GASSENHUBER
Systementwicklung
www.gassenhuber.de

Thema

Scriptimport

DATAform — datenbankgestütztes Publizieren
Database-publishing mit QuarkXPress und InDesign

DATAform
Database publishing

Gewährleistung und Haftungsbeschränkungen

1. Der Käufer und der Lizenzgeber (GASSENHUBER Systementwicklung, Regensburg) stimmen darin überein, daß es nicht möglich ist, Datenverarbeitungsprogramme so zu entwickeln, daß sie in allen Anwendungsbedingungen fehlerfrei sind. Der Lizenzgeber gewährleistet die Übereinstimmung des Programms mit den Beschreibungen des vorliegenden Handbuchs mit Ergänzungen. Die Gewährleistungsfrist beträgt sechs Monate ab Lieferung.
2. Die Gewährleistung erstreckt sich nicht auf Mängel, die durch Abweichen von dem für das Programm vorgesehenen und im Handbuch oder den Ergänzungen angegebenen Einsatzbedingungen verursacht werden.
3. Der Lizenzgeber ist zur Beseitigung von Programmmängeln verpflichtet, die innerhalb der Gewährleistungsfrist auftreten und vom Käufer schriftlich in nachvollziehbarer Weise dem Lizenzgeber mitgeteilt werden. Eventuelle Mängel des DV-Programmes sind, soweit sie offenkundig sind oder werden, dem Lizenzgeber innerhalb von zwei Wochen nach Feststellung des Mangels anzuzeigen. Werden die Anzeigepflichten nicht erfüllt, ist jegliche Gewährleistung ausgeschlossen. Mängelbehebung ist durch Programmänderung, durch Fehlerumgehung, aber auch durch Überlassung eines anderen, dem Vertragszweck entsprechenden DV-Programmes möglich.
4. Werden entsprechend Punkt 3 Fehler festgestellt und gelingt es nicht, innerhalb einer angemessenen Frist durch Nachbesserung die erheblichen Abweichungen von der Funktionsbeschreibung zu beseitigen oder so zu umgehen, daß dem Käufer eine vertragsgemäße Nutzung des Programms ermöglicht wird, kann der Vertragspartner eine Herabsetzung der Lizenzgebühren verlangen oder die Lizenz für das Programm fristlos kündigen.
5. Jede Vertragspartei haftet für von ihr zu vertretende Schäden insgesamt maximal bis zur Höhe der Einmalgebühr des DV-Programms.
6. Der Lizenzgeber haftet nicht für mangelnden wirtschaftlichen Erfolg, mittelbare Schäden und Folgeschäden und für Schäden aus Ansprüchen Dritter mit Ausnahme von Ansprüchen aus Verletzung von Lizenzrechten Dritter. Der Lizenzgeber haftet nicht für die Wiederbeschaffung von Daten.

Quark, Inc. makes no warranties, either express or implied, regarding the enclosed computer software package, its merchantability, or its fitness for any particular purpose. Quark, Inc. disclaims all warranties including, but not limited to the warranties of the distributors, retailers and developers of the enclosed software.

Without limiting the foregoing, in no event shall Quark, Inc. be liable for any special, indirect, incidental, or consequential damages in any way relating to the use or arising out of the use of the enclosed software. Quark, Inc.´s liability shall in no event exceed fifty dollars ($50.00).

The exclusion of implied warranties and/or the exclusion ot limitation of incidental or consequential damages is not allowed in some areas, so these exclusion and limitations may not apply to you.

Impressum

Stand: April 2013

DATAform DVD [9]

DATAform-Datenbank Version 10.2 (25)

DATAformXTension Version 9
für QuarkXPress Version 9

DATAformXTension Version 8
für QuarkXPress Version 8

DATAformPlugin Version 8
für InDesign 8, CS6

DATAformPlugin Version 7
für InDesign 7, CS5

Für MacOS X und Windows

GASSENHUBER Systementwicklung
D 93059 Regensburg
www.gassenhuber.de

DATAform-Zentrale
Tel. 0941 / 79 55 05
Fax 0941 / 79 55 07
info@gassenhuber.de

Internet: **www.gassenhuber.de**
Mit neuesten Informationen und Updates.

Inhalt

Thema Scriptimport

Vorbemerkung
DATAform beinhaltet zwei Schnittstellen für Textdateien:

- Einen einfach zu bedienenden Scriptimport für den Import verschiedener Textdateien. Die Textdateien müssen nicht nach einem vorgegebenen Schema strukturiert sein, die Zuordnung zu Feldern und Elementen definiert man dann durch ein Importscript.
- Import und Export von Textdateien im DATAform-Textformat. Ein Textimport über diesen Weg erfordert meist eine Anpassung der exportierenden Datenbank.

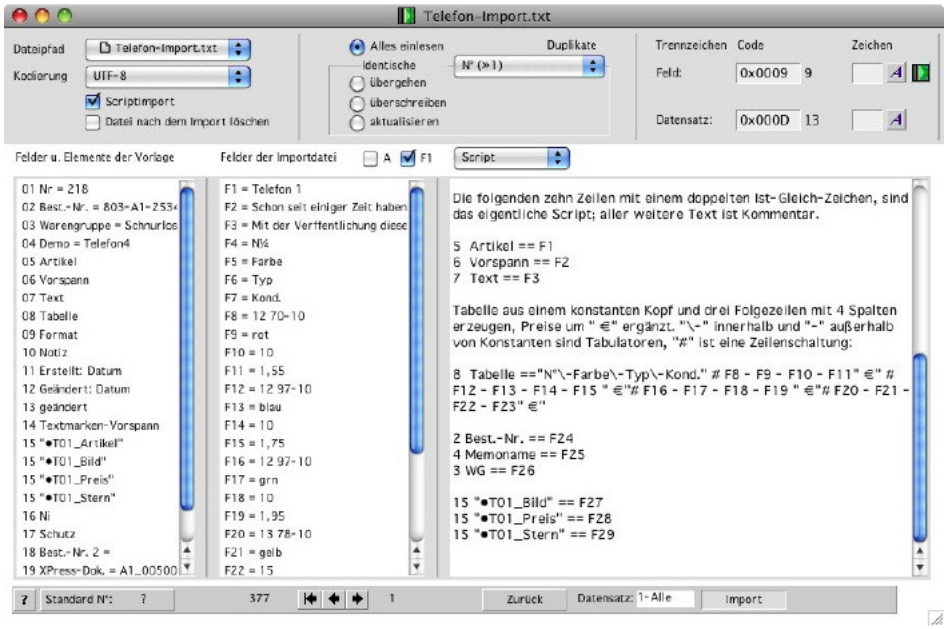
Der Scriptimport dient dem Import von Textdateien mit zwei Trennzeichen und beliebiger Feldreihenfolge. Feld- und Datensatztrenner sind typischerweise Tabulator und Zeilenschaltung. Die Feldreihenfolge wird durch ein Importscript definiert.
Das Importsscript erlaubt zudem die Summierung von importierten Feldern in einzelnen DATAform-Feldern, die zusätzliche Zuweisung von konstanten Texten und vieles mehr.

Aus einer einfachen Tab-Zeilenschaltung-Tabelle kann durch den Scriptimport ein vollständiger DATAform-Artikel aufgebaut oder können bestehende Artikel feldgenau aktualisiert werden.

Als Vorlage wird der gerade geöffnete DATAform-Artikel angesehen; er dient mit seinen Elementen als Muster, in das die Felder der Importdatei geschrieben werden. Beim Import neuer Artikel wird die Vorlage dupliziert, und die Felder werden dem Script entsprechend geändert. Artikelfelder, die das Script nicht anspricht, bleiben, wie sie sind.

Anwendungsbeispiel

- Kopieren Sie sich den Ordner DATAform-Datenbank/DATAform-Demos/DATAform-Script-Demo von der DATAform-DVD.
- Öffnen Sie in der Telefondemo einen Artikel mit den Elementen „•T01_Bild“, „•T01_Preis“, „•T01_Stern“ etc. als Artikelvorlage.
- Wählen Sie DATA/Import und öffnen Sie die Datei DATAform-Script-Demo/Telefon-Import.txt im vorhin kopierten Ordner.
- Klicken Sie auf das Ankreuzfeld „Scriptimport“ links/oben, Sie erhalten etwa dieses Bild:



- Wählen Sie den Befehl „Script laden…“ im Script-Klappmenü und öffnen Sie die Datei DATAform-Script-Demo/Telefon-Script.txt – wie in der Abbildung bereits geschehen.
- Klicken Sie dann auf „Import“ in der Fußleiste. Zwei vollständige Artikel werden erzeugt.

DATAform enthält eine einfach zu bedienende Scriptsprache, die es erlaubt, Felder beliebig zuzuordnen und Feldinhalte und Konstanten zu addieren. Ein Importscript in DATAform ist ein Text, der eingetippt oder durch einfaches Klicken geschrieben werden kann.

Ein Importscript schreiben
So schreiben Sie ein Script. Alle Script-Funktionen in 10 Abschnitten a–j:

a) Trennzeichen definieren
Klicken Sie unter Trennzeichen auf das DATAform-Symbol (d.h. die Werkseinstellung), um eine Tab-Zeilenschaltung-Datei, z.B. eine Excel-Text-Datei, zu importieren.

b) Importfelder zuordnen
Löschen Sie das alte Script im rechten Feld. Klicken Sie links z.B. auf „8 Tabelle“, im Scriptfeld erscheint: 8 Tabelle ==
Klicken Sie dann in der Mitte auf ein Feld der geöffneten Datei, Sie erhalten:
8 Tabelle == F9
Das Tabellenfeld in DATAform wird dadurch mit dem Feld 9 der Importdatei gefüllt. Für jeden Datensatz der Importdatei wird jeweils der Inhalt des 9. Feldes in einen neuen Artikel geschrieben oder wird – je nach Duplikateinstellungen – ein Artikel aktualisiert.

c) Importfelder summieren

Der obere Teil des Script-Klappmenüs zeigt verschiedene Sonderzeichen.
Fügen Sie weitere Felder und Sonderzeichen ein:

8 Tabelle == F9 - F10 - F11 - F12 # F13 - F14 - F15 - F16

Das Tabellenfeld erhält die Inhalte Feld 9 + Tabulator (-) + Feld 10 bis Feld 12; dann folgt – durch # – eine Zeilenschaltung und eine weitere Zeile aus Feldern und Tabulatoren.

Ein neues Feld wird durch einen Klick in die mittlere Liste immer dort eingefügt, wo die Schreibmarke steht. Ist Text im Scriptfeld markiert, wird er dabei überschrieben.

Zuerst wird immer ein Feld aus der linken Liste angeklickt, dann erst aus der mittleren. Klickt man in die linke Liste, wird automatisch eine neue Zeile begonnen.
Statt zu Klicken kann man die Felder aus den beiden Liste auch per „Ziehen und Loslassen“ in das Scriptfeld einfügen.

d) Rechnen und Formatieren

Mit den nächsten beiden Funktionen des Script-Klappmenüs lassen sich Feldinhalte der Importdatei direkt beim Import umrechnen und z.B. mit Währungskennzeichen versehen oder läßt sich ein Sortierwert berechnen und mit vorgängigen Nullen auffüllen.

[Importfeld;Rechenformel;Ausgabeformat]

Der Befehl fügt rechts von == dieses Beispiel ein:
8 Tabelle ==[F1;+2*1,16;###.##0,00 €]

Die Formel enthält getrennt durch Strichpunkte 3 Bestandteile:
- Importfeld: Das Importfeld ist F1, das erste Feld der Importdatei.
- Rechenformel: Der Rechenausdruck, der auf #N angewandt werden soll. Die Rechenfor-

mel lautet +2*1,16. Es wird 2 (+2) addiert und das Ergebnis wird mit 1,16 (*1,16) multipliziert. Die einzelnen Schritte werden von links nach rechts abgearbeitet; die Kommutativgesetze gelten nicht.
Hat F1 beipielsweise den Inhalt 1, so ergibt sich 1+2*1,16 = 3*1,16 = 3,48.
Als Operatoren können verwendet werden: +, -, *, / und ^ für Potenz und Wurzel.
Ausdrücke können durch runde Klammern zusammengefaßt werden, z.B.:
8 Tabelle ==[F1;+(2*1,16);###.##0,00 €]
Ist F1 gleich 1, so ergibt sich nunmehr für Feld 8 Tabelle: 1+(2,32) = 3,32
In Feld 8 Tabelle steht dann 3,32 €

- Ausgabeformat: Das Ausgabeformat ist ###.##0,00 €
  # ist Platzhalter für eine Ziffer falls die Ziffer vorhanden ist.
  0 ist Platzhalter für eine Ziffer und Füllzeichen wenn die Ziffer fehlt.
  . ist eine optionaler Tausendertrenner.
  Sind nicht genügend #- oder 0-Platzhalter links vom Komma vorhanden, wird das Zeichen < ausgegeben; rechts vom Komma werden die Ziffern abgeschnitten.
  Leertaste und andere Zeichen werden wie sie sind ausgegeben.

Das Ergebnis des gesamten Ausdrucks:
[F1;+2*1,16;###.##0,00 €]

lautet bei F1 = 1 =>	1+2*1,16 =>	3*1,16 =>	3,48 =>	3,48 €
bei F1 = 8 =>	8+2*1,16 =>	10*1,16 =>	11,6 =>	11,60 €

[Importzähler;Rechenformel;Ausgabeformat]

Der Befehl fügt rechts von == dieses Beispiel ein:
2 Sortierung ==[N#;*50;000000]

Die Formel enthält getrennt durch Strichpunkte 3 Bestandteile:
- Importzähler: N# ist eine Variable, die die Anzahl der gelesenen Sätze der Importdatei während des Importvorgangs wiedergibt. Sie entspricht der Zahl die rechts neben den Blätternpfeilen im Importscript-Dialog angezeigt wird.

- Rechenformel: *50 ist der Rechenausdruck, der auf #N angewandt werden soll, also zum Beispiel *50

- Ausgabeformat: 000000, der 3. Bestandteil gibt z.B. die aufzufüllenden Nullen an.

Beispiel 1: Beim ersten gelesenen Satz ist N#=1; wird mit 50 multipliziert => 50; wird auf 6 Nullen aufgefüllt => als Ergebnis wird 000050 in das Feld „2 Sortierung“ eingetragen.

Beispiel 2: Beim 80.ten gelesenen Satz ist N#=80; wird mit 50 multipliziert => 4000; wird auf 6 Nullen aufgefüllt => als Ergebnis wird 004000 in Feld „2 Sortierung“ eingetragen.

Beispiel 3: Die Scriptzeile:
8 Tabelle == F1 " Original"#[F1;*2;]" doppelt" # [F1;;####,00 €]" Original + €" # "Importsatz Nr.: "[N#;;] # "Sortiernummer aus N#: "[N#;*50;00000000] # "Sortiernummer aus F1: "[F1;*50;00000000]

Liefert bei Satz 2 und F1 = 1 diesen Text im Tabellenfeld:
1 Original
2 doppelt
1,00 € Original + €
Importsatz Nr.: 2
Sortiernummer aus N#: 00000100
Sortiernummer aus F1: 00000050

e1) Leere Elemente löschen

Verwendet man eine Vorlage mit mehreren Elementrahmen, von denen aber nur einige beim Import gefüllt werden, so lassen sich die nicht gefüllten Elemente gleich beim Import löschen.

Operator ==! für Elementefelder 15.
==! bedeutet: Das Ergebnis des rechten Teils der Scriptzeile darf nicht leer sein, andernfalls wird das Element gelöscht.

Beispiele:
15 "Bild 05" ==! F1
Das Element "Bild 05" wird in allen importierten Artikeln gelöscht, bei denen F1 leer ist.

15 "Bild 05" ==! F1"Text"
Das Element "Bild 05" wird nie gelöscht, da der rechte Teil immer mindestens die Konstante "Text" liefert.

15 "Bild 05" ==! F1 F13
Das Element "Bild 05" wird gelöscht, wenn F1 und auch F13 leer sind.

15 "Bild 05" ==! ""
Das Element "Bild 05" wird immer gelöscht, da der rechte Teil konstant leer ist

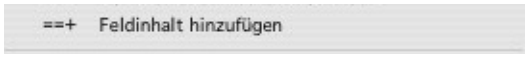
15 "Bild 05" == "Text"
15 "Bild 05" ==! ""
Gibt es mehrere Zuweisungszeilen zum selben Element, zählt die letzte. Die erste setzt den Inhalt auf „Text“, die letzte löscht ihn: Das Element "Bild 05" wird immer gelöscht.

15 "@Artikel 20" ==! ""
Löscht sowohl das Element "•Artikel 20", als auch "Artikel 20", als auch "NeuArtikel 20". Gelöscht wird dabei das erste gefundene Element des Artikels, siehe e2).

e2) Mehrere Elemente löschen

Seit DATAform 10 (20), 12.2.2013 können mit einer einzelnen Scriptzeile auch mehrere Elemente gelöscht werden. Der Bildname muß dazu mit @ beginnen und mit @ enden.
Beispiel:
15 "@Bild@" ==! ""
Löscht alle Elemente deren Namen „Bild“ enthalten. Hingegen löscht
15 "@Bild" ==! ""
nur das erste gefundene Element, das auf Bild endet.
15 "Bild@" ==! ""
Nur das erste gefundene Element, das mit Bild beginnt wird gelöscht.
Mehrere Elemente werden durch eine einzelne Zeile also nur gelöscht wenn ein @ am Anfang und am Ende des Namens in der Scriptzeile steht.

f) Feldinhalt hinzufügen



Die importierten Feldinhalte lassen sich den Feldern bestehender Artikel (Aktualisieren) oder der Importvorlage auch hinzufügen.

Operator ==+ fügt den importierten Inhalt dem alten Feldinhalt hinzu. Beispiele:

8 Tabelle ==+ #F1
Erweitert den Inhalt des Tabellenfeldes um eine Zeilenschaltung und um das Importfeld F1.

8 Tabelle == #F1
Ersetzt hingegen den bisherigen Inhalt durch den neuen.

Der ==+ Operator kann bei allen Textfeldern, sowie den Elemententexten und Bildnamen angewandt werden.

Es können auch mehrere Zeileninhalte nacheinander hinzugefügt werden, wie:
if (F12 #? "") 08 Tabelle == F12 - F19
if (F31 #? "") 08 Tabelle ==+ # F31-F38
if (F50 #? "") 08 Tabelle ==+ # F50-F57
Das Tabellenfeld enthält dadurch am Ende den Inhalt von sechs Feldern.

g) Konstanten verwenden

Die importierten Felder können beliebig durch konstante Texte ergänzt werden, z.B.:
15 "•T01_Preis" == F22 " €" - F28 " €"
ergänzt die Preise in Feld 22 und 28 jeweils um „ €“, sodass im Textelement „•T01_Preis“ in DATAform z.B. „2,35 € Tab 3,95 €“ erscheinen wird.

Innerhalb von "Konstanten" sind alle Zeichen erlaubt außer den 3 Zeichen: Zollzeichen ("), umgekehrter Schrägstrich (\) und Zeilenschaltung.
Das Zollzeichen muß innerhalb von Konstanten durch \" maskiert werden, z.B.:
"<f\"Helvetica\">" ergibt dann nach dem Import die XPressMarke <f"Helvetica">
Eine Zeilenschaltung in konstanten Texten wird durch \# erzeugt.

Die folgende Tabelle zeigt die Verwendung dieser Sonderzeichen:

Zeichen	außerhalb von Konstanten	"innerhalb von Konstanten"
Tabulator	-	\- (oder Tabulator)
Leertaste	.	Leertaste
Zeilenschaltung	#	\#
Zollzeichen	nicht erlaubt	\"
Schrägstrich	nicht erlaubt	\\

Erläuterungen:
Um außerhalb von konstanten Texten einen Tabulator hinzuzufügen, genügt ein „-“; innerhalb von Konstanten erhält man einen Tabulator durch „\-“.

Um außerhalb von Konstanten eine Zeilenschaltung einzufügen, schreibt man „#“; innerhalb von Konstanten schreibt man „\#“. Zeilenschaltungen selbst sind im Script immer das Ende einer Anweisungszeile.

h) Elemente ansprechen

In der linken Liste des Dialogs ab Zeile 15 sehen Sie die Elemente des aktuell geöffneten Artikels. Die Namen der Elemente können wie weitere Feldnamen verwendet werden. Sie müssen eindeutig sein. Es werden die Text- und Bildelemente des aktuellen Artikels angezeigt.

Beispiele. Der Text bzw. der Bildname werden in ein Element übernommen:

15 "•Preis" == F28
Dem Textelement „•Preis“ wird der Text aus Feld 28 zugeordnet.

15 "•T01_Bild" == F25
Dem Bildelement wird als Bildname das Feld 25 der Importdatei zugeordnet.

Weitere Elementfelder ansprechen
Möglich ist auch ein Scriptimport in die Element-Felder Textmarken/|Format/Pfad, sowie das Feld DATAformMarken. Im vorhergehenden Punkt wurde das Elementfeld für den

Text bzw. Bildnamen angesprochen. Über die DATAform-Elemente-Feldnummern 04 und 06 können pro Element zwei weitere Felder angesprochen werden.
Das Script-Klappmenü zeigt diese Zeilen:

```
15     "Element" ➡ Text/Bildname
15.04 "Element" ➡ Textmarken/|Format/Pfad
15.06 "Element" ➡ DATAformMarken
```

Beispiel: Import von Bildnamen und Pfaden
Name des Bildes aus F1, wie im vorhergehenden Punkt beschrieben:
15 "•A 4 Zeichnung" == F1
(15 ist gleichbedeutend mit 15.03 Elemente.Text/Bildname)

Und zusätzlich der Pfad oder Bildordner des Bildes aus F2:
15.04 "•A 4 Zeichnung" == F2

Oder den Bildordner als Konstante, wenn der Bildordner bekannt ist:
15.04 "•A 4 Zeichnung" == "•Bilder1"

Beispiel: Import von Text und Formaten
Text des Elements aus F3 etc., wie im vorhergehenden Punkt beschrieben:
15 "•A 2Tabelle" == F3 - F4 - F5

Und zusätzlich Textmarken oder ein Format für das Element aus F6:
15.04 "•A 2Tabelle" == F6

Oder ein konstantes Format für alle diese Elemente:
15.04 "•A 2Tabelle" == "|Format1"

Beispiel: Import von DATAformMarken
Fügt dem Feld der Vorlage z.B. die DATAformMarke *Z3 hinzu:
15.06 "•A 4 Zeichnung" ==+ "*Z3"
Die Bilder dieses Imports werden dadurch proportional in den Rahmen skaliert.

i) Scriptimport mit if( )

Den Zeilen in einem Importscript kann eine Bedingung vorangestellt werden, nach dem Muster: if (A =? B) 7 Text == F5
Der rechte Teil „7 Text == F5“ ist eine normale Zeile; sie wird jetzt nur dann ausgeführt wenn der linke Teil „if (A =? B)“ zutrifft; andernfalls wird die ganze Zeile ignoriert.

Die Bedingung „if (A =? B)“ trifft zu, m.a.W. liefert wahr, wenn A und B gleich sind.
A und B sind jeweils Ausdrücke wie F1 oder F2 oder "Text". Z.B.:
if (F1 =? F1) liefert immer wahr, da der Inhalt des ersten Feldes mit sich selbst gleich ist.
if (F1 =? F2) liefert nur wahr, wenn die beiden Felder dasselbe enthalten.
if ("Äpfel" =? "Birnen") liefert immer falsch.

Der Scriptimport erlaubt Textvergleiche, zeichensensible Textvergleiche und numerische Vergleiche.

Textvergleiche
if (F3 =? "") 8 Tabelle == " F3 ist leer: " F3
Ist das Feld F3 des gerade gelesenen Importsatzes leer (""), wird in das Tabellenfeld " F3 ist leer: " geschrieben. Andernfalls geschieht nichts.

if (F3 #? "") 8 Tabelle == " F3 ist nicht leer sondern: " F3
Ist das Feld F3 nicht leer (ungleich # einer leeren Zeichenkette ""), wird in das Tabellenfeld " F3 ist nicht leer sondern: " ... geschrieben.

if (F3 =? F4) 8 Tabelle == F5
Enthalten F3 und F4 denselben Text, dann soll F5 in die Tabelle geschrieben werden.

if (F2 F3 F4 =? "") 8 Tabelle == "F2 F3 F4 sind alle leer"
Ergibt F2 F3 F4 keinen Text, dann erscheint im Tabellenfeld "F2 F3 F4 sind alle leer".

if (F2 F3 F4 #? "") 8 Tabelle == "F2 F3 F4 sind nicht alle leer"
Liefert wahr, wenn mindestens eines der Felder F2, F3 oder F4 Text enthält.

if (F2 F3 - F4 #? "") 8 Tabelle == F2
Fehler: Liefert immer wahr, da A mindestens einen Tab (-) liefert!
Warum? Schritt für Schritt:
1. if (F2 F3 - F4 #? "") liefert immer wahr, denn
2. F2 F3 - F4 liefert mindestens einen -, also auch wenn alle drei Felder leer sind ist das Ergebnis ein „-“, ein Minuszeichen, gleichbedeutend mit einen Tabulator
3. Nun wird der Tabulator, das Minuszeichen mit einer leeren Zeichenkette ("") verglichen. Es wird gefragt ist ein Tabulator ungleich einer leeren Zeichenkette?
4. Die Antwort lautet immer Ja.
5. Daher liefert dieses if (F2 F3 - F4 #? "") immer wahr und ist deshalb überflüssig und eher verwirrend.

if (F1 =? "dataform")
Liefert wahr für F1 = "DATAform" oder "Dataform" oder "DATAFORM" usw.
Liefert falsch für F1 = "DATAform ist ein Programm".

if (F1 =? "dataform@")
Liefert wahr, wenn F1 mit "DATAform" beginnt.

if (F1 =? "@dataform")
Liefert wahr, wenn F1 mit "DATAform" endet.

if (F1 =? "@dataform@")
Liefert wahr, wenn F1 "DATAform" enthält.

Textvergleiche zeichensensibel
if (F1 i=? "DATAform")
Liefert wahr, wenn F1 identisch mit "DATAform" in dieser Schreibweise ist, sonst immer falsch.

Numerische Vergleiche
if (F1 n>? "10") 10 Notizen == "F1 ist größer als 10"
Ist F1 größer als 10, dann …
A und B werden in eine Zahl umgewandelt und dann verglichen. Als Dezimaltrenner kann Punkt oder Komma verwendet werden.

if (F1 n>=? "10,09") 10 Notizen == "F1 ist mindestens 10,09"
Liefert wahr, wenn F1 z.B. 10,1 ist.
Liefert falsch, wenn F1 = 9,5 und auch falsch, wenn F1 = "Text"; liefert jedoch wahr, wenn F1 = "3 bis 4 Wochen", da die Ziffern 3 und 4 die Zahl 34 ergeben.

Liste der möglichen Vergleichsoperatoren

Schreibweise	Bedeutung
if (A =? B)	wenn A textgleich B
if (A #? B)	wenn A textlich ungleich B
if (A i=? B)	wenn A textlich identisch B
if (A i#? B)	wenn A textlich nicht identisch B
if (A n=? B)	wenn A numerisch gleich B
if (A n#? B)	wenn A numerisch ungleich B
if (A n>? B)	wenn A numerisch größer B
if (A n<? B)	wenn A numerisch kleiner B
if (A n>=? B)	wenn A numerisch größer gleich B
if (A n<=? B)	wenn A numerisch kleiner gleich B

j) Scriptformat – weitere Regeln

Ein Importscript in DATAform ist ein einfacher Text, der auch direkt eingetippt oder als Textdatei in einem Texteditor geschrieben werden kann.
Am schnellsten klickt man die Felder an und tippt die Sonderzeichen direkt ein:
8 Tabelle == F9 - F10 - F11
oder kurz, da Leerzeichen und Feldnamen auch fehlen dürfen:
8==F9-F10-F11

Die Namen der Elemente nach 15 stehen zwischen Zollzeichen und dürfen nicht entfallen:
15 "•Z Bild" == F2
15 == F2 liefert hingegen eine Fehlermeldung. DATAform benötigt die Namen, um die richtigen Elemente zu finden.

Das Script wird beim Verlassen des Dialogs automatisch gesichert, auch wenn man keinen Import durchführt und auf Zurück klickt. Es wird in der Datendatei gespeichert und steht allen Netzwerkteilnehmern unmittelbar zur Verfügung.
Will man verschiedene Scripte verwenden, so kann man diese mit den Befehlen im Script-Klappmenü als Datei sichern, bei Bedarf auch in einem anderen Programm editieren und wieder laden.

Fehlermeldungen

Beim Klick auf Import prüft DATAform das Script, meldet gefundene Fehler und führt den Import im Fehlerfalle nicht durch.
Die Fehlermeldungen geben eine selbst-verständliche Auskunft, z.B. bei:

8 Tabelle == "Text F2 F3
Ein Zollzeichen " fehlt in Zeile: 1 ("Text ist nicht mit " abgeschlossen)

Oder bei:
8 Tabelle == "Text" F23 F F3
Eine Feldnummer fehlt nach F in Zeile: 1 (Nach einem F kommt keine Zahl)

Kommentare

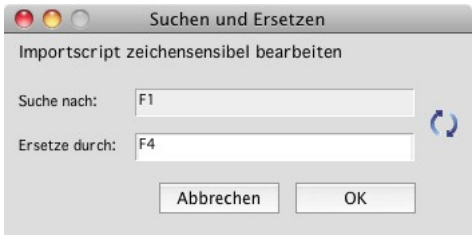
* Zeilen wie diese mit einem Asterisk am Anfang sind Kommentare, ebenso alle Zeilen, die kein doppeltes Ist-gleich-Zeichen == enthalten.

Leertasten werden überall, außer in "konstanten Texten" ignoriert und können zur lesbareren Gestaltung eingesetzt werden. (Siehe das Beispielscript „Telefon-Script.txt“, es kann wie es ist in das Scriptfeld geladen und verwendet werden.)

Behandlung von Duplikaten
Alle Duplikat-Optionen stehen zur Verfügung; die Duplikatfelder müssen gefüllt werden, z.B.:
1 N° == F1
Die Artikelnummer wird hier mit Feld 1 gefüllt; anhand der N° (»1) können nun Duplikate übergangen, überschrieben oder aktualisiert werden. (S. das eigene Kapitel „Duplikatebehandlung“ im Handbuch.)
Beim Überschreiben gilt: Die Vorlage selbst kann nicht überschrieben werden.

S & E im Script
Auch ein Importscript kann per S&E abgeändert werden. Im Scriptklappmenü gibt es den Befehl:

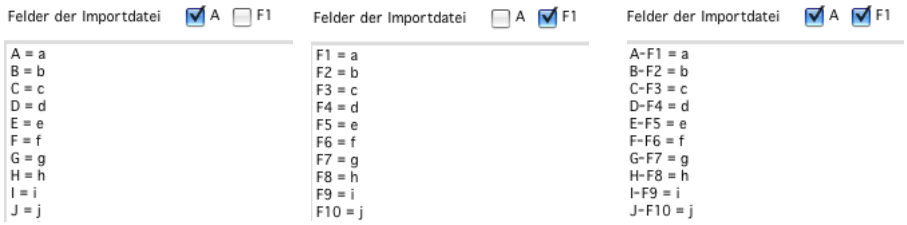
Der Befehl liefert diesen Dialog:



Mit den blauen Wechselpfeilen rechts lassen sich die Feldinhalte tauschen.

Schriftgröße ändern
Die Schriftgröße des Importscripts kann per Befehl-+ und Befehl-Minus verändert werden.

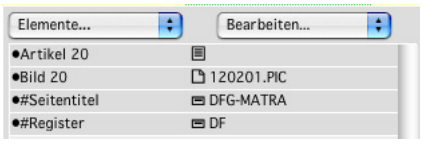
Feldbeschriftung A oder F1
Die beiden Ankreuzfelder A und F1 erlauben den Wechsel der Beschriftungen.
A beschriftet die Zeilen alphabetisch, wie in einer Tabellenkalkulation von A bis Z usw.;
F1 numerisch;
A und F1 tut beides:



Die Abbildung zeigt die 3 Möglichkeiten der Feldbeschriftungen der Importdatei.

Ein Script für viele Varianten
Diese eigene Art an Elementen macht es möglich, beim Scriptimport auch verschiedene Artikelelemente anzusprechen. Je nach Import-Datensatz lassen sich damit andere Artikelrahmen im Zielartikel erzeugen. Mit einem einzigen Importschritt können damit ganz verschiedene Artikelmodule aufgebaut werden.

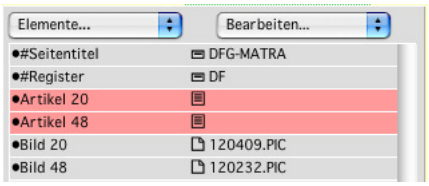
Ein Katalog besitzt beispielweise zwei verschiedene Gestaltungsvarianten, für die zwei verschiedene Musterartikel angelegt wurden:



Das linke Muster verwendet als Standardartikelrahmen das Element „•Artikel 20“, das rechte das Element „•Artikel 48“. Um schon beim Scriptimport diese verschiedenen Artikelrahmen zu erzeugen, mußte man bisher die jeweiligen Artikel in getrennten Schritten importieren und jeweils einen anderen Vorlagenartikel verwenden.
Jetzt kann man diese verschiedenen Artikelelemente in einer einzigen „Super-Vorlage“ zusammenfassen und alles in einem Schritt importieren.

Super-Vorlage erzeugen
Eine „Super-Vorlage“ ist ein normaler Vorlagenartikel, der quasi mehrere Artikelvorlagen inklusive verschiedener Standardartikelrahmen enthält.

- Man legt erst die verschiedenen Artikel an, die die gewünschten Gestaltungsvarianten repräsentieren, Artikel A, B und C etc. Und legt alle Elemente als Muster ab.
- In einem neuen Artikel X, der Super-Vorlage, fügt man dann alle Elemente ein, die man im Importscript ansprechen möchte, alle Textrahmen, Bilder etc. Dieser Artikel ist dann auch die Artikelvorlage für den Scriptimport.
- Alle Artikelrahmen fügt man dabei ein, indem man die *Hochtaste* hält und die Elemente im Klappmenü anwählt. Hochtaste-Einfügen fügt die Elemente als Vorlagenelemente ein:



Die mit Hochtaste eingesetzten Artikel erscheinen rot in der Liste. Es sind Vorlagenelemente, die erst durch einen Scriptimport zu normalen Elementen werden.
In der Abbildung sieht man zwei Standardartikelrahmen in einem Artikel! Das ist nur mit Vorlagenelementen möglich.

Rote Vorlagenelemente haben Nummern, die immer mit 2146 beginnen, gefolgt von weiteren 6 Ziffern:

Sie sind keinem Artikel zugeordnet. Erst nach dem Scriptimport werden sie in den erzeugten Artikeln zu normalen Elementen des Artikels und erhalten auch entsprechende Nummern.

Artikelrahmen importieren
Im Importdialog verwendet man die Super-Vorlage als Vorlage. Im Script muß man dann alle Duplikate an Artikelrahmen löschen, sodass immer nur ein Standardartikelrahmen oder geteilter Artikelrahmen eines Typs übrigbleibt. Ein Script hierfür enthält z.B. die Zeilen:

```
if( F3 n#? "20") 15 "•Artikel 20" ==! ""
```

if(F3 n#? "48") 15 "•Artikel 48" ==! ""

In Abhängigkeit von Feld 3 wird entweder der Artikelrahmen „•Artikel 20“ oder der Artikelrahmen „•Artikel 48“ gelöscht. Nur wenn F3 den Wert 20 enthält bleibt der Artikelrahmen „•Artikel 20“ und nur er erhalten. Und nur wenn F3 den Wert 48 enthält bleibt der Artikelrahmen „•Artikel 48“ und nur er erhalten.

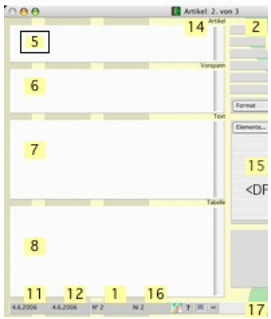
Je nach Importsatz enthält der erzeugte Artikel dann auch verschiedene Standardartikelrahmen.

Felder und Elemente
Unter „Felder und Elemente der Vorlage“, der linken Liste im Scriptimportdialog werden jetzt alle Elemente des Vorlagenartikels angezeigt, um sie per Script löschen zu können - auch geteilte Artikelrahmen und Standardartikelrahmen:

```
15 "•#Seitentitel"
15 "•#Register"
15 "•Artikel 20"
15 "•Artikel 48"
15 "•Bild 20"
15 "•Bild 48"
15 "▨ Linie"
```

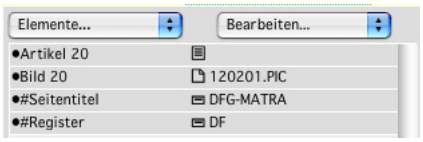
Ein Klick auf eine Zeile erzeugt z.B. 15 "•Artikel 20" == und man ergänzt den Ausdruck zu 15 "•Artikel 20" ==! ""
==! "" im obigen Beispiel bedeutet, dass das Element gelöscht werden soll, wenn die Bedingung if(F3 n#? "48") zutrifft, siehe oben.

In einen Standardartikelrahmen, wie auch in eine Linie, kann kein Feldinhalt importiert werden. Artikelrahmen und Linien werden nur deshalb in der Liste angezeigt, um durch Löschen der nicht benötigten Elemente in einem Vorlageartikel die gewünschten auszuwählen.



Um Text in einen Standardartikelrahmen oder einen geteilten Artikelrahmen zu importieren importiert man die Texte einfach direkt in die dazugehörigen Felder 5 bis 8. Z.B. durch Scriptzeilen wie:
5 Vorspann == F22
8 Tabelle == "Text" F2 F3
Standardartikelrahmen und geteilte Artikelrahmen fassen lediglich die Texte in Rahmen zusammen, die in den Haupttextfeldern 5 bis 8 stehen.

Nach dem Scriptimport erhält man Artikelvarianten wie diese:



Die Artikel der Abbildung besitzen verschiedene Standardartikelrahmen (Artikel 20, Artikel 48) und wurden dennoch durch einen einzigen Scriptimport-Schritt erzeugt.

Vorlagenelemente bearbeiten
Es gibt mehrere Möglichkeiten:
- So geht's: Da Vorlagenelemente immer einen Musterbezug besitzen, kann man die Muster selbst einfach durch den Befehl „Muster exportieren“ in QuarkXPress plazieren, ändern und zurückschreiben. Das ist auch mit roten Vorlagenelementen möglich.
- So geht es nicht: Rote Vorlagenelemente kann man nicht selbst in QuarkXPress ändern. Man könnte sie zwar in QuarkXPress plazieren, aber die Änderungen nicht rückaktualisieren. Rote Vorlagenelemente sind keinem Artikel zugeordnet und werden bei einem Rück-

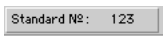
import übergangen.
- So geht es auch: Man ändert die ursprünglichen Muster in QuarkXPress, liest die Änderungen zurück und legt die Elemente wieder als Muster ab; die alten, gleichnamigen Muster werden überschrieben.

Funktionen der Fußleiste

Hilfe
Eine Online-Anleitung erhalten Sie durch einen Klick auf das linke ? in der Fußzeile.

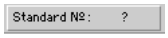
Scriptimport-Standardvorlage
Für den Scriptimport läßt sich ein Standard-Artikel festlegen, der automatisch als Vorlage verwendet werden soll:

Die Schaltfläche bietet drei Funktionen:
- Standard-Artikel festlegen:
  Klicken Sie auf die Schaltfläche. Die Artikelnr. der Vorlage wird angezeigt:

  Der aktuell als Vorlage geladenen Artikel gilt damit – für diese Datendatei und diesen Arbeitsplatz – als Standard-Importvorlage. Der Artikel Nº xy wird automatisch geladen, wenn Sie den Importdialog erneut öffnen.
  (Vor dem Aufruf des Importdialogs mußte man bisher immer erst den Vorlageartikel suchchen oder markieren. Die Vorlage wird jetzt automatisch geladen.)

- Keinen Standard-Artikel laden:
  Klicken Sie wieder auf die Schaltfläche. Statt der Artikelnr. wird ? angezeigt:

  Damit ist das automatische Laden einer Standard-Vorlage abgeschaltet. Wenn Sie den Importdialog erneut öffnen gilt wie bisher der zuletzt aktuelle Artikel als Vorlage und kann als (neuer) Standard festgelegt werden.

- Ergebnis anzeigen
  Zeigt die Schaltfläche nach dem Öffnen des Importdialogs eine Zahl an, wie

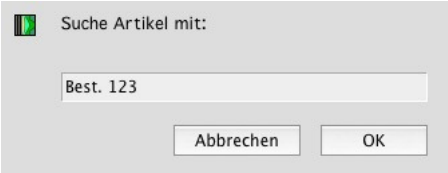
  so bedeutet das: Der früher als Vorlage festgelegte Artikel wurde gefunden.
  Zeigt die Schaltfläche jedoch ein Fragezeichen, so bedeutet das: Es wurde entweder keine Vorlage festgelegt oder der fragliche Artikel wurde zwischenzeitlich gelöscht.

Importdatei durchblättern
Mit den Pfeilen unter der mittleren Liste kann man die Importdatei satzweise durchblättern.
Der Import selbst ist unabhängig davon, welcher Satz hier gerade dargestellt wird.
Die Zahl links neben den Pfeilen bezeichnet die Anzahl der eingelesenen Zeichen.
Die Zahl rechts neben den Pfeilen die fortlaufende Nummer des aktuellen Satzes.

Blätternfunktion, Hochtaste-Blättern

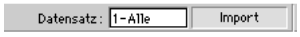
Hält man die Hochtaste gedrückt während man blättert, springt DATAform in der geöffneten Importdatei um hundert Sätze weiter bzw. wieder zurück.

Importdatei durchsuchen
Blätternfunktion, Wahl/Alt-Blättern
Hält man die Wahl/Alt-Taste beim Blättern gedrückt erscheint der Dailog:



DATAform durchsucht daraufhin die für den Import geöffnete Datei nach dem Suchwort und zeigt den gefundenen Satz an. Ein erneuter Aufruf sucht den nächsten Satz und so weiter. Die Funktion ist vorwärts und rückwärts aufrufbar.

Textimport einzelner Datensätze
Im Feld „Datensatz“, in der Fußleiste rechts, kann man angeben welche Folge an Datensätzen der Datei man importieren möchte.



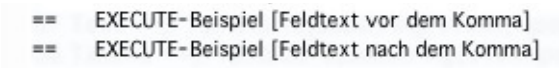
Im Feld steht zunächst: „1-Alle“ => Alle Datensätze werden importiert.
Möglich sind Angaben wie:

„4-Alle“ oder „4-“	Alle Datensätze ab dem vierten werden importiert.
„2-12“	Nur die Datensätze 2 bis 12 werden importiert.
„9-9“ oder „9“	Nur der neunte Satz wird importiert.
„1-4“ oder „-4“	Die ersten vier Datensätze werden importiert.

Das Zeichen „-“ ist das Minuszeichen.

Scriptimport mit EXECUTE
Will man per Scriptimport nur Teile eines Importfeldes importieren oder den Text gleich beim Import verändern etc., bietet der EXECUTE-Befehl ein mächtiges Werkzeug. Der Befehl liefert viele Möglichkeiten, erfordert aber auch ein gewisses Programmierverständnis.

Im Scriptimport-Klappmenü findet man hierfür zwei neue Befehle:



Die zwei Befehle fügen diese Beispiele ein:
07 Text ==[F1;EXECUTE;Substring(vText;1;Position(",";vText)-1)]
08 Tabelle ==[F1;EXECUTE;Substring(vText;Position(",";vText)+1)]
Die Beispiele werden jetzt ausführlich erläutert.

07 Text	bezeichnet wie üblich ein DATAform-Feld, in das importiert werden soll.
F1	das Feld, das importiert werden soll. Das Beipiel erzeugt immer F1. Sie müssen die Feldnummer Ihrem Import anpassen, also z.B. in F37 abändern.
EXECUTE	das 2. Argument lautet konstant EXECUTE.
Substring...	im 3. Argument steht die Formel, nach der der importierte Text bearbeitet werden soll. Die Möglichkeiten für diese Formel sind weitreichend und umfassen alle hier adäquaten Textoperationen von 4D.

Das Beispiel verwendet diese Formel: Substring(vText;1;Position(",";vText)-1)

Substring()	liefert einen Teilstring (Teil-Zeichenkette) aus vText.

Dazu zunächst ein einfacheres Beispiel: Substring(vText;1;3)

vText	lautet konstant vText. In dieser Variable steht der gerade zu importierende gesamte Text. Von diesem Text soll nun ein Teil extrahiert werden.
1	1 bedeutet, ab dem ersten Zeichen soll der Text übernommen werden
3	3 bedeutet es sollen 3 Zeichen gelesen werden.

Will man beispielweise die ersten 3 Zeichen aus F1 importieren, so lautet die komplette Formel einfach: 08 Tabelle ==[F1;EXECUTE;Substring(vText;1;3)]

Unser Beispiel verwendet weiterhin die Funtkion Position(",";vText)

Position	die Funktion sucht das 1. Argument in vText
","	ein Komma wird in vText gesucht
vText	ist der Text der Zelle, die gerade importiert wird.

Wird das Komma gefunden, so liefert die Funktion die Position, an der das Komma gefunden wurde. Ist vText beispielweise „12,95“, so liefert Position(",";vText) den Wert 3.

Nun das ganze Beispiel Substring(vText;1;Position(",";vText)-1) für 12,95.
Position(",";vText) liefert für 12,95 den Wert 3. Wir können also stattdessen 3 einsetzen: Substring(vText;1;3-1) ergibt Substring(vText;1;2), es werden also 2 Zeichen ab Zeichen 1 gelesen. Der ganze Ausdruck Substring(vText;1;Position(",";vText)-1) sucht also nach dem Komma und liest alle Zeichen bis vor dem Komma, im Falle von 12,95 liefert der Ausdruck also 12.

Der EXECUTE-Befehl erlaubt weitere Funktionen, z.B.:
08 Tabelle ==[F1;EXECUTE;Replace string(vText;"a";"bb")]
Ersetzt beim Import alle "a" durch "bb".

Einen Beschreibung der 4D-String-Funktionen finden Sie hier:
http://doc.4d.com/4D-Programmiersprache-13.2/String.201-1076659.de.html