wie folgt zu (die anderen, bereits bestehenden Felder werden hier nicht gezeigt)

N_STIL	R_TEXT	R_BACK	R_LINK
normal	000000	FFFFFF	0000FF
toll	0000FF	CCCCC	FF0000

Zwei Stile (normal, toll) mit verschiedenen Farbattributen

T_NAME	N_STIL
seite1a	normal
seite2a	toll
seite3a	

Mit diesem Beispiel testen wir mit der dritten Zeile auch gleich, was passiert, wenn man keinen Stil zuweist.

Das Problem bei dieser Erweiterung ist die Formulierung der Abfrage im Visual-Basic-Code.

Dazu stehen uns zwei Möglichkeiten zur Verfügung:

1. Grafischer Entwurf einer Abfrage qWEB

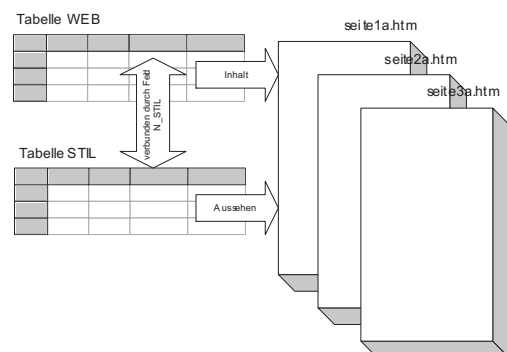
und Verwendung des Abfragenamens qWEB statt des Tabellennamens WEB

alt

```
Set DS = DB.OpenRecordset("WEB")
```

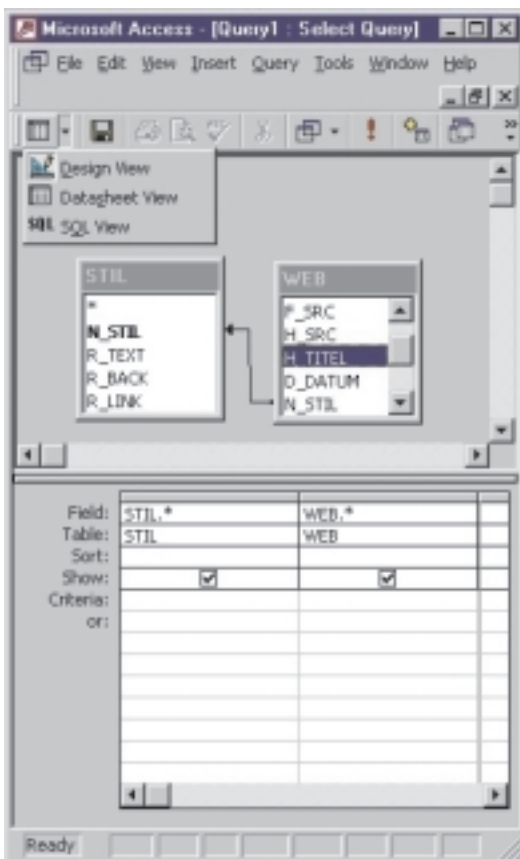
neu

```
Set DS = DB.OpenRecordset("qWEB")
```



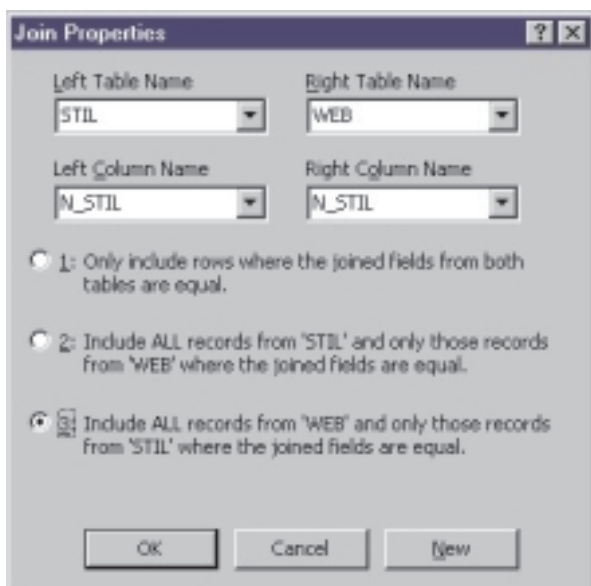
Jeder Web-Seite kann ein anderer Stil aus der Tabelle STIL zugeordnet werden.

qWEB deutet an, dass es sich im Prinzip um die Tabelle WEB handelt, aber eine besondere Abfrageform ist.



Entwurf der Abfrage qWEB. Die Beziehung zwischen den Tabellen WEB und STIL wird durch das Feld N_STIL hergestellt. Die Darstellung SQL-View ist ein der drei Darstellungsarten einer Abfrage

Diese Vorgangsweise hat den Nachteil, dass jede Änderung der grafischen Abfrage (z.B. eine Bereichseinschränkung) auch den Visual-Basic-Code mit beeinflusst.



Einstellung des Verknüpfungstyps (durch Doppelklick auf die Verbindungslinie): Alle Datensätze der Tabelle WEB und nur passende Datensätze der Tabelle STIL

Man könnte dieser Wechselwirkung entgegenreten, indem man alle Abfragen, die mit VBA-Kode zusammenhängen z.B. mit ei-

nem "v_" beginnen zu lassen, besser ist es aber gleich auf die zweite Möglichkeit überzugehen:

2. Verwendung eines SQL-Statements

MS ACCESS bietet für jeden grafischen Entwurf von Abfragen auch den zugehörigen SQL-Kode an, der auch für SQL-Laien anwendbar ist und bei mehrmaliger Anwendung auch einen Einblick in die Möglichkeiten von SQL bietet.

Von dieser Möglichkeit wird hier Gebrauch gemacht:

- Abfrage entwerfen, d.h. **Abfrage - Neu - Entwurfsansicht** - Tabellen WEB und STIL wählen.
- Das Verknüpfungsfeld ist N_STIL

Die Verknüpfung wird durch die grafische Abfragemaske auf Grund der Namensgleichheit der Felder (N_STIL) angenommen.

Als Verknüpfungstyp müssen wir angeben, dass wir alle Felder aus der Tabelle WEB wollen und nur jene Felder aus STIL, bei dem der Inhalt der verknüpften Felder übereinstimmt.

Führen wir diese Abfrage aus, dann sollten wir eine Mischung beider Tabellen erhalten, wobei wir drei Zeilen haben, wobei einer Zeile der STIL normal zugeordnet wurde, der zweiten der STIL to11 und die dritte Zeile keinen Stil enthält.

Wie man eine solche Abfrage als SQL-Statement ausführt, sieht man in der Darstellungsform SQL:

```
SELECT WEB.*, STIL.*
FROM STIL RIGHT JOIN WEB ON STIL.N_STIL = WEB.N_STIL;
```

Diesen Text können wir unverändert in VB-Kode anwenden und zwar so:

```
Dim SQLAbfrage As String
SQLAbfrage = _
    "SELECT WEB.*, STIL.* " &
    "FROM STIL RIGHT JOIN WEB ON STIL.N_STIL = WEB.N_STIL;"
Set DS = DB.OpenRecordset (SQLAbfrage)
```

Die Variable SQLAbfrage wurde eingeführt, um den Funktionsaufruf von OpenRecordset nicht zu überladen.

Beachten Sie das Zeilenverlängerungszeichen `_` und den Stringverkettungsoperator `&`. Beachten Sie auch, dass man `Set DS =` und nicht einfach `DS =` schreiben muss. Die Passage, in der das BODY-Tag geschrieben wurde, wird jetzt um die Stil-Information erweitert:

```
alt
    Print #n, "<BODY>"

neu
    Print #n, "<BODY>:"
    If Not IsNull(DS!R TEXT) Then
        Print #n, " TEXT=#" + DS!R TEXT;
    End If
    If Not IsNull(DS!R BACK) Then
        Print #n, " BGCOLOR=#" + DS!R BACK;
    End If
    If Not IsNull(DS!R LINK) Then
        Print #n, " LINK=#" + DS!R LINK;
    End If
    Print #n, ">"
```

"," am Ende der Zeile unterdrücken den Zeilenvorschub, sodass eine fließender HTML-Kode entsteht.

Die zahlreichen IsNull-Abfragen stellen sicher, dass leere Felder keine Beeinflussung des betreffenden Attributs bewirken.

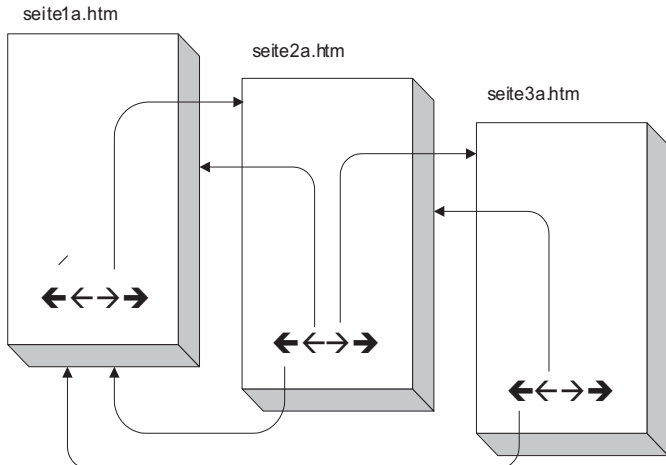
Aufgaben

1. Die beiden Texte F_SRC und H_SRC sind in eine Tabelle zu stellen, deren Hintergrundfarbe variabel sein soll.
2. Es ist ein variabler linker und rechter Rand einzustellen

3. Optional soll ein Hintergrundbild angegeben werden können.

3. Navigation

Jede Web-Seite besteht aus einem inhaltlichen Teil und aus einem Navigationsteil. Üblicherweise kann man wie in einem Buch seitenweise oder Kapitelweise verzweigen, dazu kommen Links auf Fixpunkte im Web, wie z.B. Suchen, Webindex, Antwortformulare usw.



Jede Seite zeigt auf ihren Vorgänger und Nachfolger. Alle Seiten zeigen auf die erste Seite

Wenn die Datenbank auch darüber Bescheid weiß, wie die Seiten zusammenhängen sollen und wie die entsprechenden Links heißen, dann kann der Web-Generator auch eine entsprechende Navigation vorsehen.

Um das Beispiel nicht zu kompliziert zu machen, wurden die Navigationsfelder in der Tabelle WEB händisch eingetragen. Wenn allerdings ein hierarchisches Konzept für das Web vorliegt, dann können die Wechselbezüge zwischen den Seiten auch durch das Programm selbst gefunden werden.

In der Tabelle WEB werden folgende Felder hinzugefügt

- L_DOKNEXT Name des nächsten Dokuments
- L_DOKPREV Name des vorigen Dokuments
- L_KAPNEXT Name des nächsten Kapitels
- L_KAPPREV Name des vorigen Kapitels

In unserem Beispiel ist die Navigation noch sehr einfach. Es gibt es ja noch gar keine Kapiteln sondern nur einzelne Seiten.

N_DST	F_DOKNEXT	F_DOKPREV	F_KAPNEXT	F_KAPPREV
seite1a.htm	seite2a.htm			
seite2a.htm	seite3a.htm	seite1a.htm		seite1a.htm
seite3a.htm		seite2a.htm		seite1a.htm

Das Feld für das nächste Kapitel (F_KAPNEXT) wurde daher überhaupt nicht benutzt, das Feld für das vorige Kapitel (F_KAPPREV) verweist auf die erste Seite.

N_DST	F_DOKNEXT	F_DOKPREV	F_KAPNEXT	F_KAPPREV
seite1a.htm	seite2a.htm			
seite2a.htm	seite3a.htm	seite1a.htm		seite1a.htm
seite3a.htm		seite2a.htm		seite1a.htm

Ansicht der Tabelle WEB inklusive Navigationsfelder

In der Funktion GeneriereHTMLDatei wird die neue Funktion Navigation gerufen. Da vier verschiedene Text mit demselben Verfahren zu behandeln sind, wird für das Formatieren des Hyperlink eine eigene Funktion Hyperlink formuliert. Eine kleine Besonderheit ist der Typ der beiden Übergabewerte TXT und LINK,

die eigentlich vom Typ String sind. Wenn diese Werte aber leer sind, was in diesem Fall auch einmal der Fall sein kann, wenn LINK auf kein nächstes Dokument zeigt, dann liefert die Funktion einen Laufzeitfehler. Daher wird der universellere Typ Variant gewählt, der auch leere Variablen "verträgt" und man dann diesen Umstand in der gerufenen Funktion einheitlich behandeln kann.

```
Public Sub Navigation(n As Integer, DS As Recordset)
```

```
Print #n, "<CENTER><NOBR>";
Hyperlink n, "Vorige Kapitel", DS!F_KAPPREV
Hyperlink n, "Voriges Dokument", DS!F_DOKPREV
Hyperlink n, "Nächstes Dokument", DS!F_DOKNEXT
Hyperlink n, "Nächster Kapitel", DS!F_KAPNEXT
Hyperlink n, "Webmaster", "mailto:my@name"
Print #n, "</NOBR></CENTER>";
End Sub
```

```
Public Sub Hyperlink(n As Integer, TXT As Variant, LINK As Variant)
```

```
LINK = IIf(IsNull(LINK), "", LINK)
If LINK <> "" Then
Print #n, "<FONT SIZE=1><A HREF=" +
Chr$(34) + LINK + Chr$(34) + ">";
If Not IsNull(TXT) Then
Print #n, "[" + TXT + "]";
End If
Print #n, "</FONT></A><BR>";
Else 'Es gibt keinen Link
If Not IsNull(TXT) Then
Print #n, "<FONT SIZE=1 COLOR=#555555>[" + TXT + "]"</FONT><BR>";
End If
End If
```

```
End Sub
```

Aufgaben

1. Die Navigation über Textbegriffe ist durch Symbole zu ersetzen.
2. Die generierte Seite ist durch eine Tabelle zu strukturieren und die Navigation ist in den linken Teil der Tabelle zu setzen, der eigentliche Inhalt in den rechten Teil.
3. Die Navigation ist durch einen festen Teil zu ergänzen, bei dem zu Festpunkten im Web verzweigt werden kann (Suchen, Index, FAQ...)
4. Erweitern Sie das Programm so, dass es möglich ist, alle HTML-Dateien eines Verzeichnisses in eine Navigation einzubinden, ohne dass man für jede einzelne Datei einen Tabelleneintrag erstellen muss.
5. Publizieren Sie eine Bildersammlung in einem Verzeichnis.

4. Mehrere Webs

Bei mehreren Webs ist es nützlich, ein jedes Web in einer weiteren Tabelle zu parametrisieren. Größen, die bisher in globalen Variablen oder direkt im Code zu finden waren, werden jetzt in einer eigenen Tabelle erfasst. Nützliche Parameter sind etwa:

Neue Tabelle: WEBS

Spalten:

- T_ROOT Webroot, ein Pfad auf dem aktuellen PC
- E_WEBMASTER eine E-Mail-Adresse, die auf den einzelnen Seiten verwendet werden kann
- N_WEB ein eindeutiger Name, auf den in der Tabelle WEB Bezüge genommen werden kann, das ist auch der Primärschlüssel

Beispiel für einen Eintrag

N_WEB	T_ROOT	E_WEBMASTER
my	D:\webroot\my	my@email
friend	D:\webroot\friend	friend@email
firma	D:\webroot\firma	firma@email

Quellprogramm im Web

<ftp://pcnews.at/pcn/65/fiala/webgen/prg/>

web2000.mdb	Datenbank für Office 2000
web97.mdb	Datenbank für Office 2000
STIL.txt	Textversion der Tabelle STIL
WEB.txt	Textversion der Tabelle WEB
generator.bas	Visual Basic-Programm

Neue Spalte in WEB

N_WEB

Diese Strukturänderung bedeutet daher, dass man für jedes neue Web unter dem Verzeichnis D:\webroot weitere Verzeichnisse anlegen muss. Weiters kopiert man den bisherigen Inhalt von D:\webroot in das neue Verzeichnis D:\webroot\my.

In der Tabelle WEB, die bisher nur ein einziges Web verwaltet hat, wird die neue Spalte N_WEB erzeugt, die für alle bestehenden Zeilen (seite1a.htm, seite2a.htm und seite3a.htm) die Eintragung my bekommt.

Wenn jetzt die beiden neuen Webs friend und firma mit Inhalten gefüllt werden sollen, dann muss man diese neuen Seiten nur in die bestehende Tabelle WEB eintragen. Durch die Spalte N_WEB werden diese neuen Zeilen von dem bestehenden Web unterschieden.

Damit kann dasselbe Programm eine praktisch beliebige Zahl von Webs verwalten.

Aufgabe

Die Tabelle WEBS ist zu erzeugen neben dem bestehenden Web my ein zweites, z.B. **friend** zu generieren.

Hinweis: Man benötigt dazu nur eine übergeordnete Schleife, z.B. mit dem Namen `GeneriereWebs`, die für jede Zeile der Tabelle WEBS die Funktion `GeneriereWeb` aufruft.

5. Weitere Entwicklungen

Seitenaufteilung

Wie bereits in den Aufgaben angerissen wurde, ist es vorteilhaft, eine Web-Seite mit unsichtbaren Tabellen oder mit Frames zu strukturieren. Teile der Seite enthalten die Navigationsinformationen, andere Teile den eigentlichen Inhalt.

Diese Aufteilung ist auf den ersten Blick Geschmacksache (Beispiele siehe z.B. Frontpage), doch nicht jede Anordnung ist auch für den Benutzer gleich vorteilhaft.

Welche Auswahl(en) aus den zahllosen Möglichkeiten man auch trifft, man benötigt weitere Parameter in den Tabellen, die diesen Einteilungsstil kennzeichnen.

Das Generierungsprogramm muss dann die Inhalte in verschiedene, oft auch ziemlich verschachtelte Tabellen oder Frames einfügen. Bei Frames verliert man beim händischen Editieren sehr bald den Überblick. Hier hilft der Web-Generator Ordnung zu halten, z.B. dadurch, dass die Bestandteile eines Frameset einer einheitlichen Namensgebung folgen.

Automatische Querverweise

Die händische Eingabe der Links in den Beispielen ist eigentlich nicht notwendig, wenn das Programm erkennen kann, wie die Seiten aufeinander folgen. Man kann daher eine hierarchische Nummerierung der Seiten durch eigene Spalten vornehmen, die das Programm dazu verwendet festzustellen, welche Seite die jeweils vorige Seite oder die nächste Seite ist. Damit entfällt die händische Eingabe der Links.

Grafische Oberfläche

Nachdem man ein Web mit einigen Dutzend Seiten durch Eingabe der Parameter in Tabellen aufgebaut hat, wünscht man sich, diese Vorgänge durch eine grafische Oberfläche zu unterstützen.

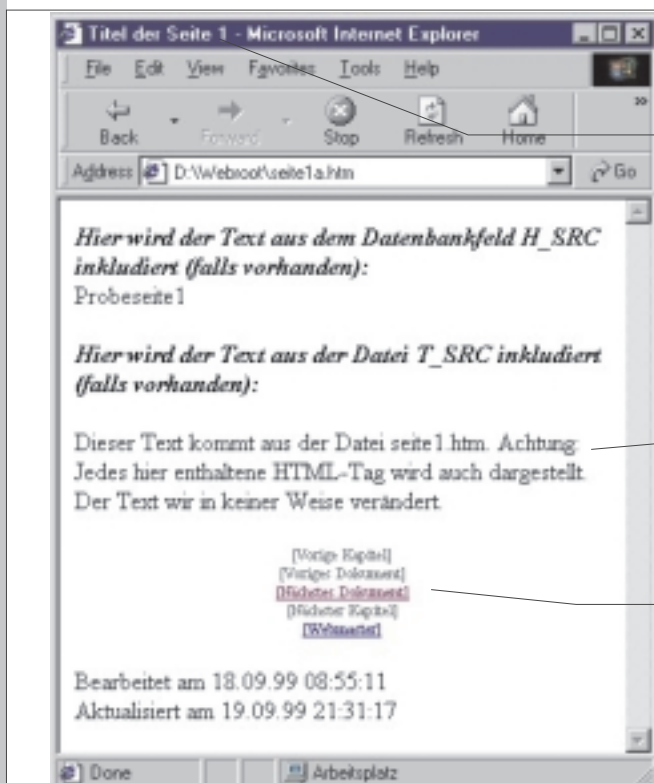
Man wird daher endbenutzerfreundliche Eingabemasken verwenden, damit nicht nur der Entwickler das Programm bedienen kann.

Überprüfung inhaltlicher Links

In den verwalteten HTML-Seiten sind in den inkludierten Seiten auch zahlreiche Links enthalten. Eine Funktion soll Links im Text aufspüren und die Richtigkeit des Links überprüfen und ein Prüfprotokoll anlegen oder eventuell einen Warnhinweis in den Text einbauen, falls ein Link gebrochen ist.

dynamisches Web

Der hier vorgestellte Webgenerator erzeugt ein Web auf Knopfdruck. Wenn man die Funktion `Generiere HTMLDatei` im Rahmen einer serverseitigen Funktion aufruft, wenn der Benutzer auf diese Seite zugreift, erhält man ein dynamisches Web, bei dem Datenbankänderungen unmittelbar beim Benutzer sichtbar werden.



wird durch die Datenbank vorgegeben

aus Datei seite1.htm

Navigation aus Datenbank; kann bei bekannter Webstruktur auch automatisch berechnet werden

Beispiel für eine generierte Webseite