

# Database Publishing mit FrameMaker und FrameScript – Handout

Ursprünglich präsentiert als Workshop auf der tekomp-Tagung 20.-22.11.2002,  
Wiesbaden, von

Dipl.-Ing. (Univ.) Michael Müller-Hillebrand, Erlangen

## Database Publishing mit FrameScript

Mit FrameScript für Windows samt dem zugehörigen Datenbank-Modul können Sie Database Publishing ohne große Lizenz-Investitionen betreiben: Von FrameMaker aus greifen Sie auf eine Datenbank zu und bauen neue Dokumente auf oder aktualisieren bestehende Dokumente.

- **Beispiel 1: Dokument-Aktualisierung**

Um eine Aktualisierung wiederholbar und stabil zu machen, müssen die zu aktualisierenden Daten im FrameMaker-Dokument eindeutig erkennbar sein, z.B. in Variablen mit definierten Namen gespeichert sein. Dann ist es unproblematisch, diese Variablen mit den entsprechenden Datenfeldern der Datenbank zu vergleichen und gegebenenfalls zu aktualisieren.

- **Beispiel 2: Dokument-Aufbau**

Beim Aufbau von Dokumenten erzeugen Sie Absätze, gegebenenfalls Tabellen und verankerte Rahmen und füllen diese mit Daten, die Sie der Datenbank entnehmen.

### *Inhalt:*

- Database Publishing mit FrameScript
- Vorbereitungen
  - ODBC-Datenquelle ISSUES einrichten
- Beispiel 1: Dokument-Aktualisierung
  - Anwendungsszenario
  - Unser Beispiel
  - Die Bestandteile der Lösung
  - Fazit
- Beispiel 2: Dokument-Aufbau
  - Anwendungsszenario
  - Unser Beispiel
  - Das Beispiel entschlüsselt
  - Fazit
  - Umbaumöglichkeiten
- Kontakt

## Vorbereitungen

Nur wenn Sie Ihren Rechner entsprechend vorbereitet haben, können Sie die Beispiele nachvollziehen.

- **FrameMaker** für Windows Version 5, 5.5, 6 oder 7, mit oder ohne SGML. Der Datenbankzugriff steht unter Mac OS leider nicht zur Verfügung.
- **FrameScript** Version 2.1R3, eine Tryout-Version steht unter <http://www.framescript.com> oder <http://elmscript.de> zur Verfügung.
- **ODBC-Datenquellen-Administrator**: Sie müssen ODBC-Datenquellen hinzufügen können; dazu benötigen Sie möglicherweise Administrator-Rechte. Unter Windows 2000 bzw. Windows XP finden Sie den entsprechenden Dialog unter *Start > Einstellungen > Systemsteuerung > Verwaltung > Datenquellen (ODBC)*.
- **Datenquelle einrichten**: siehe „ODBC-Datenquelle ISSUES einrichten“ auf Seite 3.
- **Microsoft Query** (optional): Um SQL-Abfragen einfach zu erstellen, bietet sich die mit MS Office installierte Applikation Microsoft Query an. Das Programm finde ich bei mir hier: *C:\Programme\Microsoft Office\Office\MSQRY32.EXE*. Erstellen Sie eine Verknüpfung zu diesem Programm auf Ihrem Desktop.
- **Testdaten**: Bitte laden Sie sich das ZIP-Archiv von [http://elmscript.de/res/lernen/ODBC\\_Testdaten.zip](http://elmscript.de/res/lernen/ODBC_Testdaten.zip) und entpacken Sie es an geeigneter Stelle auf Ihrem Rechner. Es enthält für jedes Beispiel ein Testdokument und ein Skript.

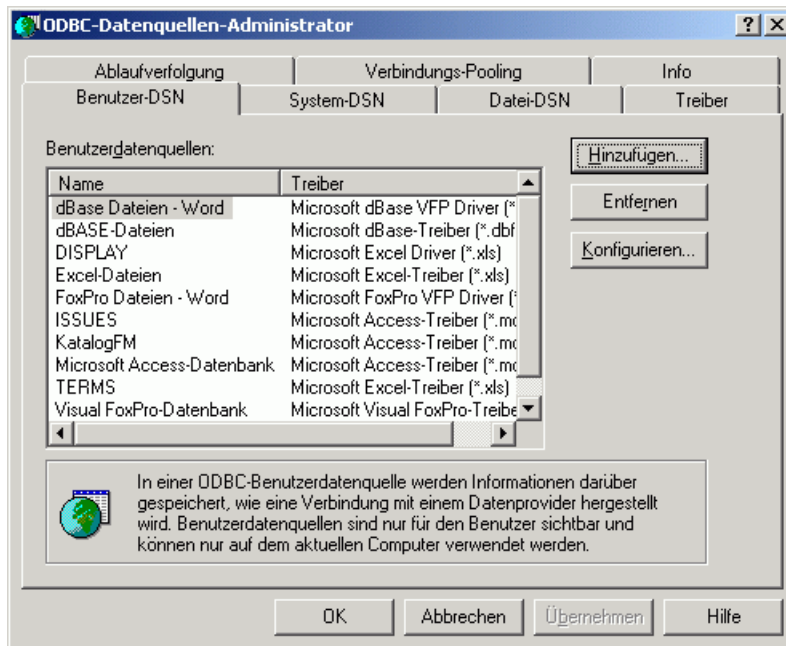
## ODBC-Datenquelle ISSUES einrichten

[Diese Anweisung entspricht der Beschreibung im Dokument *Docs\fs\DB.pdf* im FrameScript-Installationsordner zum Einrichten der ODBC-Datenquelle ISSUES.]

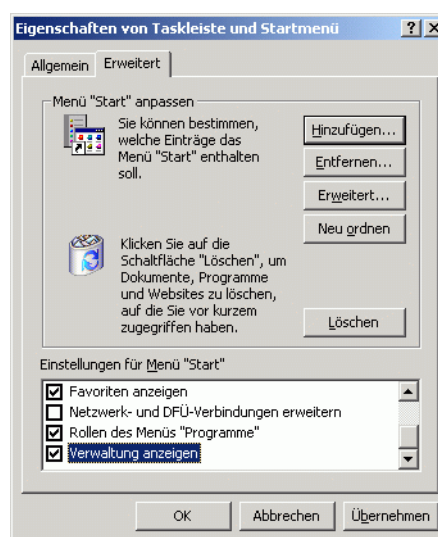
Auf eine ODBC-Datenquelle wird immer über den bei der Einrichtung angegebenen **Datenquellennamen** zugegriffen, der Name der zugrunde liegenden Datenbank-Datei ist nebensächlich. Dadurch können Sie z.B. zwischen verschiedenen Versionen einer Datenbank wechseln, ohne ein Skript modifizieren zu müssen.

Die folgende Anleitung ist mit einem deutschen Windows 2000 Professional erstellt und muss für andere Betriebssystemversionen sinngemäß angepasst werden.

1. Wählen Sie *Start > Einstellungen > Systemsteuerung > Verwaltung > Datenquellen (ODBC)*.



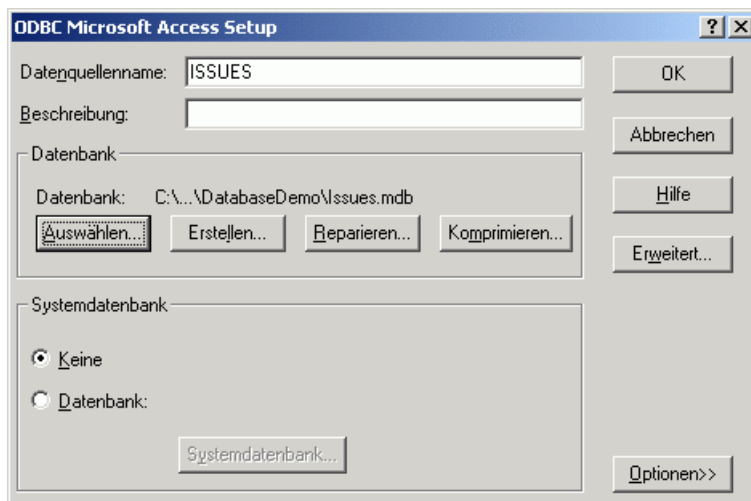
**Hinweis** Wenn Sie den Ordner »Verwaltung« unter Systemsteuerung nicht sehen, kann es sein, dass Sie nicht genügend Rechte haben, oder Sie müssen ihn unter *Start > Einstellungen > Taskleiste und Startmenü* auf der Registerkarte »Erweitert« erst aktivieren.



2. Wählen Sie *Hinzufügen* und im Dialog »Neue Datenquelle erstellen« den *Microsoft Access-Treiber*.



3. Im Dialog »ODBC Microsoft Access Setup« geben Sie nun den Datenquellennamen »ISSUES« an und wählen mit *Auswählen* die Beispieldatenbank *Demos\DatabaseDemo\Issues.mdb* aus dem FrameScript-Installationsordner.



4. Schließen Sie den Dialog mit *OK* und Beenden Sie den »ODBC Datenquellen Administrator« ebenfalls mit *OK*.

## Beispiel I: Dokument-Aktualisierung

### Anwendungsszenario

Dokumente, deren Umfang und Aufbau sich wenig ändert, die Inhalte sollen allerdings regelmäßig aktualisiert werden, z. B. Datenblätter, Preislisten.

### Unser Beispiel

Öffnen Sie im Ordner *Beispiel\_1* das Beispieldokument *VariablenAktualisieren.fm* und starten Sie mit *FrameScript > Run* das Skript *VariablenAktualisieren.fsl*.

Telefonnummern aller Mitarbeiter§	
Name§	Telefonnummer§
Name§	Telefonnummer§
Name§	Telefonnummer§
Name§	Telefonnummer§
Name§	Telefonnummer§
Name§	Telefonnummer§
Name§	Telefonnummer§
Name§	Telefonnummer§
Name§	Telefonnummer§
Name§	Telefonnummer§
Name§	Telefonnummer§

Telefonnummern aller Mitarbeiter§	
Name§	Telefonnummer§
Mosco Burrows§	2125553326§
<N.N.>§	<UNBEKANNT>§
Togo Goodbody§	2065557788§
Robin Smallburrow§	4045553933§
Salvia Bolger§	2065556068§
Marigold Cotton§	2135558221§
Linda Proudfoot§	4045556000§
Iberic Brandybuck§	2125554220§
Donnamira Boffin§	2125554400§
Hugo Bracegirdle§	2125554402§
Holman Greenhand§	2065556068§
Holfast Gardner§	2125552212§

**Anmerkung** Eine Telefonliste ist natürlich kein Dokument, welches dem genannten Anwendungsszenario entspricht, da sich hier weniger die den Personen zugeordneten Telefonnummern ändern, sondern viel eher der Personalbestand.

## Die Bestandteile der Lösung

- ODBC-Datenquelle
- Dokument mit eindeutig benannten Variablenformaten
- SQL-Anweisung zur Ermittlung der benötigten Daten
- Skript

### ODBC-Datenquelle

Die Datenquelle ISSUES aus der FrameScript DatabaseDemo (siehe *fsIDB.pdf*) verweist auf die Datenbank *Issues.mdb*, die wiederum die Tabelle *Employees* mit den Feldern *EmployeeID*, *EmployeeName*, *SupervisorID* und *WorkPhone* enthält.

EmployeeID	EmployeeName	WorkPhone	SupervisorID
1	Mosco Burrows	2125553326	
3	Togo Goodbody	2065557788	1
4	Robin Smallburrow	4045553933	1
5	Salvia Bolger	2065556068	3
6	Marigold Cotton	2135558221	4
7	Linda Proudfoot	4045556000	1
8	Ilberic Brandybuck	2125554220	7
9	Donnayira Briffin	2125554400	7

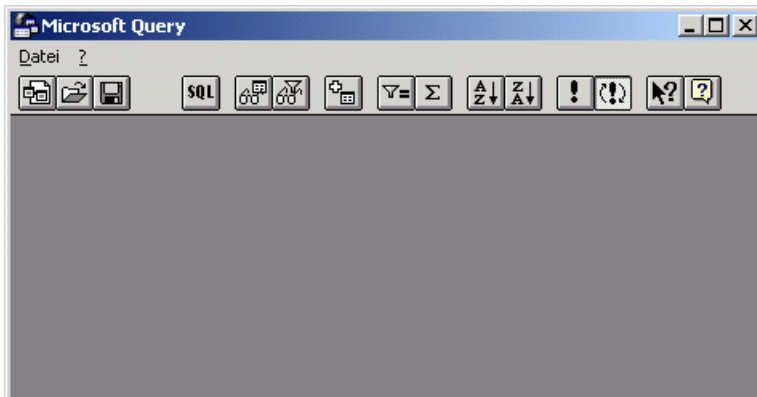
### Dokument mit eindeutig benannten Variablenformaten

Das Dokument *VariablenAktualisieren.fm* enthält die Variablenformate »Name1« bis »Name12«, sowie »Tel1« bis »Tel12« in einer Tabelle. Die Nummer am Ende des Variablennamens sei die *EmployeeID*, die Felder *EmployeeName* und *WorkPhone* sollen ausgewertet werden.

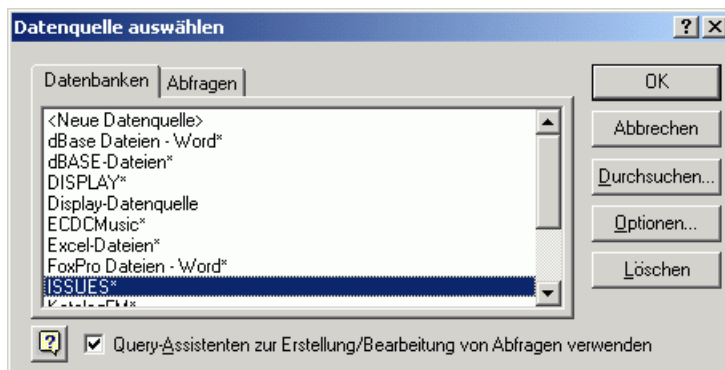
## SQL-Anweisung zur Ermittlung der benötigten Daten

Die Erstellung der SQL-Anweisung mit Hilfe von MS Query ist hier einmal ausführlich durchgeführt.

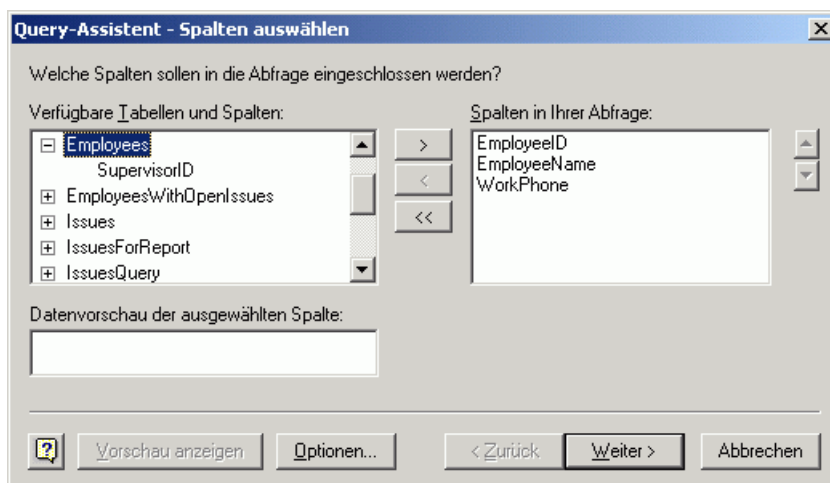
1. Starten Sie Microsoft Query.



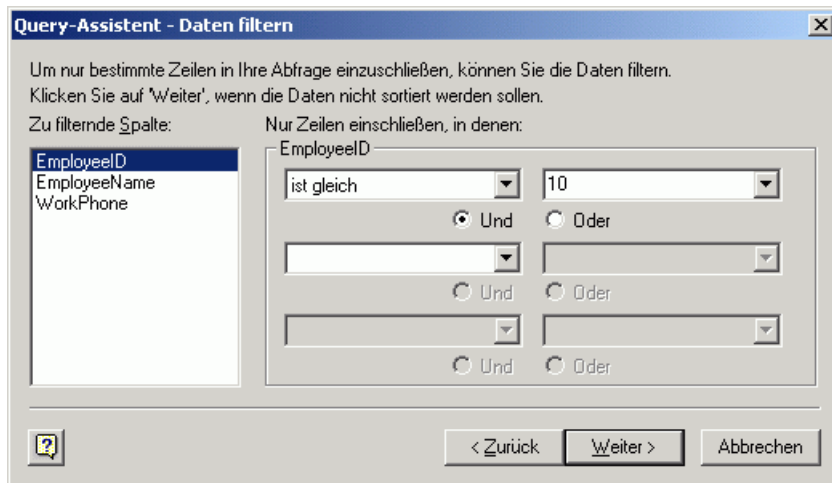
2. Erstellen Sie eine neue Abfrage (*Datei > Neu*) für die Datenquelle ISSUES und unter Verwendung des Query-Assistenten.



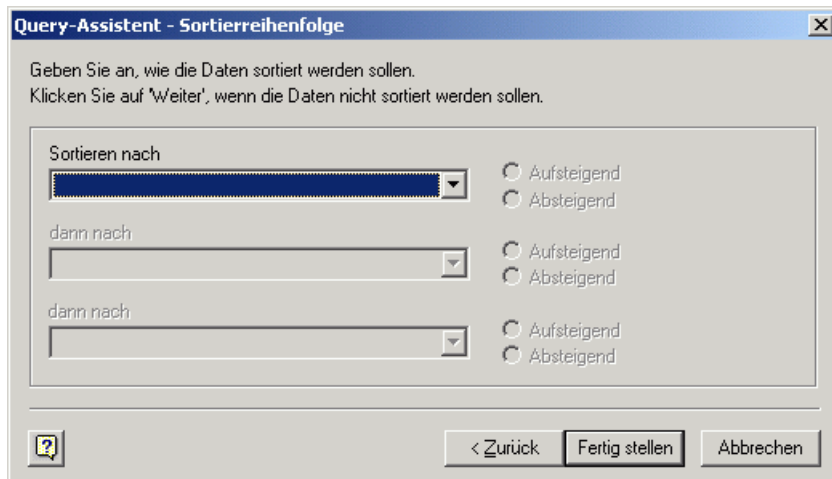
3. Wählen Sie aus der Tabelle *Employees* die Felder *EmployeeID*, *EmployeeName* und *WorkPhone*.



4. Schränken Sie die Auswahl auf einen Datensatz ein, der eine bestimmte *EmployeeID* hat. Es ist egal, welchen Wert Sie hier angeben, da wir im Skript hierfür eine Variable benutzen werden.

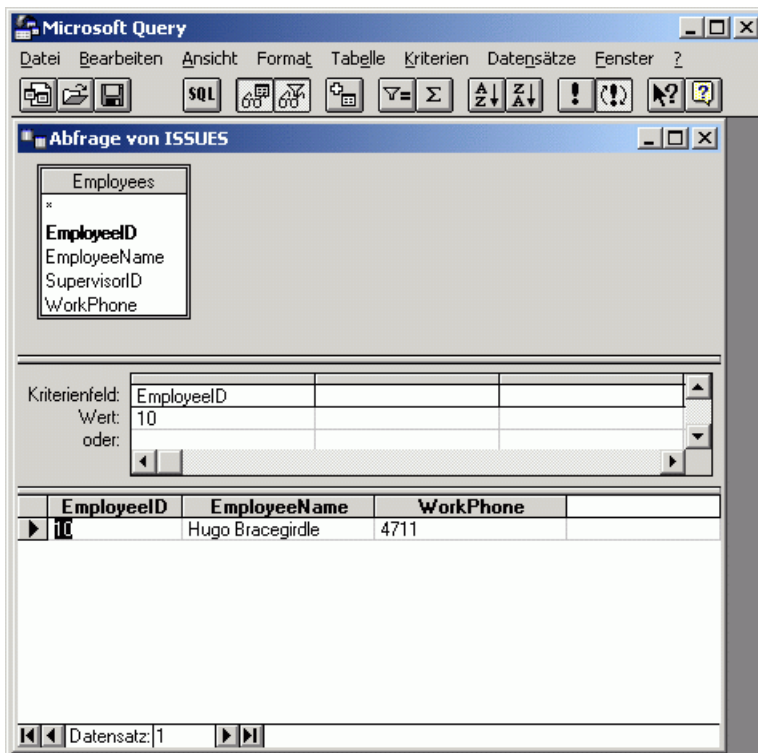


5. Einstellungen zum Sortieren benötigen wir nicht, also fertig stellen.

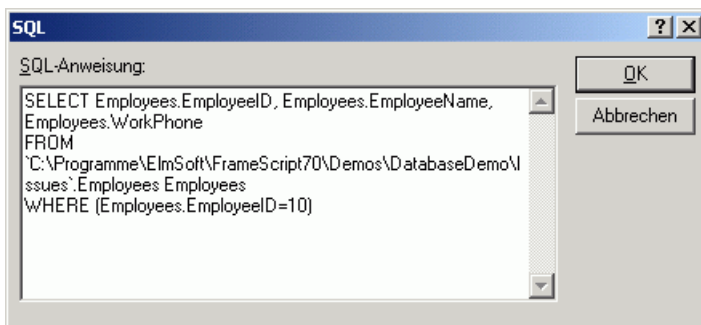




6. Die komplette Abfrage zeigt uns den auf Grund der Angaben selektierten Ergebnis-Datensatz.



7. Wählen Sie *Ansicht > SQL* um die zugehörige SQL-Anweisung zu sehen.



Diese SQL-Anweisung benötigen wir im Skript, um sie dort wie gewünscht zu verwenden.

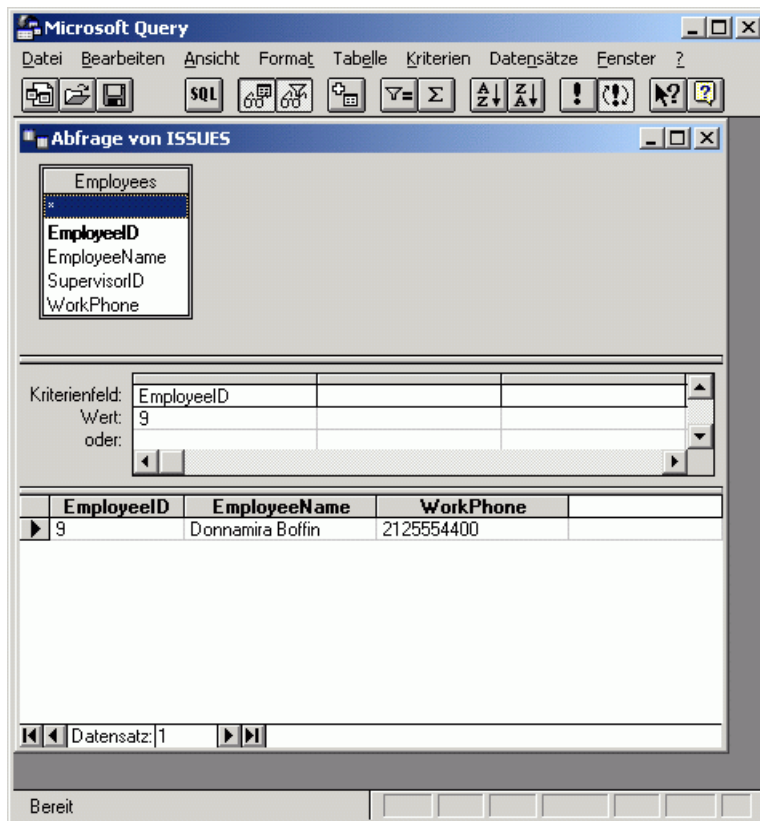
8. Kürzen Sie den überflüssigen exakten Datenbankpfad sowie redundante Tabellennamen (weil wir nur eine Tabelle abfragen):

```
SELECT Employees.EmployeeID, Employees.EmployeeName, Employees.WorkPhone
FROM ~C:\Programme\ElmSoft\FramScript70\Demos\DatabaseDemo\Issues~.Employees
Employees
WHERE (Employees.EmployeeID=10)
```

Gekürzt:

```
SELECT EmployeeID, EmployeeName, WorkPhone
FROM Employees
WHERE (EmployeeID=10)
```

9. Überprüfen Sie die Gültigkeit der Kürzung durch Eingabe in den Dialog »SQL« mit geänderter *EmployeeID*):



10. Für das Skript benötigen wir die Anweisung als String mit einer Variable an Stelle der festen *EmployeeID*:

```
Set vsSQLQuery = 'SELECT EmployeeID, EmployeeName, WorkPhone '
                + 'FROM Employees '
                + 'WHERE (EmployeeID=' + vThisNumber + ')';
```

## Skript

Das Skript *VariablenAktualisieren.fsl* enthält den kompletten Ablauf einer derartigen Aktualisierung ohne die Verwendung von Subroutinen, um das Vorgehen in einem Zug darzustellen.

Das Skript ist folgendermaßen aufgebaut:

- Prüfe, ob Dokument aktiv ist.
- Datenquellen-Informationen festlegen
- Datenbank-Objekt erzeugen und Datenbank öffnen
- Schleife durch alle Variablenformate des Dokuments
  - Einschränkung auf Variablenformate, die mit »Name« beginnen
  - Zahl nach »Name« ermitteln: `vThisNumber`
  - SQL-Anweisungs-String erstellen (siehe oben)
  - SQL-Abfrage starten
  - Ergebnis-Datensatz auswerten und *EmployeeName* sowie *WorkPhone* auslesen
  - Abfrage schließen und Abfrage-Objekt löschen
  - Gewonnene Daten nutzen und Inhalt der Variablenformate mit Name bzw. Telefonnummer aktualisieren
- Datenbank schließen und Datenbank-Objekt löschen

## Fazit

Die Formatierung der aus der Datenbank ermittelten Daten findet hier vollständig im Dokument statt. Ob Sie die Telefonnummern nun in Überschriften oder in Tabellenzellen präsentieren bleibt Ihnen überlassen, da das Skript lediglich die Variablen-Definition ändert.

## Beispiel 2: Dokument-Aufbau

### Anwendungsszenario

Dokumente, deren Aufbau sich wenig ändert, der Umfang schwankt von Mal zu Mal und die Inhalte sollen regelmäßig aktualisiert werden, z. B. Kataloge, Stücklisten (und natürlich auch Telefonlisten).

### Unser Beispiel

Öffnen Sie im Ordner *Beispiel\_2* das Beispieldokument *MitarbeiterBericht.fm* und starten Sie mit *FrameScript > Run* das Skript *DokumentAufbauen.fsl*.

**Beispiel 2: Dokument-Aufbau**

Das Skript *DokumentAufbauen.fsl* aktualisiert in diesem Dokument die Variable „EmpName“ und baut die Tabelle mit dem Tabellenformat „DBPub“ im Absatzformat „FSCTable“ neu auf.

**Mosco Burrows**

Date§	Priority§	Problem§
1996-09-07§	High§	On the proposed product label, there's a typo. The expression "Kill all our enemies" should be "Have a Nice Day" in the first sentence.§
1996-09-12§	Medium§	Our mailing didn't mention the new line, we should update it.§
1996-09-12§	Medium§	I'm looking for ideas on expanding the mailing list. Please

**Phone:§** 21255533

**Supervisor:§** NONE§

**Scroll Box**

Auswahl treffen:

- Mosco Burrows
- Togo Goodbody
- Robin Smallburrow
- Salvia Bolger
- Maigold Cotton
- Linda Proudfoot
- Iberic Brandybuck
- Donnamira Bofin
- Hugo Bracegirdle
- Holman Greenhand
- Holfast Gardner
- Bill Butcher
- Andwise Roper
- Hob Hayward**
- Gilly Baggins
- Hobson Gamgee
- Willie Banks
- Will Whitfoot
- Tom Pickthorn
- Hob Gam
- Bill Ferny
- William Hi

**Beispiel 2: Dokument-Aufbau**

Das Skript *DokumentAufbauen.fsl* aktualisiert in diesem Dokument die Variable „EmpName“ und baut die Tabelle mit dem Tabellenformat „DBPub“ im Absatzformat „FSCTable“ neu auf.

**Hob Hayward**

Date§	Priority§	Problem§
1996-09-12§	Medium§	It's past past time to put together recommendations for this year's computer upgrades. The admin staff can collect this data.§
1996-09-12§	High§	Why do I get that strange message in the Invoicing application when I try to enter a new customer? Does someone know enough about Access to figure this out?§
1996-09-12§	Medium§	There's a typo on the label. The first sentence says Exiting §

**Phone:§** 4155553244§

**Supervisor:§** Togo Goodbody - 2065557788§

## Das Beispiel entschlüsselt

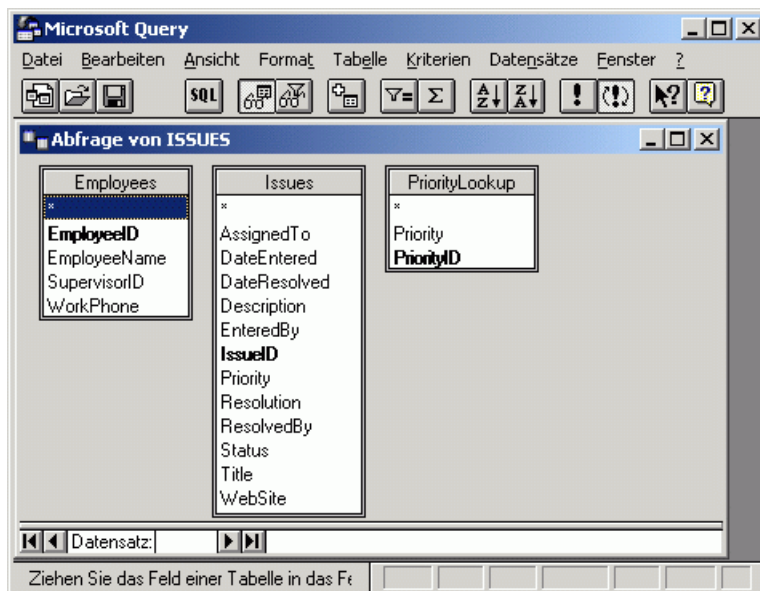
Die Bestandteile der Lösung sind:

- ODBC-Datenquelle
- Dokument mit eindeutig benannten Formaten
- SQL-Abfragen zur Ermittlung der benötigten Daten
- Skript

### ODBC-Datenquelle

Die Datenquelle ISSUES aus der FrameScript DatabaseDemo (siehe *fsIDB.pdf*) verweist auf die Datenbank *Issues.mdb*, die folgende hier interessierende Tabellen enthält:

- *Employees* mit den Feldern *EmployeeID*, *EmployeeName*, *SupervisorID* und *WorkPhone*
- *Issues* mit u. a. den Feldern *EnteredBy*, *DateEntered*, *Priority* und *Description*
- *PriorityLookup* mit den Feldern *PriorityID* und *Priority*



### Dokument mit eindeutig benannten Formaten

Das Beispieldokument *MitarbeiterBericht.fm* enthält folgende für diese Beispielanwendung relevanten Formate:

- Variablenformat »EmpName« für den Mitarbeiternamen
- Tabellenformat »DBPub« für die Berichtstabelle
- Absatzformat »FSCtable« in dem die Tabelle eingefügt wird

Diese Namen sind frei wählbar und werden in den Skripten hinterlegt.

## SQL-Abfragen zur Ermittlung der benötigten Daten

Die notwendigen SQL-Abfragen orientieren sich am Vorgehen des Skripts.

- Liste aller Mitarbeiter:

```
SELECT * FROM Employees
```

- Daten des gewählten Mitarbeiters:

```
SELECT * FROM Employees WHERE EmployeeID = <gewählte EmployeeID>
```

- Daten des zuständigen Supervisors:

```
SELECT * FROM Employees WHERE EmployeeID = <zugehörige SupervisorID>
```

- Vom Mitarbeiter angelegte »Issues«:

```
SELECT * FROM Issues WHERE EnteredBy = <gewählte EmployeeID>
```

- Prioritätsbezeichnung für jedes »Issue«:

```
SELECT * FROM PriorityLookup WHERE PriorityID = <zugehörige Priority>
```

## Skript

Das Skript *DokumentAufbauen.fsl* enthält den kompletten Ablauf einer derartigen Dokument-Generierung mit Subroutinen, und ist daher eine gute Basis für ähnliche Projekte.

Der Hauptteil des Skripts ist folgendermaßen aufgebaut:

- Prüfe, ob Dokument aktiv ist.
- Enthält das Dokument die benötigten drei Formate?
- Liste der Angestellten erstellen (*Sub BuildEmployeeList*)
- Liste der Angestellten zur Auswahl anzeigen (*Sub SelectEmployeeName*)
- Eventuell vorhandene Tabelle löschen (*Sub RemoveCurrentTable*)
- Variable mit Mitarbeiternamen aktualisieren
- Neue Tabelle mit Beschriftungen aufbauen (*Sub MakeMainTable*)
- Tabelle mit Mitarbeiterdaten füllen (*Sub FillTable*)
  - Telefonnummer des Mitarbeiter
  - Name und Telefonnummer des Supervisors
  - Für jedes »Issue«: Eingabedatum, Beschreibung und Priorität
  - Die Bezeichnung der Priorität wird aus einer separaten Tabelle gewonnen

## Fazit

Die Formatierung der aus der Datenbank ermittelten Daten findet hier teilweise im Dokument, teilweise durch das Skript statt.

- Das Aussehen der Tabelle wird durch das Tabellenformat im Dokument bestimmt, das Skript steuert die Anzahl Spalten und Zeilen sowie die Zellverbindungen in den Fußzeilen.
- Das Absatzformat mit dem Mitarbeiternamen wird ausschließlich durch das Dokument bestimmt.

## Umbaumöglichkeiten

- Die Infos zum Supervisor in die dritte Zelle der ersten Fußzeile schreiben.
- Andere Spaltenreihenfolge
- Veränderte Aufteilung der Informationen auf Kopf- und Fußzeilen
- ...

## Kontakt

Für Rückfragen erreichen Sie den Autor unter <mailto:mmh@cap-studio.de>