

GASSENHUBER
Systementwicklung
www.gassenhuber.de

Anwenderhandbuch DATAform- Datenbank 6.6



Datenbankgestütztes Publizieren
Database-publishing mit QuarkXPress



 **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 Programm-mängeln 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 of limitation of incidental or consequential damages is not allowed in some areas, so these exclusion and limitations may not apply to you.

Stand: April 2004

DATAform CD [61]

DATAform-Datenbank Version 6.6.3 (80)

DATAform-XTension Version 4 bis 6.1
für QuarkXPress Version 4 bis 6.1

Für MacOS, MacOS X, Windows

Die DATAform-Datenbank, das DATAformXTension und DATAformMarken sind Produkte von

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.

Alle genannten Warenzeichen wie QuarkXPress, XTensions, QuarkXTension, PageMaker, 4th Dimension, 4D Server, 4D Client, 4D Write, 4D Tools, 4D BackUp, 4D Remote, 4D External, Apple, Macintosh, MacOS, Power-Macintosh, Windows / NT, PhotoShop, Explorer, Netscape etc. sind Eigentum der jeweiligen Inhaber.

VI. Themenbereiche

1. Schnittstellen

Dieses Kapitel beschreibt alle DATAform-Schnittstellen im Zusammenhang, nennt praktische Anwendungen und erläutert spezielle Bedingungen im plattformübergreifenden Einsatz.

- 1.1 DATAform-Datensätze (DFX)
- 1.2 Scriptimport
- 1.3 Scriptexport
- 1.4 Textexport per 4D-Bericht
- 1.5 DATAform-Textformat
- 1.6 Austausch mit QuarkXPress
- 1.7 Import von S&E-Tabellen
- 1.8 Automatischer Import von Bildern

1.1 DATAform-Datensätze (DFX)

Die Schnittstelle dient dem Austausch von kompletten Artikeln zwischen verschiedenen DATAform-Datenbanken oder zu Archivierungszwecken. Die Artikel werden, mit oder ohne Miniaturen, in einem eigenen Format mit der Endung .DFX exportiert und können nur wieder in eine DATAform-Datenbank importiert werden.

Will man Artikel zwischen DATAform-Datenbanken austauschen, sollte man immer das DFX-Format wählen: Dateien in diesem Format sind plattformunabhängig und können ohne weitere Einstellungen in jede DATAform-Datenbank unter MacOS oder Windows importiert oder aus ihr exportiert werden.

DATAform-Datensätze exportieren

Will man einen einzelnen Artikel exportieren, so markiert man ihn in der Artikelliste oder öffnet ihn. Will man mehrere Artikel exportieren, so sucht man sie per Suchbefehl und/oder markiert sie in der Liste. Dann wählt man den Befehl DATA/Export. Der Fenstertitel des Exportdialogs zeigt die Anzahl der Artikel, die exportiert werden.

In der Fußleiste befindet sich die Schaltfläche „Datensätze.DFX“:



Das Ankreuzfeld „Mit Miniaturen“ ist voreingestellt AN. In diesem Fall werden die Miniaturbilder der Artikel mit ausgetauscht und müssen nach dem Import nicht wieder neu erzeugt werden. Klickt man auf „Datensätze.DFX“ – oder OK-Taste, wie für alle vertieft dargestellten Felder möglich – so erhält man den Datei-Anlegen-Dialog und kann die Datei sichern.

Wenn Sie die Datei unter Windows erzeugen und/oder unter Windows importieren wollen, müssen Sie die Endung „.DFX“ beibehalten.

(Hintergrund: Unter Windows wird der Dateityp in den drei letzten Buchstaben des Namens kodiert. Exportiert man DATAform-Artikel unter Windows und liest die Datei unter MacOS ein, so erscheint sie als TEXT-Datei mit einem entsprechenden Icon, was soviel bedeutet wie „allgemeine PC-Datei ohne definierten Dateityp“. DATAform überprüft deshalb auch unter MacOS die Namensendung von Textdateien und behandelt DFX-Textdateien als Dateien mit DATAform-Datensätzen. Unter Windows exportierte DATAform-Datensätze werden deshalb automatisch auch unter MacOS als solche erkannt und können ohne weiteres importiert werden.)

DATAform-Datensätze importieren

Wählen Sie den Menübefehl DATA/Import, der Datei-Öffnen-Dialog erscheint.

Öffnen Sie eine Datei mit DATAform-Datensätzen. Sie erkennen eine derartige Datei an der Endung „.DFX“/„.dfx“ oder unter MacOS auch an dem Icon:



Der Importdialog für DATAform-Datensätze erscheint:



Der Fenstertitel zeigt den Namen der geöffneten Datei. Im oberen Feld des Dialogs sieht man den kompletten Pfad zur Datei.

Nach dem Import löschen

Ist das Ankreuzfeld AN, so erscheint nach dem erfolgreichen Import der Dialog:



Klickt man auf OK, so wird die eben importierte Datei gelöscht.

Duplikatebehandlung

Im unteren Feld des Dialogs läßt sich definieren, wie DATAform Artikel behandeln soll, die bereits in der Datenbank vorhanden sind. Dieselben Duplikat-Einstellungen sind auch beim Import von Textdateien vorhanden und werden im eigenen Abschnitt „Duplikatebehandlung“ weiter unten, Seite: 192f., beschrieben.

Nach dem Import erscheinen die importierten Artikel in der Artikelliste.

1.2 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 Importsript.
- Import und Export von Textdateien im DATAform-Textformat. Ein Textimport über diesen Weg erfordert meist eine Anpassung der exportierenden Datenbank. Dieser Weg bietet mehr Möglichkeiten, erfordert aber weitergehende Kenntnisse und mehr Vorarbeiten. Siehe 1.5 Austausch im DATAform-Textformat.

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 Importsript definiert.

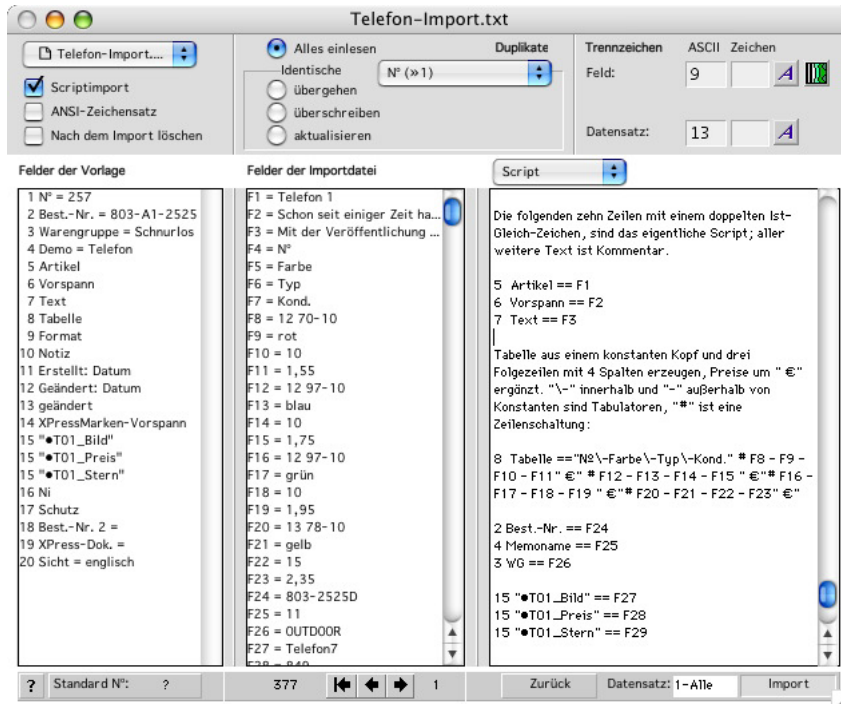
Das Importsript erlaubt zudem die Summierung von importierten Feldern in einzelnen DATAform-Feldern, die zusätzliche Zuweisung von konstanten Texten und vieles mehr.

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. Alle importierten neuen Artikel besitzen am Ende dieselbe Anzahl an Elementen.

Anwendungsbeispiel

- Kopieren Sie sich den Ordner DATAform-Datenbank/DATAform-Demos/DATAform-Script-Demo von der DATAform-CD.
- 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-Klappenmenü 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 Importskript in DATAform ist ein Text, der eingetippt oder durch einfaches Klicken geschrieben werden kann.

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*

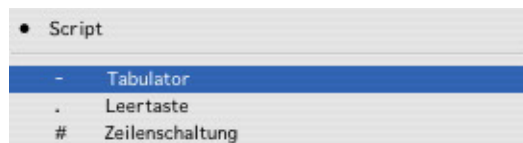
Löschen Sie das alte Script im rechten Feld. Klicken Sie links z.B. auf „8 Tabelle“, im Scriptfeld erscheint: 8 Tabelle ==

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.

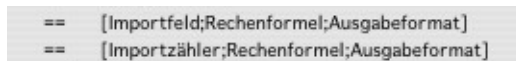
Ein neues Feld wird durch einen Klick in die mittlere Liste immer dort eingefügt, wo die Schreibmarke steht. Ist Text im Scriptfeld markiert, wird er dabei überschrieben.

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 bedeutet:

- Das Importfeld ist F1, das erste Feld der Importdatei.

- Die Rechenformel 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 beispielsweise 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 €

- 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 bedeutet:

- N# ist eine Variable, die die Anzahl der gelesenen Sätze der Importdatei während des Importvorgangs wiedergibt. Sie entspricht der Zahl die rechts neben den Blätternpfilen im Importsript-Dialog angezeigt wird.
- Beim ersten gelesenen Satz ist N#=1; wird mit 50 multipliziert => 50; wird auf 6 Nullen aufgefüllt => als Ergebnis wird 000050 in das Feld vor == eingetragen.
- Beim 80.ten gelesenen Satz ist N#=80; wird mit 50 multipliziert => 4000; wird auf 6 Nullen aufgefüllt => als Ergebnis wird 004000 in das Feld vor == eingetragen.

Zusammenfassendes Beispiel:

Die Scriptzeile:

```
8 Tabelle == F1 " Original"[F1;*2;] " doppelt" # [F1;;####,00 €]" Original + €" #
"Importsatz Nr.: "[N#;;] # "Sortiernummer aus N#:"[N#;*50;0000000]" # "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

e) *Leere Elemente löschen*

```
==! 15 "Element" löschen wenn leer
```

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 Elementfelder 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 leer ist

15 "Bild 05" ==! F5000

Gibt es keine 5000 Felder in der Importdatei, wird das Element immer gelöscht.

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.

f) *Feldinhalt hinzufügen*

```
==+ 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.

g) *Konstanten verwenden*

```
" " Textkonstante
\ - "Tabulator"
\ # "Zeilenschaltung"
\ " "Zollzeichen"
```

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 „ €“, so daß 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 XPressMarken/|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-Klappenmenü zeigt diese Zeilen:

```
15 "Element" => Text/Bildname
15.04 "Element" => XPressMarken/|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 XPressMarken 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 Bedingungen

Den Zeilen in einem Importsript 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!

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 Importsript 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" ==; 15 == liefert 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.

Ein Script kann bis zu 32000 Zeichen enthalten.

Will man verschiedene Scripte verwenden, so kann man diese mit den Befehlen im Script-Klappenmenü als Datei sichern, evtl. 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 Fehlerfall nicht durch.

Die Fehlermeldungen geben eine selbst-verständliche Auskunft, z.B. bei:

8 Tabelle == "Text F2 F3

Ein Zollzeichen " fehlt in Zeile: 1 ("Text wird 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 Asterix 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 lesbaren Gestaltung eingesetzt werden.

(S. das Beipielscript „Telefon-Script.txt“, es kann wie es ist in das Scriptfeld geladen und verwendet werden.)

ANSI-Zeichensatz

ANSI-Zeichensatz

Eine explizite Konvertierung aus dem ANSI-Format ist nur nötig bei einem Textimport von Windows-Textdateien, die von einem Windows-Programm stammen. In diesem Fall stellt man das Ankreuzfeld AN. (Mehr hierzu unter „DATAform-Textdatei“ weiter unten.)

Behandlung von Duplikaten

Alle Duplikat-Optionen stehen zur Verfügung; die Duplikatfelder müssen gefüllt werden, z.B.: 1 N^o == F1

Die Artikelnummer wird hier mit Feld 1 gefüllt; anhand der N^o (»1) können nun Duplikate übergangen, überschrieben oder aktualisiert werden. (S. das eigene Kapitel „Duplikatebehandlung“ weiter unten.)

Beim Überschreiben gilt: Die Vorlage selbst kann nicht überschrieben werden.

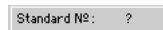
Weitere Einstellungsmöglichkeiten, wie der Bereich „Trennzeichen“, gleichen denen für DATAform-Textdateien und werden dort erläutert.

Funktionen der FußleisteHilfe

Eine Online-Anleitung erhalten Sie durch einen Klick auf das Fragezeichen 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:

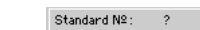


Der aktuell als Vorlage geladenen Artikel gilt damit – für diese Datendatei und diesen Arbeitsplatz – als Standard-Importvorlage. Der Artikel N° xy wird automatisch geladen, wenn Sie den Importdialog erneut öffnen.

(Vor dem Aufruf des Importdialogs mußte man bisher immer erst den Vorlageartikel suchen oder markieren. Die Vorlage wird jetzt automatisch geladen.)

- Keinen Standard-Artikel laden:

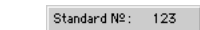
Klicken Sie wieder auf die Schaltfläche. Statt der Artikelnr. wird ? angezeigt:



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 anzeigen

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.

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.

1.3 Scriptexport

Das Modul Scriptexport ermöglicht den Export von Artikeln in einem frei definierbaren Textformat. Alle Artikel können z.B. als HTML- oder XML-Datei exportiert werden. Der Scriptexport erlaubt die freie Definition der Feldreihenfolge und ermöglicht den Zugriff auf einzelne Zellen innerhalb von Texten. Das Scriptexport-Modul wird ab Seite: 242 beschrieben.

1.4 Textexport per 4D-Bericht

Die Schnittstelle ermöglicht den einfachen Export einzelner Felder als Text für QuarkXPress, Textverarbeitungs- oder Tabellenkalkulationsprogramme. Diese Textdateien können nicht wieder in DATAform importiert werden.

Der 4D-Schnellbericht erlaubt nicht den gezielten Zugriff auf einzelne Elemente der Elementeliste oder Zellen in einem Text. In diesen Fällen muß man auf den Scriptexport zurückgreifen, siehe oben.

Anwendungsbeispiel

Im Exportdialog gibt man als Feldtrennzeichen 9 (Tabulator) und als Datensatztrenner 13 (Zeilenschaltung) an, indem man z.B. die Zahlen in die Spalte „ASCII“ einträgt:

| Trennzeichen | ASCII | Zeichen |
|---------------|-------|---------|
| Feld: | 9 | |
| Elementefeld: | 29 | |
| Datensatz: | 13 | |

Der mittlere Elementefeldtrenner wird in diesem Format nicht verwendet.

| | |
|------------------|------------|
| Neuer 4D-Bericht | 4D-Bericht |
|------------------|------------|

Dann wählt man das Schaltfeld „Neuer 4D-Bericht“ in der Fußleiste des Dialogs – oder „4D-Bericht“, wenn man ein schon einmal erstelltes 4D-Schnellbericht-Dokument öffnen will.

Im 4D-Berichteditor zieht man nun die gewünschten Felder mit der Maus in den unteren Bereich. Um die Artikel zu sortieren, zieht man das gewünschte Feld in den Bereich „Sortierfolge“. Der 4D-Schnellbericht sieht dann z.B. so aus:

Mit dem Menübefehl Ablage/Sichern (Windows: Datei/Speichern) oder Sichern unter... kann man die getroffenen Einstellungen in einer Datei auf der Festplatte sichern und später wieder verwenden. Dadurch werden nur die Einstellungen gespeichert, nicht die Artikel exportiert.

Zum Exportieren der Artikel als Text wählt man unter Ablage(Datei)/Drucken auf... als Ausgabeziel die Option „Festplatte“:

| | |
|--------------------|--------------|
| Drucken auf | Drucker |
| Kopf- & Fußteil... | ✓ Festplatte |
| Papierformat | Diagramm |

Dann wählt man den Befehl Ablage(Datei)/Drucken. Der Datei-Anlegen-Dialog erscheint. Man trägt einen Dateinamen ein und sichert die Datei.

Es wird in diesem Beispiel eine Tab-Zeilenschaltung-Liste mit den drei Feldern erzeugt, die z.B. von einem Tabellenkalkulationsprogramm richtig eingelesen werden kann, sofern in den exportierten Feldern keine Zeilenschaltungen und Tabulatoren enthalten sind.

1.5 DATAform-Textformat

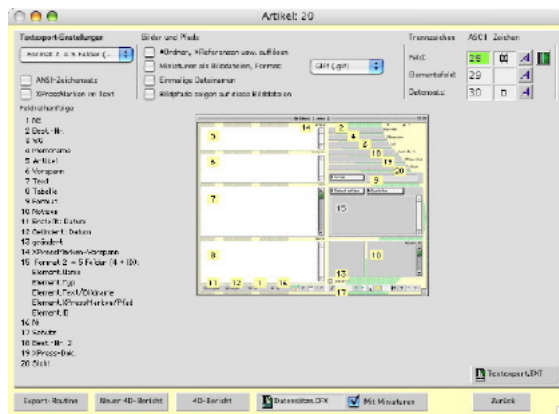
Die Schnittstelle dient dem Austausch von DATAform-Artikeln mit anderen Programmen. Über den Textimport lassen sich vollständige und individuelle DATAform-Artikel importieren, inklusive Formatzuweisung, Bildpfaden und Zuordnung von Musterrahmen.

Ein Import im DATAform-Textformat aus einer anderen Datenbank erfordert mehr Vorarbeit als ein Scriptimport. Das Format bietet aber im Gegenzug die vollständige Kontrolle über die importierten Artikel, v.a.:

- für jeden Artikel können eigene, unterschiedliche Elementlisten importiert werden. Beim Scriptimport ist das nur bedingt über das Löschen von Elementen möglich.
- alle Elementfelder können individuell gefüllt werden, beim Scriptimport nur drei Felder.

Artikel als DATAform-Textdatei exportieren

Der Menübefehl DATA/Export liefert den Dialog:

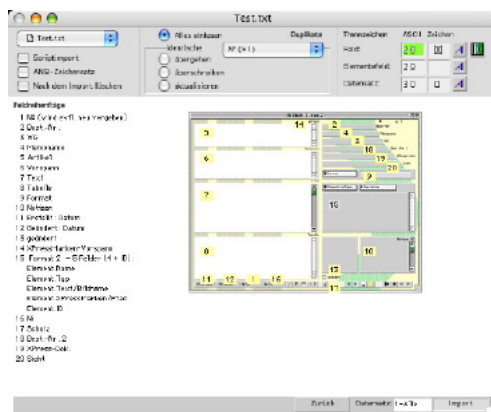


Der Fenstertitel des Exportdialogs zeigt die Anzahl der Artikel, die exportiert werden sollen. Der gesamte obere, grau hinterlegte Bereich dient dem Textexport, mit der dazugehörigen Schaltfläche „Textexport.TXT“ rechts unten. Klickt man auf „Textexport.TXT“, so erhält man den Datei-Anlegen-Dialog. Wenn Sie die Datei unter Windows erzeugen und/oder unter Windows importieren wollen, muß die Endung der erzeugten Datei „.TXT“ lauten. Sichern Sie die Datei.

DATAform-Textdatei importieren

Wählen Sie den Menübefehl DATA/Import, der Datei-Öffnen-Dialog erscheint.

Öffnen Sie die eben gesicherte Textdatei. Der Importdialog für eine Textdatei erscheint:



Stellen Sie das Ankreuzfeld „Scriptimport“ falls nötig auf AUS.

Der Fenstertitel zeigt den Namen der geöffneten Datei. Das Klappenmenü „Textdatei“ zeigt den kompletten Pfad zur Datei.

Die Funktionen der beiden Dialoge decken sich zum Teil und werden im folgenden fortlaufend erläutert.

a) *Format*

Das DATAform-Textformat definiert eine feste Feldreihenfolge aus 20 Feldern. Die Felder müssen in der Importdatei in dieser Ordnung vorliegen:

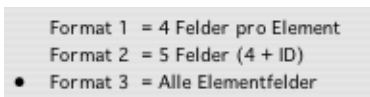
Feldreihenfolge

- 1 Nr. (wird evtl. neu vergeben)
- 2 Best.-Nr.
- 3 WG
- 4 Memoname
- 5 Artikel
- 6 Vorspann
- 7 Text
- 8 Tabelle
- 9 Format
- 10 Notizen
- 11 Erstellt: Datum
- 12 Geändert: Datum
- 13 geändert
- 14 XPressMarken-Vorspann
- 15 Format 2 = 5 Felder (4 + ID):
 - Element.Name
 - Element.Typ
 - Element.Text/Bildname
 - Element.XPressMarken/Pfad
 - Element.ID
- 16 Ni
- 17 Schutz
- 18 Best.-Nr. 2
- 19 XPress-Dok.
- 20 Sicht



Rechts sieht man die DATAform-Eingabemaske mit numerierten Feldern. In dieser Reihenfolge werden die Feldinhalte exportiert und importiert. Links daneben sieht man die aktuellen Namen der Felder.

Das Feld 15 - die Elementeliste - enthält seinerseits eine Liste aus jeweils 4–12 Feldern pro Element. Beim Export hat man die Wahl zwischen drei verschiedenen Formaten. Man wählt ein Format aus dem Klappmenü:

Format 1

Im Format 1 werden pro Element, d.h. pro Zeile in der Elementeliste, 4 Felder exportiert:

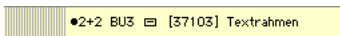
15 Format 1 = 4 Felder pro Element:
 Element.Name
 Element.Typ
 Element.Text/Bildname
 Element.XPressMarken/Pfad

Format 2

Im Format 2 wird zusätzlich zu den vier Feldern die ID, d.h. die Rahmenkennung des Elements, exportiert und importiert:

15 Format 2 = 5 Felder (4 + ID):
 Element.Name
 Element.Typ
 Element.Text/Bildname
 Element.XPressMarken/Pfad
 Element.ID

Die Rahmenkennung eines Elements sieht man im Kopf der Elementdialoge:



Dieses Element hat die ID 37103; in QuarkXPress wird ihm diese Nummer als Rahmenkennung/Objektnummer zugewiesen werden.

Format 3

Im Format 3 werden alle zwölf Felder, die ein Element beschreiben, exportiert und importiert:

15 Format 3 = Alle Elementfelder:
 Element.Name
 Element.Typ
 Element.Text/Bildname
 Element.XPressMarken/Pfad
 Element.ID
 Element.DATaformMarken
 Element.Links .Oben .Breite .Höhe
 Element.Layoutzeile
 Element.Position

Welches Format soll man wann verwenden?Format 1 (4 Felder pro Element)

Das Format 1 kann, auch ohne Vorspann, verwendet werden, wenn

- man durchgängig mit Musterrahmen arbeitet (Die Textdatei spricht Musterrahmen an, enthält aber selbst keine Rahmenbeschreibungen) und
- die ID/Rahmenkennungen der Elemente beim Import verändert werden dürfen. (Die Elemente werden in diesem Format in der Reihenfolge des Imports neu nummeriert.)

Format 2 (5 Felder, 4 + ID)

Müssen die Rahmenkennungen beim Import erhalten bleiben, so muß man Format 2 oder Format 3 verwenden. Dies ist z.B. bei folgendem Ablauf der Fall:

- Man übergibt Artikel an QuarkXPress und gestaltet die Seiten.
- Dann exportiert man die Artikel als Text, bearbeitet sie und reimportiert sie in DATAform an Stelle der ursprünglichen Artikel durch Überschreiben nach N^o.
- Schließlich exportiert man sie nach QuarkXPress und aktualisiert dort die bereits gesetzten Artikel.

Im Format 1 kann es in diesem Ablauf vorkommen, daß Elemente eine andere ID erhalten und die Artikel in QuarkXPress deshalb nicht mehr aktualisiert werden können. In den Formaten 2 und 3 ist die Aktualisierbarkeit auch bei diesem Ablauf sichergestellt.

Das Format 2 hat darüberhinaus keine Nachteile gegenüber Format 1 und kann Format 1 in jedem Fall ersetzen. Es ist in DATAform voreingestellt.

Format 3 (Alle Elementfelder)

Alle Elementeigenschaften mit Positionen und DATAformMarken werden exportiert und importiert.

Das Format muß gewählt werden, wenn man individuelle Elemente ohne Musterbezug als Text exportieren und importieren will. Alle Eigenschaften bleiben erhalten und können z.B. von einer anderen Datenbank weiter bearbeitet werden. Das Format 3 ist ein komplettes Austauschformat.

Trennzeichen im Format 3

Die in diesem Format exportierten DATAformMarken enthalten selbst einige Sonderzeichen wie „| ± » « § [] Δ *“. Diese Zeichen und andere sind daher nicht als Trennzeichen geeignet. Sicher fährt man immer mit den Standardtrennzeichen 28, 29, 30:

| Trennzeichen | ASCII | Zeichen |
|--------------|-------|---------|
| Feld: | 28 | ␣ |
| Elementfeld: | 29 | ␣ |
| Datensatz: | 30 | ␣ |

Vorspann einer DATAform-Textdatei

Alle drei Formate verwenden einen Vorspann, in dem die Formatart, die Trennzeichen und der verwendete Zeichensatz abgelegt werden: Importiert man eine DATAform-Textdatei,

| Trennzeichen | ASCII | Zeichen |
|--------------|-------|---------|
| Feld: | 162 | ␣ |
| Elementfeld: | 169 | ␣ |
| Datensatz: | 163 | ␣ |

- so werden die verwendeten Trennzeichen wieder eingestellt:
- das Ankreuzfeld „ANSI-Zeichensatz“ wird auf AN oder AUS gestellt.
- die Felderliste im Importdialog zeigt das verwendete Format an.

Der Vorspann einer Textdatei ist so aufgebaut, Beispiel:

FORMAT=2 FD=␣ ED=© RD=£ CS=0 *_*_*

FORMAT=2 Format 2, d.h. 5 Felder pro Element

| | |
|-------|---|
| FD=¢ | Feldtrenner ist ¢ |
| ED=© | Elementefeldtrenner ist © |
| RD=£ | Datensatztrenner ist £ |
| CS=0 | Der Zeichensatz ist MacOS; „ANSI-Zeichensatz“ ist AUS. (Durch CS=1 würde das Ankreuzfeld auf AN gestellt.) |
| *_*_* | Ende des Vorspanns (Stern, Minus, Stern, Minus, Stern) |

Fehlt der Vorspann in einer Importdatei, muß die Datei 4 Felder pro Element enthalten;
DATAform liest die Datei als Format 1.

b) Felderliste

| Nr. | Feldname | Feldlänge | Beispiel |
|-------|---|-----------|--------------------|
| 1 | Datensatznummer N ^o | Long | 1253 |
| 2 | Bestellnummer | 25 | 123-456-78 |
| 3 | Warengruppe | 30 | Elektro |
| 4 | Memoname | 25 | Stabilo |
| 5 | Artikel | 32K | Handy |
| 6 | Vorspann | 32K | Das Telefon... |
| 7 | Text | 32K | Lieferbar... |
| 8 | Tabelle | 32K | 123 rot ... |
| 9 | Format | 32K | 3spezial |
| 10 | Notiz | 32K | S94:126 |
| 11 | Erstellt: | 10 | 10.5.1995 |
| 12 | Geändert: | 10 | 12/9/96 |
| 13 | geändert-Ankreuzfeld | 1 | 0 |
| 14 | XPressMarken-Vorspann | 32K | @Text=... |
| 15.01 | Elemente.Name | 21 | ●Z Bild |
| 15.02 | Elemente.Typ | 2 | 27 |
| 15.03 | Elemente.Text/Bildname | 32K | Telefon4 oder »3 |
| 15.04 | Elemente.XPressMarken/Pfad ab Format 2 zusätzlich: | 32K | ●Bildordner |
| 15.05 | Elemente.ID ab Format 3 zusätzlich: | Zahl | 1001 |
| 15.06 | Elemente.DATAformMarken | 32K | @Preis: |
| 15.07 | Elemente.Links | Zahl | 3,15 |
| 15.08 | Elemente.Oben | Zahl | 6,15 |
| 15.09 | Elemente.Breite | Zahl | 100 |
| 15.10 | Elemente.Höhe | Zahl | 200 |
| 15.11 | Elemente.Layoutzeile | Ganzzahl | 0 |
| 15.12 | Elemente.Position | Ganzzahl | 0 |
| 16 | Ni | Long | 1253, ein Original |
| 17 | Schutz | Ganzzahl | 101 |
| 18 | Best.-Nr. 2 | 30 | 123-456-78B |
| 19 | XPress-Dok. | 30 | Kapitel023.qxp |
| 20 | Sicht | 30 | spanien |

Die Gesamtlänge eines Artikels als Text darf 16K nicht überschreiten.

Einige Erläuterungen zu einzelnen Feldern

Feld 1 - Datensatznummer N^o

Die angegebene Datensatznummer ist nur dann von Bedeutung, wenn das Feld als Duplikatfeld verwendet werden soll. In allen anderen Fällen vergibt DATAform beim Import eine neue eindeutige Datensatznummer und ignoriert den Inhalt von Feld 1. Das Trennzeichen ist aber erforderlich.

Feld 2–4 - Suchfelder

Die Felder 2–4 und 18–20 haben jeweils eine Längenbegrenzung, überschüssige Zeichen werden nicht gespeichert.

Felder 5–8 - Texte für die Artikelrahmen

Die Felder 5 bis 8 sind die vier Textfelder für die Artikelrahmen; diese Texte - und die Texte der Elementliste - werden an QuarkXPress übergeben.

Feld 9 - Format

Im Feld 9 kann man einen Formatnamen wie „|3spezial“ übergeben. Das „|“ sagt DATAform, daß es sich um einen Formatnamen handelt - und nicht um eine Formatbeschreibung. Das Zeichen „|“ hat den ASCII-Wert 124. (S. auch unter „Sonderzeichen“ im Anhang)

Feld 11–12 - Datumsangaben

Die beiden Datumsangaben können in verschiedenen Formen wie „15.05.1997“ oder „15/5/97“ etc. übergeben werden. Fehlt das Erstellt-Datum, wird beim Import das Tagesdatum eingesetzt.

Feld 13 - Ankreuzfeld „geändert“

Das Feld für das Ankreuzfeld wird durch „1“ auf angekreuzt gesetzt; durch alle anderen Werte auf nicht-angekreuzt.

Feld 15 - Zeilen in der Elementliste

Pro Element werden 4–12 Elementfelder beschrieben. Jeweils ein Satz an Elementfeldern erzeugt einen Eintrag in der Elementliste. Die Felder einer Elementezeile enthalten:

.01 Name

Name des Elements wie „Mein Rahmen“ oder „●Rahmen 48“, „●*Rahmen 48“.

Das Zeichen „●“ ist unter MacOS das Zeichen 165 und wird durch Wahl-Ü erzeugt.

Unter Windows ist es das Zeichen 149 und wird durch Alt+0149 erzeugt. Beim Import einer Textdatei mit Windows-Zeichentabelle ist dann „ANSI-Format“ auf AN zu stellen.

.02 Rahmentyp

Im Feld 15.02 wird die Nummer des gewünschten Rahmentyps als String übergeben, z.B. „30“ für einen Textrahmen. Der Typ 31 für den Standard-Artikelrahmen darf pro Datensatz nur einmal erzeugt werden. Wird keine dieser Nummern übergeben, wird ein Textrahmen angelegt. (Zu den „geteilten Artikelrahmen“ ab Nummer 229, s. auch den gleichnamigen Themenbereich.) Folgende Rahmentypen sind möglich:

- 23 Bildrahmen markiert
- 24 Textrahmen markiert
- 27 Bildrahmen nicht markiert
- 28 Linie
- 29 Fläche
- 30 Textrahmen nicht markiert
- 31 Standard-Artikelrahmen, Feld 1 + 2 + 3 + 4
- 229 Feld 1 + 2 + 3
- 230 Feld 1 + 2 + 4
- 231 Feld 1 + 3 + 4
- 232 Feld 2 + 3 + 4
- 233 Feld 1 + 2
- 234 Feld 1 + 3
- 235 Feld 1 + 4
- 236 Feld 2 + 3
- 237 Feld 2 + 4
- 238 Feld 3 + 4
- 239 Feld 1
- 240 Feld 2
- 241 Feld 3
- 242 Feld 4

.03 Text/Bildname

Enthält den Text für einen Textrahmen bzw. den Bildnamen für einen Bildrahmen. Für die Artikelrahmen, Flächen und Linien bleibt das Feld leer. Als Bildname kann auch eine Referenz auf ein Feld, in dem der Bildname steht, eingelesen werden, z.B. „»2“.

.04 XPressMarken/Bildpfad

Enthält den XPressMarken-Vorspann für einen Textrahmen/geteilten Artikelrahmen bzw. den Bildpfad für einen Bildrahmen.

Der XPressMarken-Vorspann in diesem Feld wird für alle Textrahmen (Rahmentyp 30) und für alle geteilten Artikelrahmen verwendet, die das Feld 1 nicht enthalten.

Für den Standard-Artikelrahmen sowie für alle geteilten Artikelrahmen, die das Feld 1 enthalten, wird das Feld nicht verwendet; für diese Rahmen gilt der XPressMarken-Vorspann in Feld 14.

Als Ordner-Pfad kann übergeben werden:

- ein MacOS-Pfad der Form „Festplatte:Ordner1:Ordner2:“
- ein Windows-Pfad der Form „C:\dir1\dir2\“ oder \\Server\Platte\dir1\
- ein Ordnernamen wie „●Bilder1000“.

TIP: Exportiert man DATAform-Artikel im Textformat, so zeigt die erzeugte Datei praktisch, wie die Textdatei aufgebaut sein muß.

Beispiel einer Textdatei

Die folgende Textdatei wurde durch den Textexportbefehl erzeugt. Exportiert wurde ein einzelner Artikel im Format 2, in dessen Felder jeweils der Feldname eingegeben wurde. Als Trennzeichen wurden ¢, © und £ verwendet. Die Elemente wurden nachträglich – zur besseren Lesbarkeit – in eigene Zeilen gesetzt, die Elementfelder wurden an Tabulatoren ausgerichtet:

```

FORMAT=2 FD=¢ ED=© RD=£ CS=0 *-.*-*62¢Best.-Nr.¢Warengruppe¢Memoname¢Arti-
kel¢Vorspann¢Text¢Tabelle¢|Format¢Notiz¢17.02.2004¢17.02.2004¢0¢¢¢
●T01_Artikel©    31©        ©                ©                62001©
●T01_Klinker©    30©        NEU©            @T01_Klinker:©    62103©
●T01_Bild©       27©        Telefon4©       ●Telefone©        62105©
●T01_Preis©      30©        199,50©        @T01_Preis:©     62102©
●T02_Bild©       27©        Telefon6©       ●Telefone©        62108©
●*Seitentitel©  30©        Warengruppe 01© @Seitentitel:©    62106©
¢0¢0¢Best.-Nr. 2¢XPress-Dok.¢Sicht¢£

```

Vor Beginn der Elementeliste stehen immer genau 14 Feldtrenner – hier ¢ –, jedes Element enthält im Format 2 fünf Elementtrenner – hier © –, die Elementeliste wird wie ein anderes Feld mit £ abgeschlossen, der Datensatz wird mit ¢ abgeschlossen. Danach kann unmittelbar der nächste Artikel folgen.

Will man nur die vier Haupttextfelder füllen, könnte das Formatschema einfach so lauten:

```
¢¢¢¢Artikel¢Vorspann¢Text¢Tabelle£¢¢¢¢Artikel¢Vorspann¢Text¢Tabelle£...
```

Scriptimport

Scriptimport

Das Ankreuzfeld des Importdialogs wechselt in den gleichnamigen Importmodus. (S. den vorhergehenden Abschnitt.)

c) ANSI-Zeichensatz

ANSI-Zeichensatz

Eine explizite Konvertierung in das ANSI-Format oder aus dem ANSI-Format ist nur nötig bei einem Textexport für die Weiterverarbeitung mit anderen Windows-Programmen oder bei

einem Textimport von Windows-Textdateien, die von einem Windows-Programm stammen.

Ist das Ankreuzfeld „ANSI-Format“ AN, so werden die exportierten Texte in das ANSI-Format umgesetzt. Öffnet man eine solche Textdatei dann unter Windows, enthält der Text auch dort z.B. die deutschen Umlaute.

Das Umgekehrte gilt für den Textimport: Stammt die Textdatei von einem Windows-Programm, d.h. wurde sie mit der ANSI-Zeichensatztafel geschrieben, so muß beim Import in DATAform das Ankreuzfeld „ANSI-Format“ auf AN gestellt werden. Die Zeichen werden dann in das Mac-Format umgesetzt – DATAform verwendet intern die MacOS-Zeichensatztafel.

Hintergrund: ANSI-Format und Mac-Format

Unter Windows und MacOS werden verschiedene Zeichensatztafeln verwendet.

(Computer speichern Buchstaben intern als Nummern ab. Das große A hat z.B. die Nummer 65, das kleine z die Nummer 122 - unter Windows und ebenso unter MacOS.)

Die beiden Tabellen sind identisch bei den ersten 127 Zeichen, unterschiedlich aber bei den meisten Zeichen im Bereich über 127. (Das große Ä hat unter MacOS die Nummer 128, unter ANSI-Windows die Nummer 196.) Die Umlaute, das ß und viele Sonderzeichen liegen

in diesem höheren Bereich. Läuft DATAform im Mehrplatzbetrieb, werden die unterschiedlichen Zeichen automatisch konvertiert. Auch eine auf einer Plattform erstellte DATAform-Datendatei kann direkt auf der anderen Plattform weiterbearbeitet werden. Exportiert man Artikel nach QuarkXPress, werden die Texte automatisch für die jeweilige Plattform umgesetzt. Dasselbe gilt für den Datenaustausch zwischen verschiedenen DATAform-Datenbanken – in keinem dieser Fälle muß man darauf achten, auf welcher Plattform man arbeitet.

XPressMarken im Text

XPressMarken im Text

Schaltet man das Ankreuzfeld „XPressMarken im Text“ AUS, werden alle im Text enthaltenen XPressMarken vor dem Export ausgefiltert.

Die XPressMarken in den Feldern 14 und 15.04 werden unverändert exportiert.

d) Trennzeichen

Viele Programme verwenden als Feldtrennzeichen für Texte Tabulator und Zeilenschaltung. Tabulator und Zeilenschaltung sind aber in unserem Fall selbst in Texten und Tabellen enthalten und deshalb hier ungeeignet.

Geeignet sind nur Zeichen, die in keinem Text, Namen, Dateipfad oder den XPressMarken vorkommen. Nicht geeignet sind außer allen Buchstaben und Ziffern z.B. folgende Zeichen: * , ; - — { } [] () \$ » • | : \ . @ " < > Zeilenschaltung (13), Zeilenvorschub (10), Tabulator (9).

Das Schaltfeld „DATAform-Symbol“ stellt die Trennzeichen wie abgebildet ein:

| Trennzeichen | ASCII | Zeichen |
|---------------|-------|---------|
| Feld: | 28 | ☒ |
| Elementefeld: | 29 | ☐ |
| Datensatz: | 30 | ☐ |

Die Trennzeichen lassen sich ansonsten auf drei Arten definieren:

- Man tippt die ASCII-Nummer des Zeichens jeweils in die linken Felder,
- man tippt das Zeichen selbst in die rechten Felder,
- man klickt auf ein „A“ und wählt das Zeichen aus einer Tabelle:



Der Dialog zeigt alle möglichen 256 Zeichen. Die Zeichensatztabelle stellt die Zeichen in der MacOS-Ordnung dar, wie sie auch innerhalb von DATAform verwendet wird.

Auswahl geeigneter Trennzeichen

Die Standardtrennzeichen sind 28, 29, 30. Dies sind in beiden Rechnerwelten drei der vier Pfeiltasten; sie können normalerweise nicht in einen Text eingegeben werden und eignen sich daher gut als Trennzeichen. Wenn möglich, sollten Sie diese Trennzeichen verwenden.

Nachteil dieser drei Zeichen: Wird der Text zur Bearbeitung in einem Textprogramm geöffnet, kann das Textprogramm diese Zeichen nicht darstellen. Stattdessen zeigt es meist ein schwarzes oder weißes Rechteck und die Zeichen sind optisch nicht unterscheidbar. In diesem Fall sind u.U. andere drei Zeichen vorzuziehen.

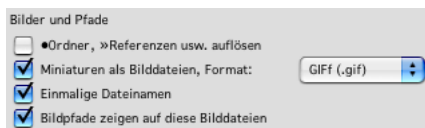
Evtl. geeignet und auch darstellbar sind dann z.B. die Zeichen: # & ¢ £ © ± μ, sowie § ®. Geeignet sind Zeichen generell dann, wenn sie sicher nicht in Texten, DATAformMarken, Namen (auch Farb- und Stilvorlagennamen) etc. enthalten sind.

Die fünf Zeichen ¢ £ © ± μ sind übrigens die einzigen Zeichen über 127, die unter Macintosh und Windows dieselben Nummern besitzen.

Folgende Zeichen sind wahrscheinlich in Ihren Texten nicht vorhanden (zumindest drei davon), werden in jedem Textverarbeitungssystem angezeigt, besitzen plattformunabhängige Nummern und eignen sich als Trennzeichen für eine Bearbeitung in einem Textprogramm:

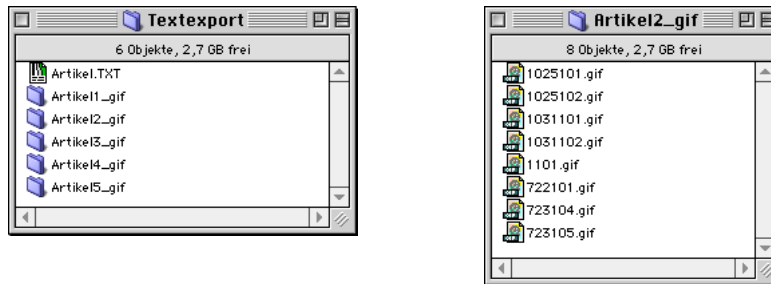
| | Nr. | Eingabe -Mac | Eingabe-Win |
|---|-----|--------------|-------------|
| & | 38 | Tastatur | Tastatur |
| ¢ | 162 | Wahl-4 | alt+0162 |
| £ | 163 | Wahl-Hoch-4 | alt+0163 |
| © | 169 | Wahl-G | alt+0169 |
| μ | 181 | Wahl-M | alt+0181 |

e) Bilder und Pfade



● Ordner, »Referenzen usw. auflösen

Ist diese Option AN, so werden die in den Elementen vorhandenen Ordnerbezüge aufgelöst, d.h. durch den Dateipfad ersetzt, für den sie stehen. Referenzen in den Bildnamen werden durch den Feldinhalt ersetzt, Bildendungen werden angefügt usw. Insgesamt wird derselbe Bildpfad berechnet und in die Textdatei exportiert, der auch bei der Übergabe an QuarkXPress gebildet wird. Diese Option ist auch ohne den Export von Miniaturen aktivierbar. Sie ermöglicht einer importierenden Datenbank den Zugriff auf die QuarkXPress-Bilder.

Miniaturen als Bilddateien exportieren

Alle in den Artikeln enthaltenen Miniaturen werden als eigene Dateien exportiert. Pro hundert Artikel erzeugt DATAform einen neuen Ordner für die Bilder dieser Artikel. Heißt die Export-Textdatei z.B. „Artikel.TXT“, so heißt der Ordner für die Miniaturen der ersten hundert Artikel „Artikel1_gif“ oder „Artikel1_pct“; der zweite Ordner für die zweiten hundert „Artikel2_gif“ usw. Dadurch können auch große Bildmengen exportiert werden, ohne das Betriebssystem durch zu große Ordner zu überlasten.

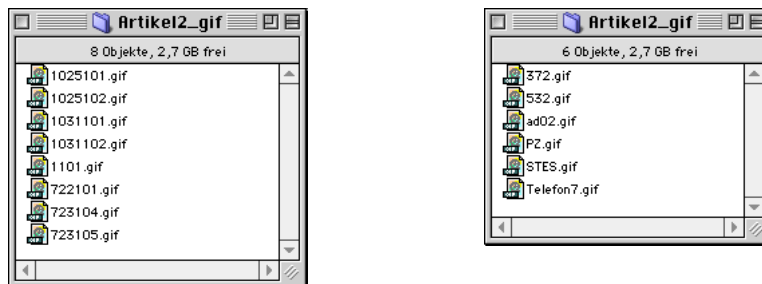


Die Miniaturen können als GIF- oder als PICT-Bilder exportiert werden.

Einmalige Dateinamen

Ist diese Option AN, so vergibt DATAform neue, einmalige Bildnamen für die erzeugten Bilddateien. Die Dateien werden nach der ID ihres Articlelements benannt, dies ist auch die DATAform-Rahmennummer in QuarkXPress. (Linkes Bild.)

Ist die Option AUS, werden die Bildnamen beibehalten und nur auf die Endung „.gif“ oder „.pct“ gesetzt. Gleichnamige Bilder werden beim Export ohne Warnung überschrieben. (Rechtes Bild) Zwei Bilder wurden überschrieben.

Bildpfade zeigen auf diese Bilddateien

Ist diese Option AN, so werden die Bildpfade und Namen der exportierten Miniaturen in die Textdatei geschrieben (in die Felder 15.03 Elemente.Text/Bildname und 15.04 Elemente.XPressMarken/Pfad).

Liest man die Textdatei wieder in eine Datenbank ein, so kann die Datenbank auf jedes dieser Bilder zugreifen.

Liest man sie in DATAform ein und ist unter Einstellungen/Bilder „Bilder liegen am individuellen Bildpfad“ aktiviert, so werden alle Bilder wieder gefunden und können z.B. durch den Befehl „Miniaturen aktualisieren“ wieder in DATAform dargestellt werden.

f) Duplikatebehandlung

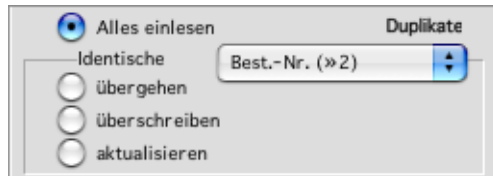
Sowohl beim Import von DATAform-Datensätzen als auch von Textdateien läßt sich definieren, wie DATAform Duplikate, d.h. Datensätze behandeln soll, die in der Datenbank bereits vorhanden sind.

Eine Anwendung dieser Funktion ist z.B. die verteilte Bearbeitung von Artikeln in verschiedenen Netzen/verschiedenen DATAform-Datenbanken. Artikel werden exportiert, in eine

andere DATAform-Datendatei eingelesen, dort bearbeitet und wieder in die ursprüngliche Datendatei zurückimportiert. Die Artikel der Stammdatenbank sollen aktualisiert werden.

Die geänderten Artikel sollen u.U. auch die Rahmeninhalte bereits existierender QuarkXPress-Rahmen aktualisieren. QuarkXPress-Rahmen sind über ihre Rahmenkennung mit den Datensatznummern (N^o) der Artikel in DATAform gekoppelt. Die Datensatznummern der Artikel müssen deshalb in diesem Fall im Gesamtvorgang erhalten bleiben.

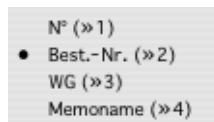
Der Duplikat-Dialog bietet die Möglichkeiten:



Duplikate/Alles einlesen

Alle Datensätze werden eingelesen und erhalten neue Datensatznummern (N^o). Die Kontrollfunktionen für Duplikate sind ausgeschaltet.

Duplikatfeld festlegen



Mit dem Klappenmenü legt man fest, anhand welchen Feldes nach Duplikaten gesucht werden soll. (Die Feldnamen können unter Einstellungen/Felder geändert worden sein.)

Duplikate/Übergehen

Ist der Artikel in der Datenbank bereits vorhanden, so wird er nicht eingelesen. Der Artikel der Datenbank wird nicht verändert. Ist das Duplikatfeld beispielsweise die Bestellnummer, so überprüft DATAform, ob die Bestellnummer bereits in der Datenbank vorkommt; in diesem Fall wird der Artikel übersprungen. Andernfalls wird er eingelesen.

Für die Vergabe neuer Datensatznummer N^o beim Einlesen neuer Datensätze gilt dabei:

- Ist das Duplikatfeld das Feld N^o (>>1), so behalten die importierten Artikel ihre Datensatznummern, es wird keine neue N^o vergeben.
- Bei den anderen drei Duplikatfeldern wird den importierten Artikeln eine neue N^o zugeteilt.

Duplikate/Überschreiben

Ist der Artikel in der Datenbank bereits vorhanden, so wird er gelöscht und durch den eingelesenen Artikel ersetzt. Der importierte Artikel erhält die N^o des überschriebenen Artikels (!). Ist das Duplikatfeld z.B. die Bestellnummer, so überprüft DATAform, ob die Bestellnummer bereits in der Datenbank vorkommt; ist dies der Fall, wird der Artikel in der Datenbank gelöscht und der neue Artikel an dessen Stelle, mit dessen N^o, importiert.

Ist der Artikel noch nicht vorhanden, so wird er als neuer Datensatz eingelesen. Für die Vergabe neuer Datensatznummer N^o beim Einlesen neuer Datensätze gilt dabei dasselbe wie beim Übergehen.

Für die verteilte Datenbearbeitung und Zusammenführung der Einzeldateien ergeben sich daraus folgende zwei Konzeptionen:

1. Überschreiben mit Duplikatfeld Bestellnummer, Warengruppe oder Memoname
 - Voraussetzung für diesen Weg sind eindeutige, nur einmal vorkommende Feldinhalte des Duplikatfeldes. (Ist ein Feldinhalt mehrfach vorhanden, erhält man beim Import eine Rückmeldung. Unter Einstellungen/Felder kann man das Feld zudem als „einmalig“ definieren, s. ebenda)

- Bei allen Importen arbeitet man immer mit derselben Duplikateinstellung.
 - Bei diesem Verfahren dürfen – sofern die Eindeutigkeit der Bestellnummern z.B. nach zugeordneten Bestellnummernkreisen eingehalten wird – in jeder Datendatei neue Artikel angelegt werden. (Beispiel: Verschiedene Übersetzer erhalten eigene Bestellnummernkreise und erzeugen danach selbst neue DATAform-Datensätze. Diese werden dann in einer Stammdatenbank zusammengeführt – erhalten dabei automatisch beim Erstimport und nur da neue Datensatz N^o – und können in QuarkXPress plaziert werden.)
 - Jede Datenbank kann nur ihre eigenen QuarkXPress-Dokumente aktualisieren, bzw. erzeugt und aktualisiert man alle QuarkXPress-Dokumente nur von einer Datenbank aus. (Beispiel: Die Korrekturen der Übersetzer werden zuerst in die Stammdatenbank importiert – die alten Artikel werden überschrieben, die N^o der Stammdatenbank (!) bleiben dabei erhalten –, von da aus werden die QuarkXPress-Dokumente aktualisiert.)
 - Bei diesem Weg sind die Datensatznummern N^o innerhalb der Stammdatenbank, und nur dort, quasi an das Duplikatfeld (die Best.-Nr. etc.) gekoppelt. Wurde z.B. einem Artikel mit einer Bestellnummer einmal in der Stammdatenbank eine N^o zugewiesen, so behält er diese seine Nummer auch dann, wenn er durch einen späteren Import überschrieben wird – die N^o, die der Artikel in der Teil-Datenbank hatte, wird ignoriert. Dies erlaubt an jedem Ablaufpunkt die Aktualisierung der QuarkXPress-Dokumente aus der Stammdatenbank heraus.
2. Überschreiben mit Duplikatfeld Datensatznummer N^o
- Bei diesem Verfahren dürfen nur in einer Datendatei neue Artikel angelegt oder Artikel dupliziert werden. (Beispiel: Alle neuen Datensätze werden in der Stammdatenbank angelegt; die Übersetzer arbeiten nur mit Exporten aus dieser Datenbank, sie legen selbst keine neuen Datensätze an. Über das Kennwortmodul kann man die entsprechenden Funktionen deaktivieren.)
 - Bei allen Importen in eine andere DATAform-Datenbank wird grundsätzlich nach dem Duplikatfeld N^o überschrieben oder übergangen. (Beispiel: Beim Import durch die Übersetzer, wie auch beim Rückimport in die Stammdatenbank, wird immer mit einer dieser Optionen gearbeitet – die N^o der Datensätze bleiben in allen Datenbanken erhalten.)
 - Alle Datendateien können zu jedem Zeitpunkt alle QuarkXPress-Dokumente aktualisieren, da die N^o in jedem Fall und in jeder Datendatei erhalten bleiben.

Duplikate/Aktualisieren

Die Option ermöglicht die gezielte Änderung einzelner Felder von Artikeln, ohne die übrigen Felder zu verändern. Beim Überschreiben wird der gefundene Artikel vollständig durch den importierten ersetzt; beim Aktualisieren hingegen werden im gefundenen Artikel nur die Felder überschrieben, die im importierten Artikel nicht leer sind. Alle anderen Felder des gefundenen Artikles bleiben unverändert.

Beispiel

In einem Katalog besitzen die Artikel je einen eigenen Textrahmen namens „●A Preis“ für einen auffallenden Preis. Diese Preise sollen anhand der Bestellnummern aktualisiert werden. Dieses Ziel läßt sich über alle drei Importarten erreichen, nachfolgend eine Skizze für den Scriptimport.

Man erstellt dazu in der Warenwirtschaft eine Tabelle mit den Bestellnummern und neuen Preisen der Art:

| | |
|-----------|----------|
| 803-2100D | 123,45 € |
| 803-2101E | 234,56 € |

Man importiert die Tabelle über ein Importsript. Als Vorlage öffnet man einen Artikel, der ein Element „●A Preis“ enthält. Die Duplikateinstellungen stellt man auf „Bestellnummer“ und

„aktualisieren“. Das Importsript lautet:

```
2 Best.-Nr. == F1
15 "●Preis" == F2
```

Beim Import wird dann zuerst der Artikel „803-2100D“ gesucht und aktualisiert; das Element „●A Preis“ erhält den neuen Text „123,45 €“. Da keine weiteren Inhalte importiert werden, bleiben die übrigen Felder und Elemente des Artikels unverändert.

Dann wird der Artikel „803-2101E“ gesucht und aktualisiert usw.

Nicht gefundene Artikel werden als neue Artikel angelegt, wobei die übrigen Felder den Inhalt der geöffneten Artikelvorlage erhalten.

1.6 Austausch mit QuarkXPress

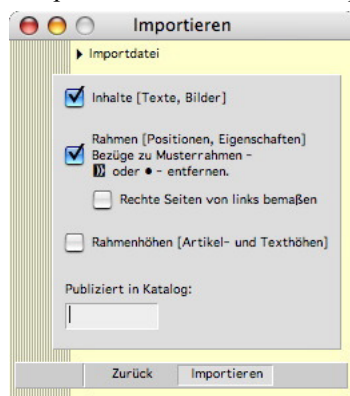
Die kleinste Einheit der bisher beschriebenen Schnittstellen ist ein ganzer Artikel mit Feldern und Elementliste.

Die kleinste Einheit eines DATAformMarken-Textes ist ein einzelner QuarkXPress-Rahmen, d.h. in der DATAform-Datenbank eine Zeile in der Elementliste. Ein DATAformMarken-Text ist eine Folge an Beschreibungen von QuarkXPress-Rahmen, durch den sich Rahmen und Rahmeninhalte erzeugen und aktualisieren lassen.

Anwendungsbeispiele

- Die normale Anwendung dieser Schnittstelle ist die Kommunikation mit QuarkXPress über das form-Menü in DATAform: die Platzierung von Rahmen in QuarkXPress, die inhaltliche Aktualisierung von QuarkXPress-Rahmen und die Rückaktualisierung von Änderungen an Rahmen oder ihren Inhalten aus QuarkXPress. Von beiden Seiten können gezielt einzelne Rahmen bzw. einzelne Zeilen in der Elementliste angesprochen werden.
- Eine fremde Datenbank kann einen DATAformMarken-Text erzeugen und darüber gezielt einzelne Zeilen in der Elementliste in der DATAform-Datenbank erzeugen oder ändern. Der Import einer DATAformMarken-Textdatei erfolgt mit dem Befehl form/Elemente importieren. Je nach Einstellungen/Satzspiegel wird dadurch die Textdatei „DATAform.AKT“ im QuarkXPress-DATAform-Ordner gelesen oder man erhält den Datei-Öffnen-Dialog zum Öffnen einer Textdatei.

Die fremde Datenbank kann, wie das DATAform-XTension, komplette Rahmenbeschreibungen erzeugen oder sich auf den Inhalt der Rahmen - Texte und Bildpfade - beschränken. Entsprechend kreuzt man im Importdialog nur Inhalte oder auch Rahmen an:



Format eines DATAformMarken-Textes

Die komplette Formatbeschreibung ist als eigenes Handbuch Teil der DATAform-Entwicklungsumgebung. Die folgenden Hinweise skizzieren lediglich die minimalen Anforderungen an eine DATAformMarken-Textdatei zur Änderung von Elementinhalten an einem Beispiel.

Der folgende Text aktualisiert den Artikelhauptrahmen eines Artikels sowie einen Zusatztextrahmen in einem anderen Artikel:

DfXT+2.0¶#11001*G11*T3*\$Text im Titelfeld ♦ Text im Vorspann ♦ Text im Textfeld ♦ Tabellentext¶#12103*G12*T3*\$NEUHEIT¶

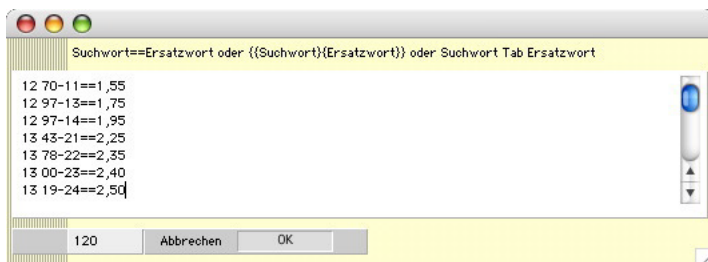
Die Textdatei muß mit „DfXT+2.0“ beginnen. Jeder Rahmen wird mit ¶ eingeschlossen. *G11 bezeichnet die Datensatznummer N^o des Datensatzes, der Datensatz N^o 11 soll aktualisiert werden. #11001 bezeichnet die Nummer des Elements, das aktualisiert werden soll. *T3 bedeutet, es handelt sich um einen Textrahmen (*T12 wäre ein Bildrahmen). Nach *\$ folgt der Text/Bildpfad des Rahmens, die Textfelder werden beim Hauptrahmen durch ♦ getrennt. (Vgl. Eingabemaske/Feldtrenner im Artikelrahmen)

1.7 Import von S&E-Tabellen

Über DATAform-Datensätze oder -Textdateien werden ganze Artikel ausgetauscht, über DATAformMarken-Texte können einzelne Zeilen in der Elementeliste erzeugt oder aktualisiert werden; über Suchen & Ersetzen-Tabellen können einzelne Wörter in Texten gesucht und geändert werden.

Die S&E-Funktionen in DATAform bieten sehr umfangreiche Funktionen z.B. zur Aktualisierung von Preisen, Neuberechnung von Preisen, allgemeinen Änderungen am gesamten Datenbestand oder ausgewählten Artikeln.

Das Format einer S&E-Tabelle und die S&E-Funktionen sind im Themenbereich „Suchen & Ersetzen“ ausführlich beschrieben.



Die Abbildung zeigt eine typische Tabelle aus Bestellnummern und zugeordneten Preisen: Die Bestellnummern werden dann in Spalte x gesucht, die Spalte y wird mit dem neuen Preis überschrieben.

1.8 Automatischer Import von Bildern

Mit dieser Importfunktion können automatisch alle Bilder aus einem Ordner importiert werden. Für jedes importierte Bild wird ein neuer Artikeldatensatz angelegt. Die Artikel können dann in einem Exportvorgang in QuarkXPress plziert werden.

Mögliche Anwendungen hierfür sind z.B. der Nachdruck alter Bücher oder die Herstellung eines Bildkatalogs.

Ablaufschema Nachdruck von Büchern

- jede Seite wird gescannt, die Dateien werden benannt wie „0001“, „0002“ usw.,
- alle Bilder werden dann automatisch in DATAform importiert,
- die entstandenen Datensätze werden nach dem Titel sortiert,
- die Bilder werden schließlich automatisch in QuarkXPress auf jeweils eigene Seiten plziert.

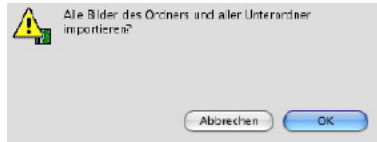
Anwendungsbeispiel Bildkatalog

- Legen Sie alle Bilder, die Sie in DATAform importieren wollen in einen Ordner oder rufen Sie den Bildimport jeweils für die einzelnen Ordner auf. Der Importvorgang importiert alle Bilder eines Ordners und seiner Unterordner.
- Rufen Sie den Menübefehl DATA/Import auf. Es öffnet sich der Importdialog:
Öffnen Sie Ihren Bildordner und markieren Sie ein beliebiges Bild in diesem Ordner. OK.

Der Importdialog zeigt alle vorhandenen Dateien an. Je nachdem, welche Datei Sie öffnen,

erhalten Sie automatisch den entsprechenden Importdialog:

- Wenn Sie eine TEXT-Datei öffnen, erhalten Sie den Textimportdialog.
- Wenn Sie eine DATAform-Datensatzdatei öffnen, erhalten Sie den Dialog zum Import von Datensätzen.
- Wenn Sie ein Bild öffnen, erhalten Sie den Dialog links:

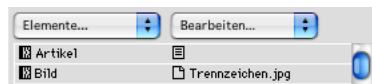


Nach OK werden alle Bilder in jeweils eigene Artikel importiert, rechtes Bild.

Es können alle Bildformate importiert werden, die QuarkXPress unterstützt - eine Liste dieser Formate finden Sie im nächsten Kapitel unter „Bildformate“.

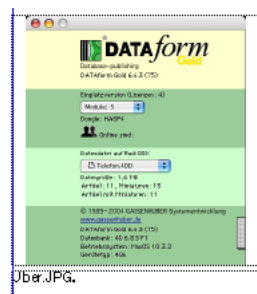
DATAform versucht, aus jedem Bild eine Miniatur zu erzeugen. Konnte für ein Bild keine Miniatur erzeugt werden, wird das Fehlerbild angelegt. Das Bild wurde aber auch in diesem Fall dem Artikel zugeordnet; es kann ebenso in QuarkXPress platziert werden.

- Öffnen Sie nach dem Importvorgang einen Artikel, so sehen Sie:
Für jedes importierte Bild wurde ein Artikel mit zwei Muster-Elementen angelegt:



Der Name des Bildes wurde zusätzlich in das Titelfeld des Artikels geschrieben. Nach diesem Feld kann man einfach sortieren oder suchen.


- Definieren Sie eine Stilvorlage für die Bildunterschrift. Für das erste Feld legen Sie beispielsweise als Standard-XPressMarke die Stilvorlage „@Bildunterschrift.“ fest.
- Ändern Sie nun die Musterrahmen (Befehl „Muster exportieren“ im rechten Klappenmenü in DATAform und dann Befehl „Rahmen importieren“ in QuarkXPress)

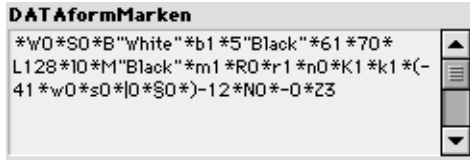


Dieser Bildrahmen wurde in QuarkXPress in eine quadratische Form gebracht, der Artikelrahmen wurde als Bildunterschrift platziert. Ändern Sie die Stilvorlage „Bildunterschrift“. Übernehmen Sie die Rahmenänderungen zurück in die Datenbank. („Gruppe exportieren“ aus QuarkXPress und „Elemente importieren“ in DATAform.)

- Bildpfade
Die Pfade zu den importierten Bildern wurden am individuellen Bildelement abgespeichert. Um diese Pfade zu verwenden, stellen Sie in DATAform unter Einstellungen/Bilder die Bildsuche auf „individuellen Bildpfad“ und/oder aktivieren Sie die tolerante Suche.
- Satzspiegel
Definieren Sie unter Einstellungen/Satzspiegel Ihr Layout, die Spaltenanzahl etc.
- Suchen und sortieren Sie die Bilder und platzieren Sie sie in QuarkXPress. Die Bilder werden dabei je nach Definition des Musters „Bild“ skaliert und im Rahmen platziert.

- Bilder dem Rahmen anpassen
Sind die Bilder unterschiedlich groß, so ist eine einheitliche Skalierung für alle Bilder unbefriedigend; die Bilder sollten in diesem Fall jeweils passend in den Rahmen verkleinert werden. Um das zu erreichen, gibt es folgenden „Trick“:

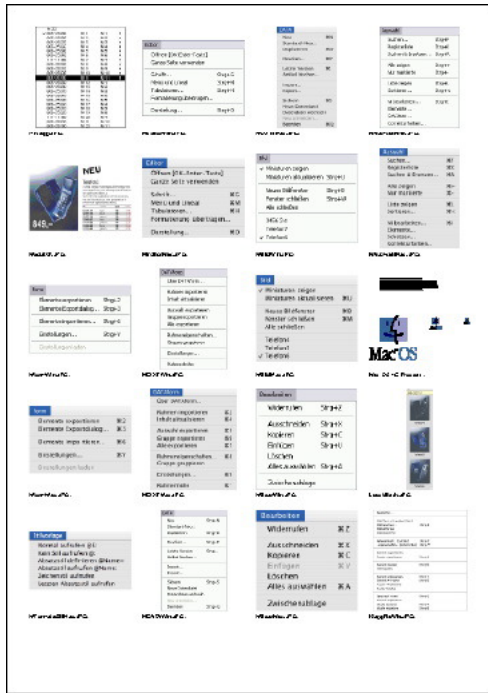
Bearbeiten Sie unter Einstellungen/Rahmen das Rahmenmuster „ Bild“ und fügen Sie als zusätzliche DATAformMarke den Ausdruck „*Z3“ (Großes Z ist erforderlich.) ein:



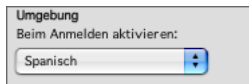
Die Bilder werden dadurch proportional in den Rahmen eingepaßt.
Der Ausdruck „*Z3“ bleibt bei einer Änderung des Musters in QuarkXPress und einer Rückaktualisierung nicht erhalten. Die Marke wirkt nur in der Richtung DATAform → QuarkXPress.

- Nachfolgend alle Möglichkeiten für *Z, vgl. das Bildmenü „Stil“ in QuarkXPress:
- *Z4 Rahmen an Bild anpassen, ab QuarkXPress 5, wie per Menü „Stil“ in QuarkXPress
 - *Z3 Bild proportional an den Rahmen anpassen, wie Befehl-Wahl-Hoch-F in QuarkXPress
 - *Z2 Bild vollständig an den Rahmen anpassen, wie Befehl-Hoch-F in QuarkXPress.
 - *Z1 Bild im Rahmen zentrieren, wie Befehl-Hoch-M in QuarkXPress.
 - *Z0 (oder fehlende Marke) Das Bild wird, je nach Angaben, skaliert und versetzt.

- Je nach Ihren Layoutdefinitionen erhalten Sie Bildkatalogseiten wie diese, ein Katalog aller Bilder dieses Handbuchs:



Umgebung



Per Klappmenü kann man dem Kennwort eine eigene Umgebung zuordnen. Zu einer Umgebung gehören alle in den Einstellungsseiten blau hervorgehobenen Merkmale. Meldet sich dieser Benutzer an, wird für ihn diese Umgebung, z.B. mit spanischen Feldbeschriftungen, geladen. Das Einrichten der Umgebungen erfolgt auf der Seite form/Einstellungen/Verwaltung, siehe Seite: 159.

2. Modul: Scriptexport

Der Scriptexport in DATAform ermöglicht den Export der Artikel in einem frei definierbaren Textformat. Alle Text und Bildpfade können z.B. als HTML-Datei oder XML-Datei exportiert werden.

Der Scriptexport erlaubt die freie Definition der Feldreihenfolge und ermöglicht den Zugriff auf einzelne Zellen innerhalb von Texten. Eine Zelle ist z.B. der Text in der dritten Spalte der vierten Zeile oder der Text der letzten Spalte der ersten Zeile.

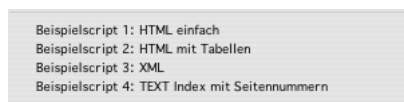
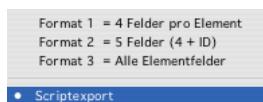
Eine mit einem Exportscript erzeugte HTML-Datei kann komplette Artikel inklusive Bildverweise enthalten und mit jedem Browser dargestellt werden. Der eigentliche Zweck des Exportscripts ist aber die Datenübernahme z.B. in einen Webserver.

Exportscript schreiben und anwenden

In sechs Schritten zu einer HTML-Version Ihrer Artikel.

DATAform enthält drei Funktionen zur Erzeugung eines fertigen Exportscripts auf der Grundlage des aktuellen Artikels. Die Funktionen liefern Beispiele, Vorschläge für einen schnellen Zugang. Ein Ihrem Datenbestand angemessenes Script wird in aller Regel eine Nachbearbeitung erfordern.

- Suchen Sie einige Artikel, die Sie exportieren möchten.
- Wählen Sie den Befehl Export im DATA-Menü.
- Wählen Sie im Exportdialog links oben im Klappmenü „Textexport-Einstellungen“ die Zeile „Scriptexport“, linkes Bild:



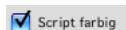
- Wählen Sie im Klappmenü „Script“ rechts oberhalb des Textfeldes die Zeile „Beispielscript 1: HTML einfach“, rechtes Bild. Ein komplettes HTML-Script mit Referenzen auf alle Felder des aktuellen Artikels wird in das Textfeld eingesetzt.
- Klicken Sie auf die Schaltfläche „Textexport.txt“ rechts unten und sichern Sie die Datei im folgenden Datei-Dialog.
Das Exportscript wird nun auf jeden Artikel der Auswahl angewandt und seine Daten werden in die Datei geschrieben.
- Ziehen Sie erzeugte Datei auf einen HTML-Browser: Alle Feldnamen, Inhalte und Bilder aller Artikel werden fortlaufend dargestellt.
- Wählen Sie nun den Befehl „Beispielscript 2: HTML mit Tabellen“, verfahren Sie wieder wie oben beschrieben und betrachten Sie das Ergebnis im Browser: Der Kopfteil, das Tabellenfeld und die Elemente der Artikel werden als HTML-Tabelle dargestellt.

Bestandteile eines Exportscripts

Ein Exportscript funktioniert nach dem Grundprinzip des Platzhalters. In einem beliebigen HTML-, XML-Text oder auch zwischen Tabulatoren werden Platzhalter als Referenzen auf DATAform-Texte eingefügt. <DF Field="10"/> ist beispielsweise ein Platzhalter auf Feld

Nummer 10, das Notizfeld.

In der linken Liste des Exportdialogs sehen Sie die Felder des aktuellen Artikels. Klicken Sie in eine Zeile, so wird der Platzhalter für dieses Feld in das Scriptfeld rechts eingefügt.



Stellen Sie das Ankreuzfeld rechts oben im Dialog auf AN. Sie sehen das Script etwa so (Sie sehen die Farben der Abbildung in der PDF-Version dieses Handbuchs):

```

<html>
<body>
<DF Record="Start"/>
<Table border="2">
<TR>
<TH>Bestellnummer
<TH>NR
<TR>
<TD><DF Field="2"/>
<TD><DF Field="3"/>
<TD><DF Field="16"/>
</Table>
<DF Field="1" Line="All" Column="All" Ifbefore=<br><B>Artikel:</B><B>ENDBetrag</B></DF>
<DF Field="5" Line="All" Column="All" Ifbefore=<br><B>Umspann</B><B>or</B>ENDBefore</B></DF>
<DF Field="7" Line="All" Column="All" Ifbefore=<br><B>Text</B><B>or</B>ENDBefore</B></DF>
<Table border="2">
<DF Field="8" Line="Table" Column="Table" Ifbefore=<TH>Tabelle</DFbefore>
</Table><br>
<DF Record="End"/>
</body>
</html>

```

Grün hinterlegt Das Script zwischen den beiden grünen Kennern wird pro exportiertem Artikel angewandt. Die blauen Texte darüber und darunter werden nur einmal pro Export in die Datei geschrieben.

Blaue Schrift Dieser Text wird unverändert exportiert. Der blaue Text innerhalb der grünen Markierungen wird für jeden Artikel aufs Neue ausgegeben.

Gelb hinterlegt <DF >-Platzhalter für die Daten des Artikels. Alle diese Platzhalter werden pro Artikel durch ihren jeweiligen Inhalt ersetzt und ausgegeben.

Es kann also beliebiger (blau dargestellter) HTML- oder XML-Text in den Editor eingegeben werden, in den dann die eigentlichen Datenbank-Inhalte der Artikel per <DF >-Platzhalter eingebettet werden.

Der Platzhalter <DF > kann verschiedene Attribute enthalten: Record, Field und Char. Eine Liste der Attribute mit kurzen Erklärungen finden Sie im Klappmenü „Script“ rechts oberhalb des Scriptfeldes. Wählen Sie dort ein Attribut aus, wird es an der Position der Einfügemarke ins Scriptfeld eingefügt.

```
<DF Record="Start"/>
```

```
<DF Record="End"/>
```

Die beiden Marken markieren Beginn und Ende des Scriptteils, der auf jeden Artikel beim Export angewandt wird. Davor stehen Texte, die nur einmal am Dateianfang ausgegeben werden sollen wie <html><body>. Danach stehen Texte, die nur einmal am Dateiende ausgegeben werden sollen wie </body></html>. DATAform-Platzhalter wie <DF Field=""> dürfen nicht außerhalb dieser Marken verwendet werden.

```
<DF Field="x"/>
```

Platzhalter für das Feld Nr. x. Die möglichen Nummern und ihre Bedeutung entnehmen Sie der linken Liste im Scriptexport-Dialog. Ein Klick in dieser Liste trägt den Platzhalter automatisch ein. Der Platzhalter <DF Field=""> kann verschiedene Attribute besitzen, die nun genau erläutert werden.

DF Field - Zeilen und Spalten

Line=, Column=

Line="All" Column="All" oder Line="0" Column="0" oder fehlend. Beispiele:

```
<DF Field="8" Line="All" Column="All"/> oder <DF Field="8"/>
```

Gibt den gesamten Text des Feldes 8, der Tabelle, aus. Ersetzt Zeilenschaltungen durch „
“ wenn das Exportscript zu Beginn <html> enthält.

Line="3" Column="All"

Liefert die komplette 3. Zeile des Feldes.

Line="4" Column="3"

Liefert aus der 4. Zeile die 3. Spalte, also das Feld 4/3.

Line="All" Column="3"

Liefert aus allen Zeilen die Spalte 3, also die komplette 3. Spalte. Ersetzt Zeilenschaltungen durch „
“ wenn im HTML-Format exportiert wird.

LFrom="Bottom"

Zählt die Zeilen von unten. Beispiel:

```
<DF Field="8" Line="1" Column="1" LFrom="Bottom"/>
```

Liefert die erste Zelle der letzten Zeile aus dem Tabellenfeld.

CFrom="Right"

Zählt die Spalten von rechts. Jede Spalte wird eigens von rechts gezählt. Beispiel:

```
<DF Field="8" Line="1" Column="1" LFrom="Bottom" CFrom="Right"/>
```

Liefert die letzte Zelle des Tabellenfeldes.

Line="Table", Column="Table"

Line="Table" Column="Table"

Liefert den kompletten Text als Inhalt einer HTML-Tabelle für einen HTML-Export. Tabellenanfang und -ende müssen definiert werden. Z.B.:

```
<Table border="2">
```

```
<DF Field="8" Line="Table" Column="Table"/>
```

```
</Table>
```

Liefert aus der Telefondemo diese Tabelle:

| N° | color | typ | cond. |
|----------|--------|-----|-------|
| 12 70-11 | red | 10 | 1,55 |
| 12 89-12 | yellow | 10 | 1,65 |
| 12 97-13 | blue | 10 | 1,75 |
| 12 97-14 | green | 10 | 1,95 |
| 13 43-21 | red | 15 | 2,25 |
| 13 78-22 | yellow | 15 | 2,35 |

Line="4" Column="Table"

Liefert die 4. Zeile als Inhalt einer HTML-Tabelle. Tabellenanfang und -ende müssen definiert werden. Weiteres Beispiel:

```
<Table border="2">
```

```
<TR><DF Field="8" Line="1" Column="Table"/>
```

```
<TR><DF Field="8" Line="2" Column="Table"/>
```

```
<TR><DF Field="8" Line="1" Column="Table" LFrom="Bottom"/>
```

```
</Table>
```

Liefert eine Tabelle mit den ersten beiden und der letzten Zeile aus dem Tabellenfeld.

| N° | color | typ | cond. |
|----------|--------|-----|-------|
| 12 70-11 | red | 10 | 1,55 |
| 13 78-22 | yellow | 15 | 2,35 |

Line="Table" Column="2"

Liefert die komplette 2. Spalte als Inhalt einer HTML-Tabelle. Tabellenanfang und -ende müssen definiert werden.

| |
|--------|
| color |
| red |
| yellow |
| blue |
| green |
| red |
| yellow |

DF Field - ZahlenType="Number"

Das Ergebnis des Platzhalters muß eine Zahl entsprechend dem Eingabe-Zahlenformat sein (S&E-Dialog/Berechnen und Numerieren).

Leertasten vor und nach der Zahl werden übergangen. Ansonsten darf der Ergebnistext keine weiteren Zeichen enthalten.

Ist das Ergebnis eine Zahl, so wird sie ins Ausgabe-Zahlenformat gesetzt und ausgegeben. Die Umformatierung betrifft nur den Export, die Daten in DATAform werden nicht verändert.

Ist das Ergebnis keine Zahl, wird nichts oder der Text zwischen IFempty= und ENDempty ausgegeben. Siehe das Beispiel 2 unter IFempty weiter unten.

Type="Number?"

Das Ergebnis kann eine Zahl sein.

Ist das Ergebnis eine Zahl, so wird sie ins Ausgabe-Zahlenformat gesetzt und ausgegeben.

Ist das Ergebnis keine Zahl, so wird das Ergebnis ausgegeben.

Type="Text"

Das Ergebnis muß Text, darf keine Zahl sein.

Ist das Ergebnis eine Zahl, so wird nichts oder der Text zwischen IFempty= und ENDempty ausgegeben.

Ist das Ergebnis keine Zahl, so wird das Ergebnis ausgegeben.

DF Field - Bedingte Ausgaben

Die folgenden Attribute erlauben die Ausgabe von Texten in Abhängigkeit vom Ergebnis.

IFbefore=xxENDbefore

Der Text zwischen den Marken, im Beispiel „xx“, wird nur ausgegeben, wenn das Ergebnis des Platzhalters nicht leer ist. In diesem Fall wird der Text zwischen den Marken vor dem Ergebnis des Platzhalters ausgegeben.

IFafter=xxENDafter

Der Text zwischen den Marken, im Beispiel „xx“, wird nur ausgegeben, wenn das Ergebnis des Platzhalters nicht leer ist. In diesem Fall wird der Text zwischen den Marken nach dem Ergebnis des Platzhalters ausgegeben.

Beispiel:

```
<DF Field="15" Name="3+3 M1" IFbefore=ENDafter/>
```

Nur wenn das Element "3+3 M1" gefunden wird und wenn es einen Bildnamen enthält, wird vor der Pfadangabe , der HTML-Aufruf für das Bild.

IFempty=xxENDempty

Der Text zwischen den Marken „xx“ wird nur ausgegeben, wenn das Ergebnis des Platzhalters leer ist.

Beispiel 1:

```
<DF Field="8" Line="5" Column="10" IFempty=Zelle 5/10 ist leerENDempty/>
```

Ist die Zelle 5/10 leer, so wird stattdessen der Text „Zelle 5/10 ist leer“ ausgegeben. Den erzeugten HTML-Text kann man dann nach diesen Fehlermeldungen durchsuchen etc.

Beispiel 2:

```
<DF Field="8" Line="5" Column="1" CFrom="Right" Type="Number" IFempty=Preis fehlt!ENDempty/>
```

Die letzte Zelle in Zeile 5 muß eine Zahl enthalten (Type="Number"). Ist das nicht der Fall, so wird stattdessen „Preis fehlt!“ ausgegeben.

Zusammenfassendes Beispiel

Der Text folgender Tabelle wurde in DATAform im Tabellenfeld eingegeben:

| Nº | color | typ | cond. |
|----------|--------|-----|-------|
| 12 70-11 | red | 10 | 1,55 |
| 12 89-12 | yellow | 10 | 1,65 |
| 12 97-13 | blue | 10 | 1,75 |
| 12 97-14 | green | 10 | 1,95 |
| 13 43-21 | red | 15 | 2,25 |
| 13 78-22 | yellow | 15 | 2,35 |

Die Tabellen in den verschiedenen Artikeln sind bis zu 10 Zeilen hoch.

Es soll immer eine passende HTML-Tabelle erzeugt werden.

Spalten und Zeilen sollen getauscht werden.

Typ und color soll getauscht werden.

Die Preise sollen überprüft werden und mit der Endung EUR ausgegeben werden.

Im Browser soll die Tabelle dann so erscheinen:

| Nº | 12 70-11 | 12 89-12 | 12 97-13 | 12 97-14 | 13 43-21 | 13 78-22 |
|-------|----------|----------|----------|----------|----------|----------|
| typ | 10 | 10 | 10 | 10 | 15 | 15 |
| color | red | yellow | blue | green | red | yellow |
| cond. | 1,55 EUR | 1,65 EUR | 1,75 EUR | 1,95 EUR | 2,25 EUR | 2,35 EUR |

Folgendes Exportscript erfüllt diese Aufgaben:

```

<html>
<body>
<DF Record="Start"/>

<Table border="2">
<TR>
<DF Field="8" Line="1" Column="1" IFbefore=<TH>ENDbefore />
<DF Field="8" Line="2" Column="1" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="3" Column="1" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="4" Column="1" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="5" Column="1" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="6" Column="1" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="7" Column="1" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="8" Column="1" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="9" Column="1" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="10" Column="1" IFbefore=<TD>ENDbefore />
<TR>
<DF Field="8" Line="1" Column="3" IFbefore=<TH>ENDbefore />
<DF Field="8" Line="2" Column="3" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="3" Column="3" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="4" Column="3" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="5" Column="3" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="6" Column="3" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="7" Column="3" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="8" Column="3" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="9" Column="3" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="10" Column="3" IFbefore=<TD>ENDbefore />
<TR>
<DF Field="8" Line="1" Column="2" IFbefore=<TH>ENDbefore />
<DF Field="8" Line="2" Column="2" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="3" Column="2" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="4" Column="2" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="5" Column="2" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="6" Column="2" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="7" Column="2" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="8" Column="2" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="9" Column="2" IFbefore=<TD>ENDbefore />
<DF Field="8" Line="10" Column="2" IFbefore=<TD>ENDbefore />
<TR>
<DF Field="8" Line="1" Column="4" IFbefore=<TH>ENDbefore Type="Number?" />
<DF Field="8" Line="2" Column="4" IFbefore=<TD>ENDbefore Type="Number?" />
<DF Field="8" Line="3" Column="4" IFbefore=<TD>ENDbefore Type="Number?" />
<DF Field="8" Line="4" Column="4" IFbefore=<TD>ENDbefore Type="Number?" />
<DF Field="8" Line="5" Column="4" IFbefore=<TD>ENDbefore Type="Number?" />
<DF Field="8" Line="6" Column="4" IFbefore=<TD>ENDbefore Type="Number?" />
<DF Field="8" Line="7" Column="4" IFbefore=<TD>ENDbefore Type="Number?" />
<DF Field="8" Line="8" Column="4" IFbefore=<TD>ENDbefore Type="Number?" />
<DF Field="8" Line="9" Column="4" IFbefore=<TD>ENDbefore Type="Number?" />
<DF Field="8" Line="10" Column="4" IFbefore=<TD>ENDbefore Type="Number?" />
</Table>

<DF Record="End"/>
</body>
</html>

```

DF Field - Elementnamen

Name=

Bei Platzhaltern für das Feld 15, die Elementliste, muß nach *Name=* der Name des Elements angegeben werden.

Beispiel 1:

```
<DF Field="15" Name="3+3 B1" IFbefore=ENDafter/>
```

Pro Artikel sucht DATAform beim Export nach einem Element mit dem Namen "3+3 B1" und gibt dessen Bildpfad aus, wenn es ein Bildelement ist.

Beispiel 2:

```
<DF Field="15" Name="1ST5 T5" />
```

Pro Artikel sucht DATAform beim Export nach einem Element mit dem Namen "1ST5 T5" und gibt dessen Text aus, wenn es ein Textelement ist.

Die übergebenen Elementnamen können auch das Jokerzeichen @ enthalten. „@“ steht für einen, keinen oder mehrere Buchstaben. Elementnamen sollte so aufgebaut werden, daß der erste Teil die Gruppenzugehörigkeit angibt, der letzte Teil die Elementart wie Text1, Text2, Bild1, Bild2 innerhalb der Gestaltungsvariante. Innerhalb der Gruppe muß jeder Name einmalig sein. Werden beim Export mehrere Elemente mit demselben Namen gefunden, wird nur das erste exportiert.

Beispiel 3:

```
<DF Field="15" Name="@T5" Line="2" Column="1" CFrom="Right" Type="Number?"/>
```

Pro Artikel sucht DATAform beim Export nach einem Element dessen Name mit "T5" endet. Wird das Element gefunden und ist es ein Textelement, wird die letzte Zelle der zweiten Zeile ausgegeben. Enthält die Zelle eine Zahl, so wird die Zahl dem Ausgabe-Zahlenformat entsprechend formatiert. Enthält sie Text, wird der Text ausgegeben.

HTML-Bildpfade

Ist das Ankreuzfeld AN, werden folgende Funktionen geändert:

- Klickt man in der linken Liste auf ein Bildelement, so wird der Platzhalter für dieses Bild in das Script eingefügt und der Platzhalter enthält dann schon die HTML-Marken für den Bildverweis ``.
- Die exportierten Bildpfade werden in dieser Einstellung als relative Pfade und in HTML-Schreibweise generiert. Beispiel: Die erzeugte Datei heißt „Artikel.txt“. Die automatisch exportierten GIF-Bilder werden dann im Ordner „Artikel1_gif“ abgelegt, s. „Miniaturen als Bilddateien“ exportieren auf Seite: 192. Die exportierten Bildpfade lauten dann z.B. ``

Die übrigen Einstellungen „ANSI-Zeichensatz“, „XPressMarken im Text“ sowie für Bilder und Pfade sind dieselben wie für den Textexport in den Formaten 1 bis 3, siehe Seite: 189ff.

HTML-Ausdrücke werden auch bei ausgefilterten XPressMarken exportiert. Schaltet man den Export von XPressMarken im Text- oder Scripexport ab, werden HTML-Ausdrücke wie `` oder `<A HREF...>` nicht ausgefiltert, obwohl sie XPressMarken ähneln. Diese Ausdrücke werden exportiert. Für die HTML-Ausgabe können damit beliebige Bilder in den Text eingebunden werden.

Anwendungsbeispiel: Ausgabe von Sonderzeichen als Bilder in HTML.

Um das Eurozeichen oder andere, eigene Zeichen für den Browser auszugeben ist die Konvertierung in ein GIF-Bild oft der richtige Weg. Mit einer S&E-Tabelle kann man alle Sonderzeichen durch eine Bildreferenz ersetzen, die dann im Browser dargestellt wird, z.B.:

```
€==
```

```
<f"Meine Schrift">f<f$>==
```

In diesem Beispiel müßte man neben die erzeugte Export-Textdatei ("Artikel.txt") einen Ordner "Konstanten" anlegen, der die Bilder enthält.

XML-Export

Wählen Sie den Befehl „Beispielscript 3: XML“ im Script-Klappmenü rechts oben. Ein Exportscript für XML wird erzeugt.

Alle Felder werden durch dieses Beispiel als ganzes exportiert. Will man Zellen einzeln ausgeben, fügt man die entsprechenden Attribute Line= und Column= ein.

Die HTML-Tabellenformatierungen werden unterdrückt; Line="Table" ist gleichbedeutend mit Line="All". Außer der evtl. Umsetzung in den ANSI-Zeichensatz werden die Texte unverändert exportiert.

Das Beispielscript bildet XML-konforme Marken-Namen unter Verwendung der aktuellen Feldnamen.

ANSI-Zeichensatz wird beim Einsetzen des Scriptbeispiels auf AN gestellt.

Das Feld des XPress-Markenvorspanns wird nicht plaziert, da XML-Browser mit dessen Inhalt nicht zurecht kommen. Für einen Export für eine Datenübernahme kann das Feld aber ohne weiteres hinzugefügt werden.

Die Bildpfade werden je nach Einstellungen in HTML-Notation oder als Originalpfade ausgegeben.

DF CharNr und andere Formate

Mithilfe des Platzhalters <DF CharNr="x"/> können beliebige Exportformate generiert werden. Das Exportscript besteht dann z.B. aus der Zeile:


```
<DF Field="1"/><DF CharNr="28"/><DF Field="2"/><DF CharNr="28"/><DF
Field="3"/><DF CharNr="28"/><DF Field="4"/><DF CharNr="30"/>
```

Die Felder 1 bis 4 werden mit Zeichen 28 getrennt ausgegeben. Als Datensatztrenner erscheint das Zeichen 30 am Ende. Zwischen den Platzhaltern dürfen in diesem Fall keine Zeilenschaltungen eingegeben werden, sie erscheinen sonst als Teil der Daten (in HTML oder XML werden Zeilenschaltungen ignoriert). Auch hier können natürlich die Attribute Line= und Column= für den Zugriff auf einzelne Felder verwendet werden. Die Einstellungen für Bildpfade etc. sind dieselben wie bei den anderen Formaten.

Eine derartige Datei kann dann z.B. mit dem DATAform-Scriptimport und den Trennzeichen 28-30 wieder importiert werden.

Seitenindex

Das vierte Beispiel im Script-Klappmenü dient der Erzeugung eines Seitenindex:



```
Beispielscript 1: HTML einfach
Beispielscript 2: HTML mit Tabellen
Beispielscript 3: XML
Beispielscript 4: TEXT Index mit Seitennummern
```

Der Befehl erzeugt dieses kleine Script:

```
<DF Field="2"/><DF CharNr="9"/><DF Field="10" Line="2" LFrom="Bottom"/><DF
CharNr="13"/>
```

Das Script erzeugt eine Tab-Zeilenschaltungsdatei mit allen Bestellnummern + Tab + Seitennummern aus der vorletzten Zeile des Notizfeldes der ausgewählten Artikel. Die Seitennummern werden vorher aus QuarkXPress importiert, siehe „Elemente importieren“, Seite: 123f.

Sonderzeichen, reservierte Zeichen

DATAform Sonderzeichen

Folgende Zeichen haben an bestimmten Positionen eine besondere Bedeutung.

| | MacOS | Windows | Anwendung |
|---|--------------------------|-------------------------|---|
| » | Wahl-Hoch-Q ASCII 200 | Alt+0187 ASCII 187 | Referenz auf ein Feld anstatt eines Bildnamens; Bildname steht im Feld »Nr., z.B. „»4“ |
| @ | Tastatur ASCII 64 | Tastatur ASCII 64 | Platzhalter beim Suchen; oder Stilvorlagenaufruf wie „@Tabelle:“ |
| ● | Wahl-Ü ASCII 165 | Alt+0149 ASCII 149 | Namensanfang für Element mit Musterbezug wie „●Rahmen“ oder für Ordnernamen |
| | Wahl-7 ASCII 124 | Strg+Alt+< ASCII 124 | Beginn eines Formataufrufs wie in „ 3spezial“ |

Die Sonderzeichen können in DATAform auch durch das ●-Zeichenmenü, das Formate-Klappenmenü oder das Referenzmenü erzeugt werden.

Reservierte Zeichen: " ¶ @ : { } ; =

In Schrift-, Farb-, Musterseitenamen und im Seitenpräfix der Kapitelseite in QuarkXPress sind die beiden Zeichen " (ASCII 34) und ¶ (ASCII 166) nicht erlaubt. Diese Zeichen sind für die Kommunikation zwischen QuarkXPress und DATAform reserviert.

ASCII 166 erscheint unter MacOS als ¶, unter Windows meist als ¶.

Praktisch relevant ist hauptsächlich: Farbnamen dürfen kein Zollzeichen (") enthalten. (Enthalten Farbnamen ein Zollzeichen, so kann diese Farbe von DATAform in QuarkXPress nicht zugewiesen werden. Als Farbe wird dann weiß oder schwarz verwendet.)

Namen von Stilvorlagen dürfen zudem diese Zeichen nicht enthalten: @ : { } ; =

Reservierte Zeichen in Texten mit XPressMarken: < \ @

Text mit XPressMarken darf folgende drei Zeichen nicht enthalten: < \ @

Diese Zeichen sind als Kenner für XPressMarken reserviert. Notfalls müssen sie durch die entsprechende XPressMarke maskiert werden. Ein < z.B. durch <\> (Siehe hierzu ausführlich den Abschnitt „Text mit XPressMarken“ im Datenbank-Grundlagenkapitel, Seite: 80f.)

Enthält ein Text diese Zeichen, so interpretiert sie der XPressMarken-Filter als Beginn einer XPressMarke und unterdrückt den folgenden Text.

Zeichensatztabellen

Genormter ASCII-Zeichensatz für alle Betriebssysteme

| | | | | | | | | | |
|-------|---|------|------|------|-------|-------|-------|--|--|
| 9 | Tabulator, HT | | | | | | | | |
| 10 | Zeilenvorschub, Linefeed, LF | | | | | | | | |
| 13 | Zeilenschaltung, Wagenrücklauf, Carriage Return, CR | | | | | | | | |
| 32 | Leertaste | | | | | | | | |
| 32 | 44 , | 56 8 | 68 D | 80 P | 92 \ | 104 h | 116 t | | |
| 33 ! | 45 - | 57 9 | 69 E | 81 Q | 93] | 105 i | 117 u | | |
| 34 " | 46 . | 58 : | 70 F | 82 R | 94 ^ | 106 j | 118 v | | |
| 35 # | 47 / | 59 ; | 71 G | 83 S | 95 _ | 107 k | 119 w | | |
| 36 \$ | 48 0 | 60 < | 72 H | 84 T | 96 ` | 108 l | 120 x | | |
| 37 % | 49 1 | 61 = | 73 I | 85 U | 97 a | 109 m | 121 y | | |
| 38 & | 50 2 | 62 > | 74 J | 86 V | 98 b | 110 n | 122 z | | |
| 39 ' | 51 3 | 63 ? | 75 K | 87 W | 99 c | 111 o | 123 { | | |
| 40 (| 52 4 | 64 @ | 76 L | 88 X | 100 d | 112 p | 124 | | |
| 41) | 53 5 | 65 A | 77 M | 89 Y | 101 e | 113 q | 125 } | | |
| 42 * | 54 6 | 66 B | 78 N | 90 Z | 102 f | 114 r | 126 ~ | | |
| 43 + | 55 7 | 67 C | 79 O | 91 [| 103 g | 115 s | 127 | | |

Erweiterter ASCII-Zeichensatz, MacOS / ANSI-Windows

| | | | | | | | |
|-----|-------|-----|-------|-----|--------|-----|--------|
| 128 | Ä 196 | 160 | † 134 | 192 | ı 191 | 224 | ‡ 135 |
| 129 | Å 197 | 161 | ° 176 | 193 | ı̇ 161 | 225 | · 183 |
| 130 | Ç 199 | 162 | ¢ 162 | 194 | ¬ 172 | 226 | , 130 |
| 131 | É 201 | 163 | £ 163 | 195 | √ 178 | 227 | „ 132 |
| 132 | Ñ 209 | 164 | § 167 | 196 | f 131 | 228 | ‰ 137 |
| 133 | Ö 214 | 165 | • 149 | 197 | ≈ 179 | 229 | À 194 |
| 134 | Ü 220 | 166 | ¶ 182 | 198 | Δ 185 | 230 | Ê 202 |
| 135 | á 225 | 167 | ß 223 | 199 | « 171 | 231 | Á 193 |
| 136 | à 224 | 168 | ® 174 | 200 | » 187 | 232 | Ë 203 |
| 137 | â 226 | 169 | © 169 | 201 | … 46 | 233 | È 200 |
| 138 | ä 228 | 170 | ™ 153 | 202 | ˆ 160 | 234 | Í 205 |
| 139 | ã 227 | 171 | ˘ 180 | 203 | À 192 | 235 | Î 206 |
| 140 | å 229 | 172 | ¨ 168 | 204 | Å 195 | 236 | Ï 207 |
| 141 | ç 231 | 173 | ≠ 164 | 205 | Ö 213 | 237 | Ì 204 |
| 142 | é 233 | 174 | Æ 198 | 206 | Œ 140 | 238 | Ó 211 |
| 143 | è 232 | 175 | Ø 216 | 207 | œ 156 | 239 | Ô 212 |
| 144 | ê 234 | 176 | ∞ 129 | 208 | – 150 | 240 | 🍏 173 |
| 145 | ë 235 | 177 | ± 177 | 209 | — 151 | 241 | Ò 210 |
| 146 | í 237 | 178 | ≤ 141 | 210 | “ 147 | 242 | Ú 218 |
| 147 | ì 236 | 179 | ≥ 142 | 211 | ” 148 | 243 | Û 219 |
| 148 | î 238 | 180 | ¥ 165 | 212 | ‘ 145 | 244 | Ü 217 |
| 149 | ï 239 | 181 | μ 181 | 213 | ’ 146 | 245 | ı̇ 208 |
| 150 | ñ 241 | 182 | ∂ 143 | 214 | ÷ 247 | 246 | ˆ 136 |
| 151 | ó 243 | 183 | Σ 144 | 215 | ◊ 190 | 247 | ˜ 152 |
| 152 | ò 242 | 184 | Π 222 | 216 | ÿ 255 | 248 | ˘ 175 |
| 153 | ô 244 | 185 | π 254 | 217 | ÿ̇ 159 | 249 | ˘ 188 |
| 154 | ö 246 | 186 | ∫ 157 | 218 | / 189 | 250 | ˘ 221 |
| 155 | õ 245 | 187 | ª 170 | 219 | € 128 | 251 | ° 215 |
| 156 | ú 250 | 188 | ° 186 | 220 | < 139 | 252 | ˘ 184 |
| 157 | ù 249 | 189 | Ω 166 | 221 | > 155 | 253 | ˘ 158 |
| 158 | û 251 | 190 | æ 230 | 222 | fi 138 | 254 | ˘ 240 |
| 159 | ü 252 | 191 | ø 248 | 223 | fl 154 | 255 | ˘ 253 |

Erweiterter ASCII-Zeichensatz, ANSI-Windows / MacOS

| | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|---|-----|-----|---|-----|
| 128 | € | 219 | 160 | 202 | 192 | À | 203 | 224 | à | 136 |
| 129 | ∞ | 176 | 161 | ı | 193 | Á | 231 | 225 | á | 135 |
| 130 | , | 226 | 162 | ¢ | 162 | Â | 229 | 226 | â | 137 |
| 131 | f | 196 | 163 | £ | 163 | Ã | 204 | 227 | ã | 139 |
| 132 | „ | 227 | 164 | ≠ | 173 | Ä | 128 | 228 | ä | 138 |
| 133 | ... | 201 | 165 | ¥ | 180 | Å | 129 | 229 | å | 140 |
| 134 | † | 160 | 166 | Ω | 189 | Æ | 174 | 230 | æ | 190 |
| 135 | ‡ | 224 | 167 | § | 164 | Ç | 130 | 231 | ç | 141 |
| 136 | ^ | 246 | 168 | ¨ | 172 | È | 233 | 232 | è | 143 |
| 137 | % | 228 | 169 | © | 169 | É | 131 | 233 | é | 142 |
| 138 | fi | 222 | 170 | ª | 187 | Ê | 230 | 234 | ê | 144 |
| 139 | < | 220 | 171 | « | 199 | Ë | 232 | 235 | ë | 145 |
| 140 | Œ | 206 | 172 | ¬ | 194 | Ì | 237 | 236 | ì | 147 |
| 141 | ≤ | 178 | 173 | Ⓐ | 240 | Í | 234 | 237 | í | 146 |
| 142 | ≥ | 179 | 174 | ® | 168 | Î | 235 | 238 | î | 148 |
| 143 | ð | 182 | 175 | ¯ | 248 | Ï | 236 | 239 | ï | 149 |
| 144 | ∑ | 183 | 176 | ° | 161 | ı | 245 | 240 | ı | 254 |
| 145 | ´ | 212 | 177 | ± | 177 | Ñ | 132 | 241 | ñ | 150 |
| 146 | ’ | 213 | 178 | √ | 195 | Ò | 241 | 242 | ò | 152 |
| 147 | “ | 210 | 179 | ≈ | 197 | Ó | 238 | 243 | ó | 151 |
| 148 | ” | 211 | 180 | ´ | 171 | Ô | 239 | 244 | ô | 153 |
| 149 | • | 165 | 181 | μ | 181 | Õ | 205 | 245 | õ | 155 |
| 150 | – | 208 | 182 | ¶ | 166 | Ö | 133 | 246 | ö | 154 |
| 151 | — | 209 | 183 | · | 225 | ° | 251 | 247 | ÷ | 214 |
| 152 | ~ | 247 | 184 | ¸ | 252 | Ø | 175 | 248 | ø | 191 |
| 153 | ™ | 170 | 185 | Δ | 198 | Ù | 244 | 249 | ù | 157 |
| 154 | fl | 223 | 186 | ° | 188 | Ú | 242 | 250 | ú | 156 |
| 155 | > | 221 | 187 | » | 200 | Û | 243 | 251 | û | 158 |
| 156 | œ | 207 | 188 | ˘ | 249 | Ü | 134 | 252 | ü | 159 |
| 157 | ∫ | 186 | 189 | / | 218 | · | 250 | 253 | ˘ | 255 |
| 158 | ˜ | 253 | 190 | ◊ | 215 | ∏ | 184 | 254 | π | 185 |
| 159 | ÿ | 217 | 191 | ı | 192 | ß | 167 | 255 | ÿ | 216 |

Nicht alle Zeichen werden in allen Schriften gleich dargestellt oder sind dort vorhanden. Die hier dargestellten Zeichen stammen aus einer Macintosh-Times.

