

MIDOS2000

Dokumentation der Grundfunktionen

Inhaltsverzeichnis

- 1 Was heißt MIDOS
- 2 Zur Installation von MIDOS
- 3 Erstellung einer neuen MIDOS-Datenbank
 - 3.1 Erstellung der Eingabebeschreibung
 - 3.2 Zuordnung der RAK-Felder
 - 3.3 Erfassung von Daten
- 4. Ausgeben von Daten
 - 4.1 Erstellung von Ausgabeformen
 - 4.2 Wiederfinden von Datensätzen
- 5. Aufbereitung von Anwendungen
 - 5.1 Retrieval-Anwendung
 - 5.2 HTML-Anwendung
 - 5.3 Perl-Anwendung
- 6. Anhang
 - 6.1 Anweisungen der Ausgabebeschreibung
 - 6.2 Kommandoübersicht von MIDOS-UPDATE
 - 6.3 EXEC-Anweisungen für die MIDOS-UPDATE-Feldmanipulation

1 Was heißt MIDOS

MIDOS heißt: Modulares Information- und DokumentationsSystem. Es beinhaltet Programme zur Erfassung, Pflege und Wiederfindung von Datensätzen zu beliebigen Dokumenten.

Das MIDOS-Format wird wie folgt charakterisiert:

Eine MIDOS-Datei ist eine Textdatei, die bis zu 2,1 GByte Zeichen umfassen kann.

Ein MIDOS-Datensatz besteht aus max. 250 Felder (=Zeilen), die mit den Zeichen &&& beendet werden. Ein Datensatz kann bis zu 500 000 Zeichen umfassen.

Ein Feld besteht aus dem Feldnamen mit max. 15 Zeichen und dem Feldinhalt mit max. 500 000 Zeichen. Mehrfachbelegungen z.B. mehrere Autoren (in MIDOS als Aussagen bezeichnet) werden mit einem Senkrechtstrich getrennt.

Ein Feld besteht aus einem Kurzfeldname (max. 15 Zeichen), einem Doppelpunkt und dem Feldinhalt (max. 500000 Zeichen). Im MIDOS-Erfassungsprogramm wird die Langform des Feldnamens verwendet, die aber nicht gespeichert wird.

Der Feldinhalt kann aus beliebig viele Paralleleinträgen (=Aussagen) bestehen, die mit Senkrechtstrich getrennt sind.

Die MIDOS-Datensätze werden variabel dh. ohne Füllzeichen gespeichert. Dadurch ergeben sich sehr speicherökonomische Dateien.

Die Datensatzstruktur wird durch eine Eingabebeschreibung festgelegt, die in der Datei MISCHABS gespeichert ist. Das Ausgabelayout für Listen kann aus max. 20 Ausgabebeschreibungen gewählt werden:

MIDOS2000 / Hilfsprogramme

Datenbeschreibung ändern / erstellen

C:\MIDOS97\UMWELT\MISCHABS

Feldtypen: A ... beliebiger Text (max. 30000 Zeichen)

Feld	Feldname	ausführl. Benennung	Felderläuterung	Typ	Länge
1	EN	Laufende Nummer	Wird automatisch vergeben	N4	20
2	ART	Dokumentenart	Charakterisierung des Doku	A	20
3	WOR	Verfasser Vorlage	Personenname von Autoren	A	200
4	VERF	Verfasser Ansetzung	Personenname in Ansetzung	A	200
5	ARB	Arbeitsstelle Persönl. Verf.	Körperschaft, bei der der Ve	A	200
6	VFKVO	Korp. Verf. Vorlage	Name der Körperschaft in Vo	A	200
7	VFK	Korp. Verf. Ansetzung	Name der Körperschaft in An	A	200
8	SKO	Sitz der Körperschaft	Offizieller Sitz der Körpersch	A	200
9	TIT	Sachtitel	Benennung des Dokuments	A	300
10	STU	Aufsatztitel	Titel des Aufsatzes	A	300

Info: zur aktuellen Tabellenspalte

Lade Vorlage Drucken Alles löschen Zeile löschen Zeile einfügen OK Abbruch

Beispiel einer Eingabebeschreibung

Gespeichert wird nur der Feldname, Doppelpunkt und der Feldinhalt. Die ausführliche Benennung (auch in MIDOS als Langform des Feldnamens bezeichnet) wird nur in Eingabe- und Ausgabemasken eingeblendet. Die Felderläuterung ist nur in der Eingabemaske mit der Hilfetaste F1 abfragbar. Als Typ wird vorwiegend A für beliebige Texteingaben verwendet. Es sind aber auch weitere Typen definierbar:

A ... beliebiger Text (max. 30000 Zeichen)

N ... Zahl (allgemein)

Nxy ... Zahl (mit x Vorkomma-/ y Dezimal-Stellen)

D ... Datum* (11.11.00)

D1 ... Datum (11.11.2000)

D2 ... Datum* (00.11.11)

D3 ... Datum (2000.11.11)

D4 ... Datum* (11.11.00, automatisch)

D5 ... Datum (11.11.2000, dto.)

D6 ... Datum* (00.11.11, dto.)

D7 ... Datum (2000.11.11, dto.)

D8 ... Datum* (11.11.00, autom. bei Änderung)

D9 ... Datum (11.11.2000, dto.)

D10 ... Datum* (00.11.11, dto.)

D11 ... Datum (2000.11.11, dto.)

Z2 ... Zeit (hh:mm)

Z3 ... Zeit (hh:mm:ss)

*) wegen 2 stelliger Jahreszahl möglichst nicht ver

Da MIDOS nur die wirklich erfassten Zeichen speichert kann der Zahlenwert für (Feld-) Länge theoretisch auf Maximum (500000 Zeichen) gesetzt werden.

Der Zahlenwert für Länge sollte aber trotzdem ungefähr der max. vorkommenden Feldlänge entsprechen, da die Gestaltung der Eingabemaske davon abhängt. Bis zu einer Feldlänge von 60 Zeichen werden variable einzeilige Fenster und darüber 3 zeilige Fenster mit Scrollbalken für die Eingabe der Feldinhalte generiert

In MIDOS2000 hat die Anzahl der Aussagen (=Mehrfachbelegungen) keine Bedeutung mehr. Generell sind beliebig viele Aussagen speicherbar und in Ausgabeformen ausgebaut.

So sehen Daten im Speicherformat aus:

(wenn die Datei z.B. mit WordPad geöffnet wird)

EN:0001
ART:Mehrbändiges Werk
VVOR:Gabriele Schwarz
VERF:Schwarz, Gabriele
TIT:Allgemeine Siedlungsgeographie
GT:Lehrbuch der Allgemeinen Geographie ; Bd. 6
BD:Teil 1. Die ländlichen Siedlungen. Die zwischen Land und Stadt stehenden Siedlungen. - 481, XLIII S. : Ill.
^ISBN 3-11-007895-3 ^div. Lit. | Teil 2. Die Städte. - 1089, LX S. : Ill. ^ISBN 3-11-011019-9 ^div. Lit.
AG:4. Aufl.
VO:Berlin ; New York
VE:de Gruyter
EJ:1988
SPR:D
VB:Landesumweltamt Brandenburg, Landesumweltbibliothek, Präsenzbestand
STO:Landesumweltamt Brandenburg, Landesumweltbibliothek
SIGN:RA 30 18/2 | RA 30 19/2
NOT:RA 30
DSK:Siedlungsgeographie | Siedlungsstruktur | Dorf | Gewerbefläche | Industrieansiedlung | Siedlungsentwicklung |
Stadtstruktur | Stadtgeschichte | Ballungsgebiet
EDAT:10.03.97
ENAME:JK
&&&

EN:0002
ART:Band/Schriftenreihe
VVOR:Herbert Bartels, Burmeister, Rüdiger Erdmann, Karl-Heinz Lins, Erich Möller, Karl Neese
VERF:Bartels, Herbert ...
TIT:Starkstromanlagen in baulichen Anlagen für Menschenansammlungen : Erläuterungen zu DIN VDE 01
08
GT:VDE-Schriftenreihe ; Bd. 61
VO:Berlin ; Offenbach
VE:vde-verlag
EJ:1993
SEI:152 S.
ISSN:0506-6719
ISBN:3-8007-1870-7
SPR:D
VB:Landesumweltamt Brandenburg, Landesumweltbibliothek, Präsenzbestand
STO:Landesumweltamt Brandenburg, Landesumweltbibliothek
SIGN:TE 35-4 /2
NOT:TE35
DSK:Elektrizität | Gebäudetechnik | Schutzvorschrift | DIN-Norm | Anlagensicherheit | Öffentliches Gebäude
EDAT:10.03.97
ENAME:JK
&&&

Was man aus diesen Daten machen kann zeigen folgende Ausgabebeispiele:Liste im RAK-Format

Schwarz, Gabriele:
Allgemeine Siedlungsgeographie / Gabriele Schwarz. - 4. Aufl. - Berlin ; New York : de Gruyter, 1988
(Lehrbuch der Allgemeinen Geographie ; Bd. 6)
Teil 1. Die ländlichen Siedlungen. Die zwischen Land und Stadt stehenden Siedlungen. - 481, XLIII S. : Ill. ISBN
3-11-007895-3 div. Lit.
Teil 2. Die Städte. - 1089, LX S. : Ill. ISBN 3-11-011019-9 div. Lit.

Normalliste

Laufende Nummer	0001
Dokumentenart	Mehrbändiges Werk
Verfasser Vorlage	Gabriele Schwarz
Verfasser Ansetzung	Schwarz, Gabriele
Sachtitel	Allgemeine Siedlungsgeographie
Gesamttitle	Lehrbuch der Allgemeinen Geographie ; Bd. 6
Band	Teil 1. Die ländlichen Siedlungen. Die zwischen Land und Stadt stehenden Siedlungen. - 481, XLIII S. : Ill. ISBN 3-11-007895-3 div. Lit.; Teil 2. Die Städte. - 1089, LX S. : Ill. ISBN 3-11-011019-9 div. Lit.
Ausgabe	4. Aufl.
Verlagsort	Berlin ; New York
Verlag	de Gruyter
Erscheinungsjahr	1988
Sprache	D
Verfügbarkeit	Landesumweltamt Brandenburg, Landesumweltbibliothek, Präsenzbestand
Standort	Landesumweltamt Brandenburg, Landesumweltbibliothek
Signatur	RA 30 18/2; RA 30 19/2
Klassifizierungsnotation	RA 30
Deskriptoren	Siedlungsgeographie; Siedlungsstruktur; Dorf; Gewerbefläche; Industrieansiedlung; Siedlungsentwicklung; Stadtstruktur; Stadtgeschichte; Ballungsgebiet
Erfassungsdatum	10.03.97
Erfasst von	JK

Tabelle

lfd. Nr.	Dokumentenart	Verfasser Ansetzung	Sachtitel	Gesamttitle	Band
1	Mehrbändiges Werk	Schwarz, Gabriele	Allgemeine Siedlungsgeographie	Lehrbuch der Allgemeinen Geographie ; Bd. 6	Teil 1. Die ländlichen Siedlungen. Die zwischen Land und Stadt stehenden Siedlungen. - 481, XLIII S. : Ill. ISBN 3-11-007895-3 div. Lit.; Teil 2. Die Städte. - 1089, LX S. : Ill. ISBN 3-11-011019-9 div. Lit.
2	Band/Schriftenreihe	Bartels, Herbert ...	Starkstromanlagen in baulichen Anlagen für Menschenansammlungen : Erläuterungen zu DIN VDE 01 08	VDE-Schriftenreihe ; Bd. 61	

2 Zur Installation von MIDOS

MIDOS wird mit dem Installationsprogramm SETUP.EXE der CD installiert. Dabei ist ein Programmverzeichnis zu vereinbaren. Bei einer Netzinstallation wird automatisch ein örtliches Verzeichnis C:\M2000 angelegt, das nur die INI-Dateien M2000.INI, M2001.INI, M2002.INI, ... enthält. Die Datei M2000.INI enthält die Grundeinstellungen von MIDOS2000, die Datei M20001.INI alle angemeldete MIDOS-Datenbanken der Liste 1 (Buch1), die Datei M2002.INI alle angemeldete MIDOS-Datenbanken der Liste 2 (Buch2), usw.

3 Erstellung einer neuen MIDOS-Datenbank

3.1 Erstellung der Eingabebeschreibung

Eine neue MIDOS-Datenbank wird mit der Funktion des Grundmenüs „Datenbank“ / „Neu“ eingerichtet. Danach sind:

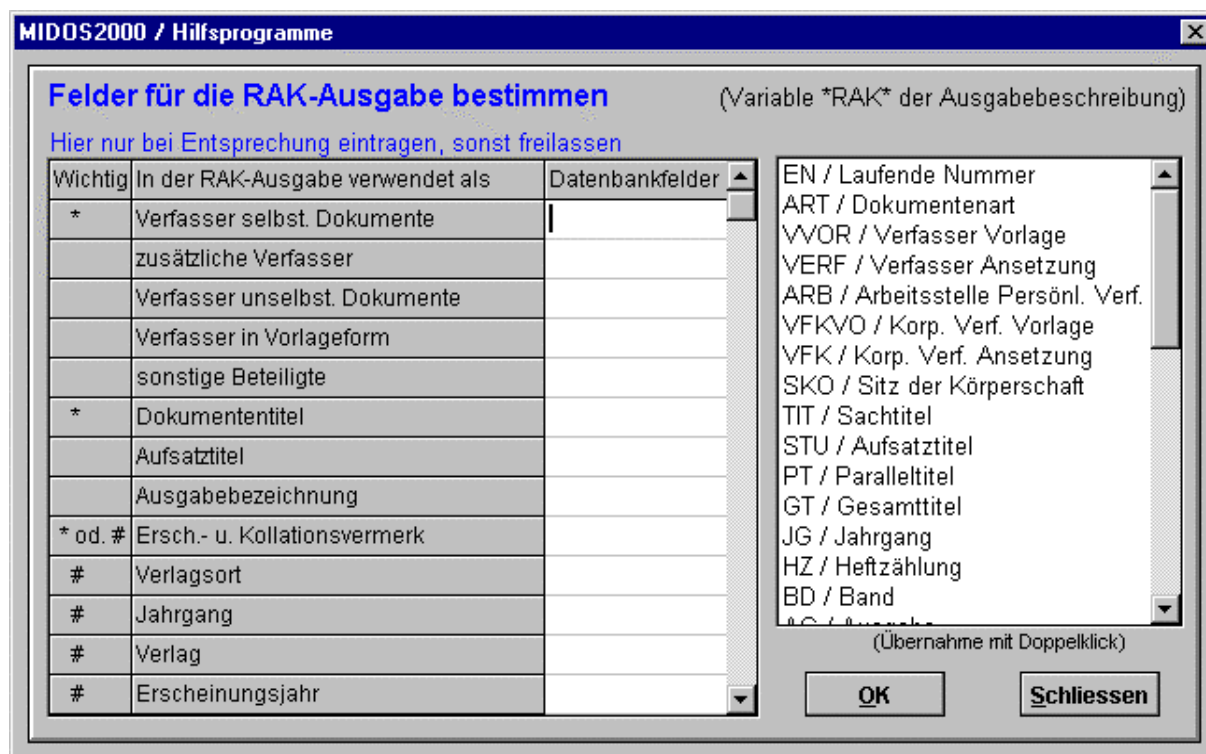
- der Titel der Datenbank (allg. Bezeichnung, max. ca. 50 Zeichen)
- der Dateiname der Datenbankdatei (max. 8 Zeichen)
- die Datenbankfelder mit jeweils:
 - o dem Feldnamen (keine Umlaute, keine Leerzeichen, kein Unterstrichstrich, max. 15 Zeichen)
 - o der ausführlichen Feldbezeichnung (max. ca. 40 Zeichen)
 - o wahlweise der Felderläuterungen (max. 160 Zeichen)
 - o dem Feldtyp (siehe Typenliste oben)
 - o der Feldlänge (max 500000 Zeichen)

eingeben.

Als Eingabehilfe werden nach der Eingabe eines Feldnamens und der Pfeiltaste nach unten Standardwerte eingetragen (Feldbezeichnung=Feldname, Typ=A, Länge=32000).

3.2 Zuordnung der RAK-Felder

MIDOS enthält einen RAK-Generator, der für Ausgabefunktionen ein Literaturstellenformat nach den Regeln der allgemeinen Katalogisierung erzeugt. Dazu sind aber die Felder der Eingabebeschreibung bestimmten RAK-Kategorien zuzuordnen. Diese Zuordnung erfolgt mit der Funktion „Bearbeiten“ / „RAK-Felder definieren“:



Die Zuordnung erfolgt mit Doppelklick.

In der Ausgabe wird das RAK-Format mit der Variablen <stern>RAK<stern> aufgerufen.

3.3 Erfassung von Daten

Das Erfassungsprogramm wird im Menü „Bearbeiten“ / „Datensatz ändern“ oder mit dem Bleistiftbutton in der rechten Menüliste des Grundmenüs aufgerufen.

Vorher kann der Datensatz in der Kurzanzeige markiert werden, der geändert werden soll. Bei einer leeren Datenbankdatei wird im Erfassungsprogramm sofort die „Neueingabe“ der Daten aufgerufen, sonst kann der angewählte Datensatz editiert werden. Im Editiermodus ist es möglich mit dem Button "Neu" die Erfassung neuer Datensätze aufzurufen.

Füllen Sie jetzt alle Felder aus. Mehrfacheinträge trennen Sie bitte mit dem Senkrechtstrich (=Taste F2 oder Button "x|y").

Mit den Buttons "1", "3", "10" oder "20" können Sie die sichtbare Fensterhöhe des aktuellen Feldes auf die entsprechende Zeilenzahl setzen. Der Button „H“ lässt das aktuelle Feld „verschwinden“. Es wird nur im Hintergrund mitgeführt und kann nicht verändert werden. Ein Rücksetzen dieser Funktionen ist mit dem Button "0" möglich.

Ein Feldwechsel erfolgt mit den Cursortasten oder der ENTER-Taste. Das Betätigen der Enter-Taste im letztem Feld bei der Neueingabe bewirkt nach einer Abfrage zur Speicherung des aktuellen Satzes das Initialisieren einer leeren Eingabemaske zur Eingabe des nächsten Datensatzes. Das Gleiche erfolgt mit dem Button "Nächste" (oben links).

Einträge aus dem vorhergehenden Satz können Sie mit der Taste F12 holen. Mit der Taste F11 können Sie Feldänderungen rückgängig machen.

Datensätze können mit dem "Del"-Button gelöscht werden. Das Einfügen von leeren Datensätzen vor dem aktuellen Satz erfolgt mit der Funktion "Bearbeiten"-„Satz einfügen".

Die Eingabe von neuen Datensätzen beenden Sie mit dem Button "Beenden" im Fenster Neuerfassung. Danach werden bei Bestätigung der Abfrage zur Speicherung die Datensätze an der Datenbank angehängt. Jetzt befinden Sie sich im Korrekturmodus des Editors und können auch die neu eingegebenen Datensätze ändern.

Im Korrekturmodus können Sie auch mit den Tasten Pos1/Ende zwischen den Datensätzen wechseln.

Das Erfassungsprogramm bietet eine Reihe von Funktionen zur Textübernahme und zur Datenkontrolle:

Textübernahme

Masken (auch Schablonen genannt)

MIDOS bietet die Möglichkeit, über teilausgefüllte Datensätze bestimmte Feldinhalte bei der Erfassung bereitzustellen. Die Erfassung der vorgelegten Texte erfolgt im Menü „Bearbeiten“ / „Maske (Schablone)“. Die Masken sind dann unter einem beliebigen Namen zu speichern. Nach Verlassen des Maskenerfassungsmodens kann die Maske mit einer Auswahlliste aufgerufen werden:



Es sind auch mehrere Masken gleichzeitig ladbar. Dabei werden die belegten Felder kumuliert, aber nicht die Feldinhalte. Das heißt, dass bei der Belegung des gleichen Feldes von unterschiedlichen Masken die zuletzt geladene Maske die ausschlaggebende ist.

Ablage

- markierten Text in die Ablage kopieren: CTRL + EINF

- markierten Text zur Cursorstelle kopieren: SHIFT + EINFÜG



Mit diesem Buttonblock kann (von oben links)

- das aktuelle Feld vollständig zur Ablage kopiert werden
- der gesamte Datensatz zur Ablage kopiert werden
- Die Ablage selber editiert werden
- Einzelfelder zurückkopiert werden (auch wenn vorher ein ganzer Datensatz in die Ablage kopiert wurde)
- Oder der gesamte Datensatz zurückkopiert werden.

Textbausteine

Als Textbausteine sind hier nur max. 255 Zeichen lange Sätze zu verstehen, die nach Eingabe eines kurzen Aufrufnamens und anschließender Betätigung der Taste F10 eingefügt werden. Die Textbausteine werden in einer Tabelle definiert, die mit der Funktion CTRL-F10 aufgerufen werden kann.

Aufrufname	Inhalt des Textbausteines (max. 255 Zeichen)
MFG	Mit freundlichen Grüßen
P	Progris GmbH Berlin

Die Textbausteine können Sie auch aufrufen, indem Sie im Erfassungstext den Aufrufnamen eintragen und unmittelbar danach die Taste F10 drücken.

Speichern Einfügen aktuellen Textbaustein übernehmen
Sortieren Löschen Schliessen

Die Textbausteine sind mit Aufrufname und Inhalt zu erfassen und dann in eine beliebige Datei zu speichern.

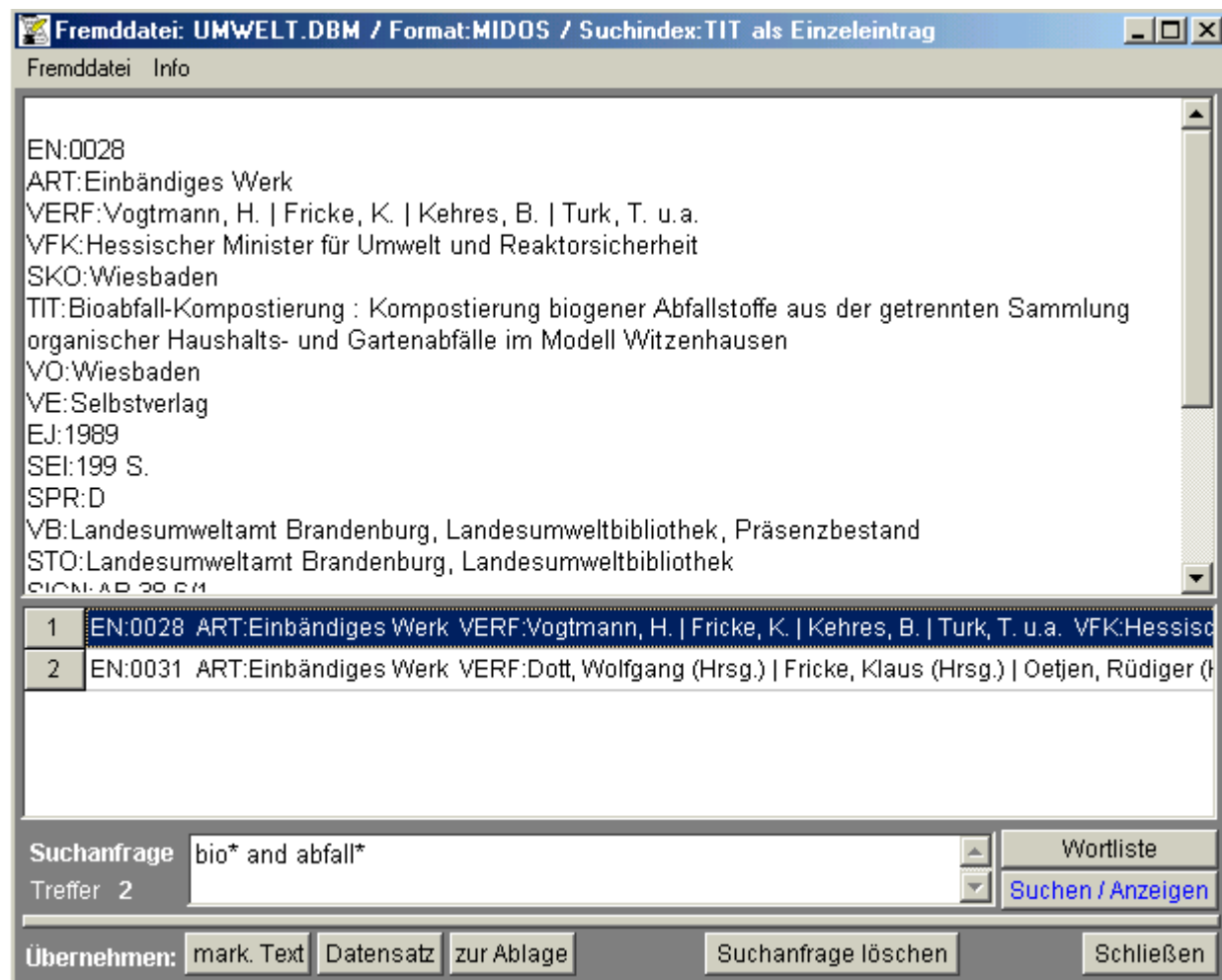
Fremdtext/Fremddatensätze

Mit dem Button „Fremd“ ist über ein Menü möglich Datensätze aus fremden MIDOS-Dateien und Texte aus fremden Textdateien zu kopieren.

Eine MIDOS-Datenbank wird dort mit der Funktion „Fremddatei“/„Öffnen“ vereinbart. Dann ist das Format auf „MIDOS“ einzustellen und der MIDOS-Dateiname einzutragen.

Aus der Liste „Felder holen“ sind dann die Indexfelder (=Suchfelder) mit Doppelklick auszuwählen. Diese Felder können dann in „Suchanfrage“ abgefragt werden, oder mit „Wortliste“ angezeigt werden. Wie der Feldinhalt in der Abfrageliste gespeichert werden soll, ist mit der Checkbox „Index mit Einzelwortauflösung“ einzustellen. Ist diese Checkbox markiert, dann steht jedes Wort des Feldes einzeln in der Liste und kann auch einzeln abgefragt werden. Sonst werden max. 250 Zeichen des Feldes in der Liste aufgenommen und können nur als Ganzes direkt abgefragt werden. Eine sequentielle Abfrage jedes Einzelwortes und Teilwortes ist im Wortlistenfenster mit den „Lupen“-Buttons in allen Fällen möglich.

Die Suchwörter werden durch Übernahme mit Doppelklick aus der Wortliste oder durch Eingabe in das Feld „Suchanfrage“ geschrieben. Nach Klicken auf Button „Suche/Anzeige“ werden die Datensätze im Speicherformat als Kurz- und Vollanzeige dargestellt:



Der gerade aktuelle Datensatz der Vollanzeige kann nun mit dem Button „Datensatz“ in die aktuelle Erfassungsmaske übernommen werden. Bedingung dafür ist immer, dass gleiche Feldnamen vorliegen. Eine Anpassung von Feldern, Filtern von Zeichenketten usw. ist mit der Funktion „Fremddatei/„Anpassen“ möglich.

Weiterhin kann auch bestimmter Text markiert werden. Dieser Text kann dann durch mit dem Button „mark. Text“ oder mit den üblichen WINDOWS-Funktionen in das aktuelle Feld kopiert werden

Texte suchen (und automatisch ersetzen)

Mit dem Button „<Lupe>“ oder „weiter“ wird das Suchmenü aufgerufen:



Hier kann entsprechend der Suchfeldeinstellung ein beliebiges Wort im Datensatz gesucht werden und ggf. durch den Eintrag im Feld „ersetzen durch“ ersetzt werden.

Mit dem Button „Suche ab Satz1“ wird der Suchzeiger auf Satz 1 gesetzt, sonst wird immer weiter gesucht.

Der Schalter „Nur aktuelles Feld verwenden“ bewirkt, dass eine automatische Such- und Ersetzfunktion aktiviert wird und der Vorgang sich nur auf das aktuelle Feld bezieht.

Automatische Schlagwortvergabe

Mit dem Button „aut. Schlagwort“ oder Taste F6 starten Sie die automatische Schlagwortvergabe. Voraussetzung ist folgende Parametereinstellung (Aufruf: „Optionen“ / „AutoSchlagwort“):

- Positivliste und/oder Synonymliste: WTX- oder Text-Datei; Textdatei hat zeilenweise folgenden Aufbau: <Deskriptor>;<Synonym1>;<Synonym2>; ... max 1 Mbyte Umfang je Datei
- Suchfelder: ein oder mehrere Textfelder (Titel, Referat) für die Wortanalyse
- Zielfeld: Feld für das Ergebnis der Analyse (vorhandene Einträge werden berücksichtigt)
- Wortlängenänderung: 0 bis 9 Zeichen für Längenänderungsmanipulation; je mehr Zeichen desto länger dauert die Analyse; Richtwert:2
- Schalter „Sortieren“: Ergebnis wird alphabetisch sortiert
- Wortanalyse: Ein- bis Dreiwortanalyse; richtet sich nach der verwendeten Positivliste; wenn diese nur max. 2 Wortpaare beinhaltet, dann ist auch nur eine Zweiwortanalyse sinnvoll
- Schalter „Ä wie A“ :Sonderzeichen werden bei der Analyse auf Ihre Grundzeichen zurückgesetzt.

Weitere Funktionen des Erfassungsprogramms

- Rechtschreibprüfung des aktuellen Feldes mit F5
- Erzeugung einer Wortliste zum aktuellem Feld mit „Funktionen“ / „Wortliste erzeugen“
- Prüfung des aktuellen Feldes mit der Wortliste mit F7
- Wortliste anzeigen zum aktuellem Feld anzeigen und Listebegriffe übernehmen mit F4
- Datensätze drucken mit CTRL D
- Automatische Nummerierung
- Feld sortieren (Mehrfachbelegungen wie mehrere Autoren oder Deskriptoren) mit „Bearbeiten“ / „Feld sortieren“
- Thesaurusmodul
- Programmierte Datensatzprüfung

Ein weiteres Programm zur Korrektur und Eingabe von Datensätzen steht mit dem **Tabelleneditor** zur Verfügung. Mit diesem Programm können vorwiegend Datensätze mit kurzen Feldinhalten tabellarisch übersichtlich behandelt werden. Trotzdem sind Datenfelder mit bis zu 32000 Zeichen Feldinhalt editierbar. Sämtliche Änderungen können durch Nichtspeicherung verworfen werden.

Das Programm wird mit dem Button „Tabelleneditor“ der rechten Menüleiste aufgerufen. Folgende Funktionen stehen im Tabelleneditor zur Verfügung:

- Navigieren zwischen Feldern und Datensätzen mit den Pfeiltasten
- Feld editieren mit F2 oder Mausklick
- Suchen und Ersetzen mit F3 (wahlweise mit Schnellersetzfunktion „Quick“ über MWUPDATE-Job)
- Wortliste aufrufen mit F4 (Wortliste <feldname>@.wtx muss existieren)
- Thesaurus aufrufen mit F9 (Thesaurus <name>.mth muss existieren)
- Großes Editierfenster für aktuelles Feld aufrufen mit F12 oder Doppelklick oder Button Vollfeldanzeige
- Feldinhalt löschen mit dem „Del“-Button
- Feld mit konstantem Text oder mit einem anderen Feldinhalt füllen mit dem „Text“-Button
- Datensatz löschen und Datensatzeinfügen mit den gleichnamigen Buttons
- Datensatz anfügen mit der ENTER-Taste oder mit dem gleichnamigen Button
- Editor beenden mit der Esc-Taste

4. Ausgeben von Daten

Mit maximal gleichzeitig verfügbaren 20 Ausgabeformaten (Reports), die mit einem internen Editor jederzeit änderbar sind, können aus den Datensätzen Direktausgaben in ansprechender Form über den Drucker, HTML-Dateien mit oder ohne JavaScript-Suchmaschine oder RTF-Dateien erzeugt werden. Damit sind interaktive wie auch klassische, gedruckte Informationsdienstleistungen herstellbar.

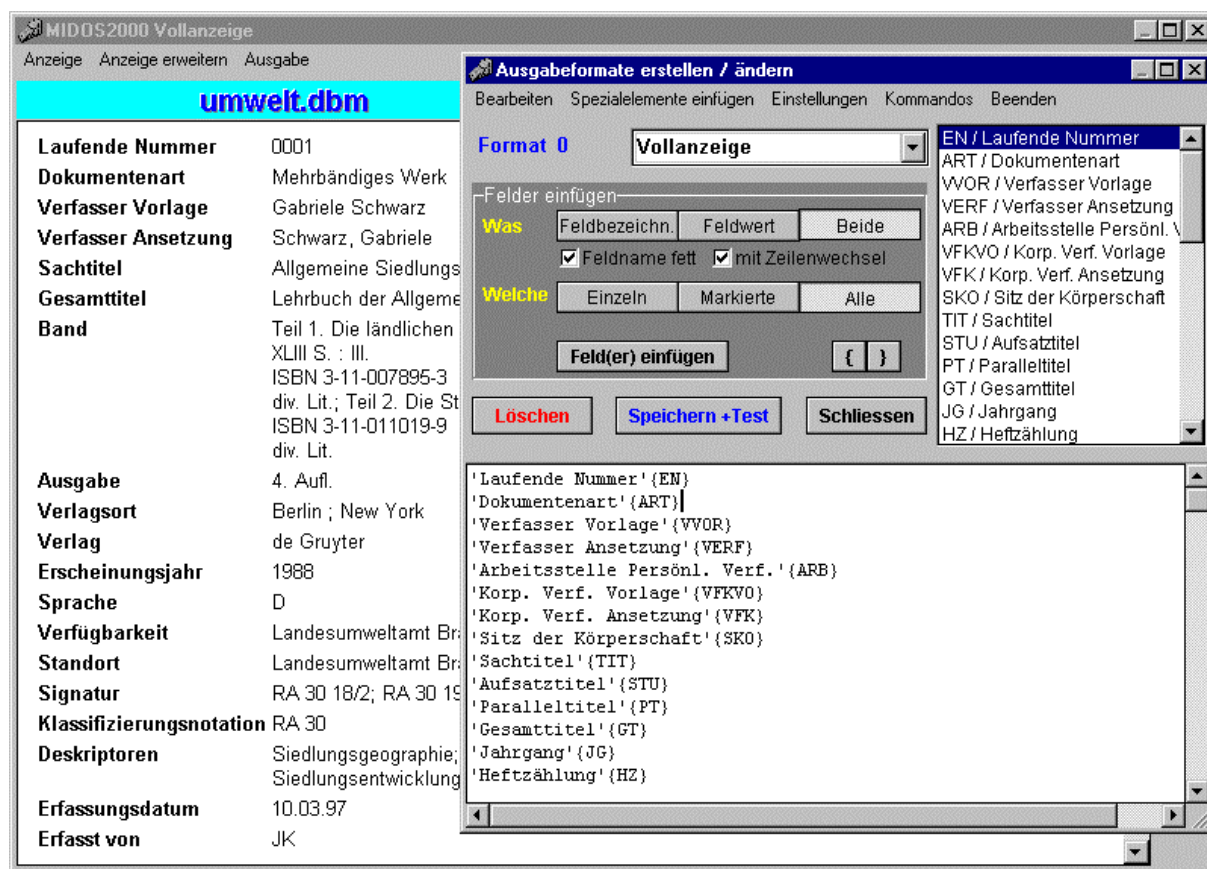
4.1 Erstellung von Ausgabeformaten

Eine Ausgabebeschreibung enthält die Vorschriften für die Anordnung von Felder, Texten, Links zu externen Dateien (HTM, PDF, Grafiken, WORD uvm.) sowie Kommandos zur Steuerung der Ausgabe mit bedingten Anweisungen, Schleifenanweisungen und Sprungbefehlen.

Ausgabebeschreibungen können u.a. in den Funktionen Vollanzeige, Bibliografie und JavaScript-Suchmaschine über ihren Ausgabeformatnamen ausgewählt werden. In MIDOS-Update kann eine Ausgabebeschreibung in der Funktion AFORMAT verwendet werden.

Aufruf des Editors zum Ändern der Ausgabeformate

Der Editor ist in der im Grundmenü mit „Bearbeiten“ / „Ausgabeformate“ und in der Vollanzeige mit „Anzeige“ / „Ändern“ aufrufbar.



Eine MIDOS-Ausgabebeschreibung kann einfache MIDOS-Anweisungen und auch HTML-Anweisungen bis zur HTML-Version 3 enthalten.

Die einfachste Art Felder auszugeben ist folgende:

Beispiel: **Sachtitel:{TIT}**

Innerhalb der geschweiften Klammer muss ein MIDOS-Feldname stehen. Ab der Position der öffnenden geschweiften Klammer wird dann der Feldinhalt des angegebenen Feldes dargestellt. Der Feldbezeichner kann noch in Hochkomma stehen, dann wird dieser fett gedruckt und alle folgenden Felder tabellarisch formatiert. *Hochkomma darf nur in dieser Form einmal pro Zeile angewendet werden.* Sonst ist für Fettmarkierung `` zu verwenden.

Eine schnelle Form alle Felder der Datenbank darzustellen ist folgende:

- Ausgabebeschreibungseditor aufrufen und Grundeinstellung nicht ändern
- Button „Löschen“ klicken (aktuelle Ausgabebeschreibung wird gelöscht)
- Button „Felder(er) einfügen“ klicken (alle Felder der Datenbank werden in das Editorfeld kopiert)

- Button „Speichern+Test“ klicken (Ausgabebeschreibung wird gespeichert und dann in der Vollanzeige mit Daten angezeigt)

Für spezielle Aufbereitungsformen stehen im Anhang die hauptsächlichsten Anweisungen der MIDOS-Ausgabebeschreibung.

4.2 Wiederfindung von Datensätzen

Für die Wiederfindung von Datensätzen stehen folgende Retrieval-Funktionen zur Verfügung

- Volltextrecherche
- Indexrecherche

Mit der **Volltextrecherche** ist grundsätzlich jedes Wort oder Teilwort des Datensatzes such- und findbar *ohne dass vorher ein Index generiert werden muss*. Die Suchwörter brauchen dabei nicht trunkiert eingegeben werden.

Mit der **Indexrecherche** ist nur jeder Eintrag der Indexliste suchbar. Die Suche geht hier aber sehr schnell. Der hierzu notwendige Index wird bei MIDOS2000 nach einer Datenänderung automatisch aktualisiert. Zur Teilwortsuche muss eine Trunkierung mit „*“ erfolgen.

Die Suchwörter können auch einem Thesaurus oder einer Suchliste mit Doppelklick entnommen werden.

Einige Möglichkeiten der Volltextrecherche werden an Beispielen aufgezeigt:

Einstellung Feldauswahl	eingegebene Suchfrage	Erläuterung
alle Felder	Gewinde	Diese Suchanfrage findet Datensätze, die "Gewinde" als Einzelbegriff oder auch Begriffe wie "Gewindeenden" oder "Außengewinde" in beliebigen Feldern enthalten. Wenn nur vollständige Begriffe gesucht werden sollen, dann ist der Button "Vollbegriff" vor dem Abschicken der Suchfrage zu klicken.
alle Felder	Gewinde and metrisch not Trapezgewinde	Die Begriffe können auch miteinander mit den Suchoperatoren OR , AND und NOT verbunden werden. Diese Suchanfrage schränkt die obere Suchfrage auf "metrisch" ein und schließt auch "Trapezgewinde" aus.
T / Titel	Gewinde	Hier wird der Begriff "Gewinde" nur im Feld Titel gesucht.
alle Felder	T=Gewinde or R=Gewinde	Auch kann die Suche auf mehrere einzelne Felder ausgedehnt werden. Hier wird der Begriff "Gewinde" nur in den Feldern T (Titel) und R (Registerdeskriptoren) gesucht. Als Feldname ist hier nur der Kurzfeldname anzugeben, der aus der Feldliste entnommen werden kann. Die Feldliste ist auf "in allen Felder" einzustellen.
alle Felder	a>01.01.1995 and a<31.12.1997	Es besteht auch die Möglichkeit Größer-/Kleiner-Suche in Zahlen- und Datumsfeldern durchzuführen. Dabei ist die ausführliche Datumsform einzugeben. Die Feldliste ist hier auch auf "in allen Felder" einzustellen. Im Beispiel werden nur Dokumente gefunden deren Ausgabedatum (Kurzfeldname:A) größer 1.1.95 und kleiner 31.12.97 ist.
beliebig	Gewinde and [trapez or metrisch]	Weiterhin kann eine Klammerebene mit den Symbolen "[]" verwendet werden. Im Beispiel werden nur Dokumente gefunden die "Trapez"+"Gewinde" oder "Metrisch"+"Gewinde" enthalten.

Zur **Ausgabe der Trefferlisten** stehen zahlreiche Funktionen zur Verfügung:

- HTML-Ausgabe der Vollanzeige mit Formulardruck
- Tabellenausgabe der Kurzanzeige
- Listenausgabe der Kurzanzeige

- Bibliografie im HTML- und RTF-Format über Menü „Funktionen“ / „Bibliografie“
- Statistische Aufbereitungen über Menü „Ausgabe“ / „Statistik“
- Dateiausgabe im HTML-, RTF-, Text-, Delimeted- oder MIDOS-Format

5. Aufbereitung von Anwendungen

5.1 Retrieval-Anwendung

Aufruf: Das Programm wird im Menü „Funktionen“ Punkt „Retrieval-Anwendung“ aufgerufen.

Funktionen des Programms

In einem Zielverzeichnis wird eine geschlossene Retrieval-Anwendung generiert, die unter WINDOWS-kompatiblen Betriebssystemen auch auf CD lauffähig ist.

Merkmale der Anwendung:

- Anwendung ohne Installation direkt startfähig
- Zusätzliche Installationfunktion für Programmgruppe, Desktop, Programmkopieren und Netzwerk vorhanden, sowie Deinstallationsprogramm
- Retrieval über große Datenmengen
- Feste Menü- und Kopffenster bei FRAME-Option
- Suchfragefenster mit bis zu 20 Suchfeldern mit Einzelwort und Volltextfunktion
- variabel einstellbare Trefferliste mit Lesezeichen, Notizen und Bestellung (max. 999 sach- oder personengebundene Lesezeichengruppen)
- Bestellung mit Preview, Direktdruck, E-Mail, WORD-Datei sowie Adressdatenbank)
- HTML-basierende Vollanzeige der Datensätze mit
 - direkten Internet-Links und E-Mails sowie lokale HTML-Links
 - Links zu beliebigen externen Dateien (zB. PDF,DOC,..) mit Aufruf des Viewers
 - verlinkten Feldern untereinander (zB. Schlagwörter mit Vorgänger und Nachfolgerlink)
 - Links zwischen verschiedenen Felder mit gleichem Inhalt (zB. Ersetzt durch...)
 - Grafikdateien (BMP-,GIF-,JPG-Format, animierten Gifs)
 - Sekundärrecherchen über Feldinhalte (zB. alle Datensätze zu einem Deskriptor)
- HTML-basierende Druckfunktion der Datensätze
 - als Tabelle entsprechend der Trefferliste
 - als Liste entsprechend der Vollanzeige
 - mit einstellbarer Treffermenge und Seitenparametern
 - mit Preview
- Datenexport als HTML-, RTF-, Text-delimeted, Nur-Text- oder MIDOS-Datenbankformat

Bedienung

Dieses Programm ermöglicht eine einfache Bestimmung der Hauptparameter der Retrievalanwendung. Über Karteikarten haben Sie Zugang zu den Fenstern SUCHFRAGE, KURZANZEIGE, VOLLANZEIGE, FAME und ALLGEMEINE EINSTELLUNGEN. Notwendige Parameter, die nicht durch das Menü festgelegt wurden, werden als Script generiert, das im letzten Fenster WEITERE EINSTELLUNGEN (ALS SCRIPT) zugänglich ist. Aller Werte werden in der Dateien M2RECH.INI gespeichert.

Im Fenster SUCHFRAGE legen Sie die Suchfelder fest. Die Suchfelder können Sie aus Ihrer Datenbeschreibung auswählen. Eine Feldliste erhalten Sie, wenn Sie auf die Tabellenzeile doppelt klicken. Die Bezeichnungen der Suchfelder können Sie nachträglich von Hand ändern.

Für jedes Feld wird aus den Einträgen ein Index generiert. Wenn Sie „Einzelwortindex“ wählen, wird der Index wortweise aufgebaut; wählen Sie diese Option nicht, wird jeweils der gesamte Feldinhalt in die Indexliste geschrieben. Durch den Button „Volltextsuche“ wird ein Suchfeld erzeugt, das eine freie Suche über alle Felder erlaubt.

Im Fenster KURZANZEIGE legen Sie die tabellarische Anzeige Ihrer Suchergebnisse fest. Wie bei der Suchfrage erhalten Sie eine Feldliste, wenn Sie auf die Tabellenzeile doppelt klicken. Die Bezeichnungen der Suchfelder können Sie nachträglich von Hand ändern.

Im Fenster VOLLANZEIGE legen Sie fest, wie die vollständige Anzeige eines kompletten Datensatzes aussehen soll. Dazu wählen Sie Anzeigeformate aus, welche Sie vorher in MIDOS2000 definiert haben. Ausserdem können Sie Ihre Datensätze verlinken. Links zwischen einzelnen Datensätzen erlauben das Blättern über Pfeilen. Die Felder, die verlinkt werden sollen, werden durch Semikolon getrennt aufgezählt. Links zwischen verschiedenen

Feldern des selben Datensatzes werden unterstrichen. Diese Querlinks werden durch Gleichheitszeichen miteinander gekoppelt.

Im Fenster FRAME können Sie feste Menü- und Kopffenster festlegen und damit ein WEB-Layout simulieren. Diese Fenster basieren auf HTML-Dateien (MENU.HTM und KOPF.HTM), die mit Wortlistenlinks und Links auf externe Dateien belegt werden können. Die Dateien können auch per Hand geändert werden. Dann sind diese Dateien (wie auch andere per Hand geänderte oder kopierte Dateien) mit Schreibschutz zu belegen, um ein Überschreiben oder Löschen bei der nächsten Generierung zu vermeiden

Im Fenster ALLGEMEINE EINSTELLUNGEN können Sie die Gestaltung der Anwendung festlegen: Wenn Sie eine Anzeige mit Frame wählen, können Sie ein Firmenlogo einbinden. Ausserdem können Sie verschiedene Darstellungs-, Start- und Exportmöglichkeiten bestimmen.

Obligatorisch für die Erzeugung der Anwendung ist nur die Bestimmung eines Suchfeldes. Für Kurz- und Vollanzeige sowie für die allgemeinen Einstellungen werden Standardwerte eingesetzt, wenn spezielle Festlegungen fehlen.

Das Fenster WEITERE EINSTELLUNGEN (ALS SCRIPT) sollte anfangs nicht geändert werden. Es enthält alle Parameter, die nicht durch das Menü eingestellt werden. Dieses Script kann bei Einhaltung der Parametersyntax auch geändert werden. Mit dem Button RESET können die Standardeinstellungen wiederhergestellt werden.

Retrieval einstellen

Nr.	Bezeichnung der Suchfrage	Suchfelder	Hint	Spezielle Einstellungen
1	Titel	TIT		"EINZEL"
2	Untertitel	UT		"EINZEL"
3	Land	LAND		
4	Genre	GENRE		
5	Produktion	PROD		
6	Regie	R		
7	Buch	B		
8	Darsteller	D		
9	Volltextsuche	VOLLTEXT	Freie Suche über alle Felder	"VOLLTEXT"=@DATEN.DBM
10				

Suchfeldbestand: 00 (0 ... 90 Pixel) mit Thesaurustabelle: [] Einzelwortindex: [] Volltextsuche: []

Gesamte Anwendung erzeugen [] Nur Datenbankdateien aktualisieren [] Nur Einstellungen neu speichern [] Anwendung starten [] Schließen []

Ergebnis

Anwendung erzeugen

Wenn mindestens ein Suchfeld vereinbart wurde kann die Anwendung generiert und getestet werden:

Mit dem Button "Gesamte Anwendung erzeugen" wird

- das Zielverzeichnis erstellt. (Wenn es bereits existiert wird der Inhalt gelöscht)
- alle notwendigen Programmdateien in das Zielverzeichnis kopiert
- alle Hilfsdateien, die im Menü vereinbart wurden, kopiert
- die Datenbank-Indexlisten generiert
- ggf. die Datenbank verlinkt
- die Einstellungen in die Datei M2RECH.INI kopiert (Inhalt ist chiffriert)
- die Datenbank chiffriert

Mit dem Button "Nur Indexlisten erzeugen" wird nur die Datenbank aktualisiert.

Die Einstellungen und der sonstige Inhalt des Zielverzeichnisses bleiben erhalten.

Mit dem Button "Nur Einstellungen neu speichern" werden die Einstellungen in die Datei M2RECH.INI gespeichert. Die Daten der Datenbank und die anderen Dateien im Zielverzeichnis bleiben unverändert.

Wenn der Schalter "Sofort starten" gesetzt ist, wird bei allen Testfunktionen anschließend die Datenbank gestartet.

Anwendung vervielfältigen

Ihre Nutzer können auf Ihre Rechercheanwendung über ein Netzwerk zugreifen wenn Sie das komplette Verzeichnis ...\\M2RETANW in ein Verzeichnis kopieren, das allen Nutzern zugänglich ist.

Wenn Sie Ihre Rechercheanwendung auf CD-ROM verbreiten möchten empfehlen wir Ihnen folgende Schritte: Um Starten und Setup zu vereinfachen, ist es empfehlenswert, folgende Verzeichnisstruktur anzulegen.

```
..\CDROOT\  
    |- autorun.inf  
    |- setup.exe  
    |- start.exe  
..    \CDROOT\PROG
```

Kopieren Sie das Verzeichnis M2RETANW in das Verzeichnis PROG und die Dateien autorun.inf, setup.exe, start.exe aus dem Verzeichnis PROG eine Ebene höher. Sie entsprechen den Dateien m2setup.exe und m2start.exe und sind entsprechen nur im CDROOT-Verzeichnis umzubenennen. Schreiben Sie das Verzeichnis CDROOT auf CD-ROM.

5.2 HTML-Anwendung

Aufruf

Das Programm wird im Menü „Funktionen“ Punkt „HTML-Anwendung“ aufgerufen.

Funktionen des Programms

Das Programm generiert eine ausschließlich auf HTML und Javascript basierende Anwendung, die im Internet, lokal sowie auf CD lauffähig ist. Da alle Suchlisten sowie die Felder der Kurz- und Vollanzeige als JavaScriptfelder in den HTML-Seiten eingebunden sind, können - bedingt durch vertretbare Ladezeiten der HTML-Seiten - nur bis ca. 10 000 Dokumente verwendet werden.

Weitere Merkmale der Anwendung:

- Die Anwendung ist betriebssystemunabhängig. Es wird nur ein installierter Internetbrowser benötigt.
- Die Fenstergestaltung ist frameorientiert
- Suchfragefenster mit bis zu 26 Suchfeldern mit Indexlistefunktion
- Trefferliste mit änderbarer Anzeigeeinstellung Liste/Tabelle und Trefferzahl
- Bestellung mit E-Mail- und Brief- / Fax-Ausgabe
- Vollanzeige der Datensätze mit verlinkten Feldern untereinander und zwischen verschiedenen Feldern

Bedienung

Folgende Parameter müssen eingestellt werden:

- Datenbankdatei (automatisch von MIDOS2000 ausgefüllt)
- Titel der Anwendung
- Zielverzeichnis (automatisch von MIDOS2000 ausgefüllt)
- Suchanfragen: Nach Doppelklick auf die Eingabetabelle sind die Felder für die Suchanfrage aus einem Menüfenster auswählbar. Die Spalte „Zeilen“ wird nur ausgewertet, wenn der Schalter „Suchanfragen mehrzeilig gesetzt ist. Der Einzelwortindex (Stern eingeben) sollte nur über Textfelder (Titel,...) angewendet werden.
- Ausgabefelder der Trefferliste: Mit Doppelklick einige (bis ca. 5), kurze Felder auswählen, die das Dokument charakterisieren (z.B. Verfasser, Titel).
- Vollanzeigeanzeigeformat

Die weiteren Parameter dienen der Gestaltung der Anwendung und können bei Bedarf geändert werden:

- Verlinkte Felder: Mit Doppelklick Felder mit einem kontrollierten Wortvorrat (Klassifikation, Deskriptoren) auswählen. Diese Felder werden automatisch verlinkt. Die Einträge dieser Felder erhalten Linkpfeile, die mit dem nächsten und vorhergehenden Datensatz verbunden sind, der auch diese Einträge enthält.
- Dok./Datei: nur wirksam wenn der Schalter „Einzelsatzanzeige“ nicht gesetzt ist; bestimmt die Anzahl der Datensätze pro Datei für die Vollanzeige; sollte dann 20 betragen
- Windings-Linkpfeile: nur wirksam wenn der Schalter „mit Grafiklinks“ nicht gesetzt ist; bestimmt die Symbole für die Feldverlinkung
- Die Schalter „Startseite1“ und „Startseite2“ erstellen Dummy-Startseiten, die vom Anwender mit den Edit-Buttons änderbar sind.
- Die folgenden Schalter „mit Suchanfragen“, „mit Wortlisten“, mit „Übersichtslisten“, mit Stylesheets, mit BODY-Einstellung schalten die Karteikartenfunktionen ein/aus ohne das die dortigen Einträge gelöscht werden müssen. Der Schalter „mit Suchanfragen“ sollte immer gesetzt sein
- Schalter „Suchanfrage mehrzeilig“: Zeilen sind dann in der Tabelle einstellbar (3..5)
- Schalter „Suchfrage mit Formular“: die Eingabe der Suchfrage erhält ein fensterähnliches Design
- Schalter „Einzelsatzanzeige“: Die Datensatzvollanzeige erfolgt über in JavaScriptfelder gespeicherte Daten. Diese sind unabhängig von der Einstellung „Dok./Datei“ in 20 Datensätze pro Datei gespeichert.
- Schalter „mit Suchwortmarkierung“: Nur bei „Einzelsatzanzeige“ möglich, wenn die Suchfelder auch ausgegeben werden
- Schalter „Wortliste mit Trefferzahl>1“: In Wortlisten wird die Trefferzahl nur bei Treffern >1 in Klammern mit ausgegeben
- Schalter „Link Dokument“: In Wortlisten wird bei einem Treffer zur Vollanzeige und bei mehreren Treffern zur Trefferliste (Kurzanzeige) gewechselt.
- Schalter „mit Datensatznavigator“: Jeder Datensatz erhält in der Vollanzeige Linksymbole für den nächsten, vorhergehenden Datensatz und zur Trefferliste
- Schalter „mit back.gif/next.gif“: Grafiklinksymbole für nächste und vorhergehende Wortlistenseiten; sonst Textlink
- Schalter „mit Grafikbuttons“: Verwendung der Grafikdateien L1.GIF ... L4.GIF statt der Windingszeichen
- Schalter „mit Dokumentlisten“: für Anwendungen ohne Vollanzeige
- Schalter „mit Bestellung“: E-Mail und/oder Briefadresse eingeben; bei der Briefadresse neue Zeilen mit ^

Wortlisten sind in HTML-Format gespeicherte Indexlisten. Vereinbaren sie dazu

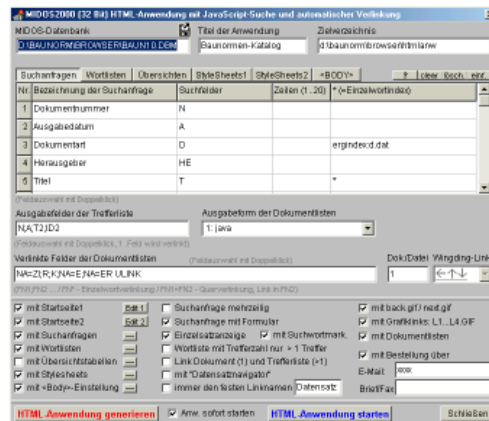
- die Bezeichnung der Wortliste
- mit Doppelklick die Felder und
- die Spaltenzahl (1..3) für die Darstellung. Wenn zum gleichen Feld eine Suchanfrage existiert, kann ein Link zur Trefferliste eingestellt werden.

Übersichtslisten sind Rechercheausgabelisten im HTML-Format. Damit können problemorientierte Listen generiert werden. Vereinbaren Sie hier die

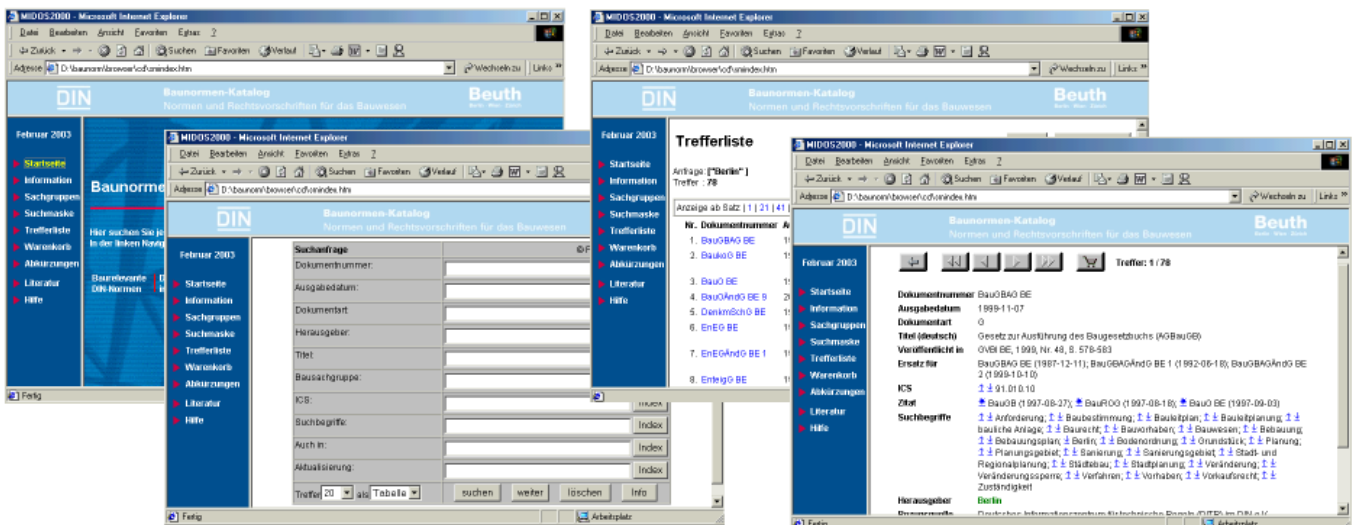
- Benennung des Links für den Aufruf
- Die Ausgabefelder mit Doppelklick und
- die Suchfrage (Syntax wie MIDOS-Volltextrecherche)

Stylesheets- und Body-Einstellung sollten nur bei entsprechender Kenntnis geändert werden.

HTML-Anwendung einstellen



Ergebnis



Die Anwendung wird mit dem Button „HTML-Anwendung generieren“ im Zielverzeichnis erzeugt und wird dann sofort gestartet, wenn der Schalter „Anw. sofort starten“ gesetzt ist. Das Zielverzeichnis kann auf CD oder auf einen WEBserver kopiert werden. Der Aufruf erfolgt mit der HTML-Datei SMINDEX.HTM.

5.3 PERL-Anwendung

Die HTML-Anwendung nach Punkt 5.2 besitzt hinsichtlich der Dokumentanzahl Einschränkungen. Die Perl-Anwendung ist CGI-orientiert und kann wesentlich mehr Dokumente verwalten. Die Bedienung entspricht der der HTML-Anwendung. Sie ist aber nur auf WEB-Server lauffähig.

Die PERL-Anwendung steht erst ab ca. III/2003 zur Verfügung.

6. Anhang:

6.1 Anweisungen der Ausgabebeschreibung

Die vier Formen des Aufrufs eines Feldes, dessen Feldwert an die Position des Feldaufrufes gestellt wird, berücksichtigen einige kleinere Manipulationsmöglichkeiten:

{FN}

Das Kommando bewirkt das Einfügen des Feldes mit dem Feldnamen FN. Das Feld wird in voller Länge ausgegeben. Es wird ein Umbruch am Fenster ausgeführt, wobei am Leerzeichen in der Nähe des rechten Rahmens des Fensters umgebrochen wird. Wird kein Leerzeichen gefunden erfolgt ein Zwangsumbruch am rechten Rahmen des Fensters.

Alle Umbrüche werden dynamisch berechnet. Wenn Sie direkt auf den Drucker ausgeben wollen, sollten Sie die dort angebotenen Preview benutzen, um Ihre Ausgabegestaltung zu begutachten.

{FN:50}

Das Kommando bewirkt das Einfügen des Feldes FN mit maximal 50 Zeichen. In Abhängigkeit von der gewählten Schrift kann eine feste Breite (bei Schriftart Courier oder mit HTML-Tag `<tt>Text</tt>` für teletyp) erreicht werden oder nicht.

{FN:20:50}

Das Kommando bewirkt das Einfügen des Feldes FN, ab dem 20. Zeichen des Feldes, wobei bis zu 50 Zeichen ausgegeben werden. Ist der Feldwert kürzer, wird mit Leerzeichen aufgefüllt.

{FN[x]} x = [1...32000]

{FN[x]:F}

Das Kommando bewirkt das Einfügen der x. Aussage des Feldes FN.

Hinweis: Ausgabe eines Feldes bzw. die Ausgabe der x. Aussage eines Feldes entfällt normalerweise, wenn das Feld bzw. die Feldaussage nicht vorhanden ist. Wird der Feldaufruf mit der Kennung F beendet, dann werden Platzhalter für nichtvorhandene Felder bzw. Aussagen benutzt. Damit ist eine tabellarische Anordnung von Feldern mit mehreren Aussagen möglich, auch wenn diese Felder oder Aussagen fehlen sollten.

Beispiele:

{VF[3]:F}

Wirkung: Wenn die 3. Aussagen des Feldes VF fehlt, soll ein Platzhalter gesetzt werden.

{VF[3]:50:F}

Wirkung: Wenn die 3. Aussage vorhanden ist, sollen 50 Zeichen davon ausgegeben werden, wenn nicht, soll ein Platzhalter der Länge 50 eingesetzt werden.

{*}

Wirkung: Der Spezialaufruf Stern {*} bewirkt die Anzeige des Datensatzes im Speicherformat. Die Ausgabe von Datensätzen im Speicherformat erreichen Sie auch über die Funktion **Ansicht,Datenbank im Speicherformat** oder **Datensatz im Speicherformat** im Grundmenü des MIDOS-Datenbankmanagers oder über die rechte Maustaste in der Kurzanzeige.

Textaufruf

An jeder **beliebigen Stelle** einer Ausgabebeschreibung **kann ein Text stehen**. Steht der Text vor oder nach einem Feldaufruf, dann wird der Text nur ausgegeben, wenn das Feld vorhanden ist. Ausnahme: Parameter F im Feldaufruf.

Befinden sich **mehrere Feldaufrufe in einer Zeile**, werden die Feldwerte hintereinander in einer Zeile ausgegeben. Die ggf. zwischen den Feldwerten gewünschten konstanten Texte können dann nicht in Abhängigkeit vom Vorhandensein des Feldwertes gesetzt werden, sondern werden immer ausgegeben.

Für solche Spezialaufbereitungen muss die Ausgabeaufbereitung gezielt auf die Ausgabe von Zeilenwechsel umgestellt werden. Dafür gibt es den Parameter **mit Zeilenwechsel**. Sie müssen die notwendigen Zeilenwechsel durch Angabe des Zeichens ^ selbst setzen. Dabei gilt wieder, dass ein Zeilenwechsel, der auf gleicher Zeile wie der Feldaufruf steht, ausgeführt wird, wenn das Feld vorhanden ist, sonst nicht. Durch IF-Bedingungen können weitere Verfeinerungen der Einblendung von Zeilenwechseln in Abhängigkeit vom Inhalt der Datensätze gesteuert werden.

Achten Sie auf den Unterschied zwischen einem leeren Feld (der Feldname ist in dem Dokument vorhanden, es hat aber keinen Feldwert) und einem nicht vorhandenen Feld (der Feldname ist nicht im Datensatz).

Beispiel:

Wenn mit der Normaleinstellung Umbruch wie CRLF in der Ausgabebeschreibung gearbeitet wird:

Konstanten Text vor dem Feldwert ausgeben. {FELD1}
 {FELD1} konstanten Text nach dem Feldwert ausgeben.
 Vor {FELD1} und nach dem Feldwert Text ausgeben.
 Text erzwingen, auch wenn Feld fehlt: {FELD1:F}
 Feld {FELD1} und {FELD2} auf gleicher Zeile ausgeben.

Wenn mit ^ als Umbruchzeichen gearbeitet wird:

Text und Umbruch vor dem Feld ^{FELD1}
 {FELD1} ^ Umbruch und Text nach dem Feld.

Äußere Einstellung

Als äußere Einstellung der Ausgabebeschreibung sind zwei Schaltflächen verfügbar:

- **Feldname fett**, um schnell für alle im folgenden ausgewählten Felder, die in die Anzeige kommen sollen, automatisch die Feldbezeichnung fett einzustellen.
- **mit Zeilenwechsel**, um speziell aufbereitete Ausgaben herstellen zu können. Der Standardwert ist so eingestellt, dass Zeilenwechsel (CRLF/Enter/Return) in der Ausgabebeschreibung als auszuführende Zeilenwechsel gedeutet werden.
- Bei der Anzeige der Ausgabebeschreibung mit dem MIDOS-Texteditor sehen Sie diesen Parameter mit ZW=1. Wenn Sie den Parameter auf ZW=0 setzen, müssen Sie auszuführende Zeilenwechsel mit dem Zeichen ^ selbst definieren.
- die **Eintrückungsformatierung**, die Sie mit dem Parameter ER=1 als Überbleibsel aus früheren MIDOS-Versionen sehen, gibt es in dem Sinne "Zeichenzahl, um die Folgezeilen eingerückt werden" nicht mehr. Die Einrückungsformatierung wird mit dem Zeichen ~ (tilde) dargestellt. Das Maß der Einrückung wird in Pixel hinter einem **\$-Kommando** angegeben oder dynamisch aus den Wertelängen der Feldbezeichnungen errechnet.

Die Ausgabebeschreibung mit Zeilenwechsel werden Sie benötigen, wenn kompliziertere Abhängigkeiten mit Sprungbefehlen und feldaussagenabhängigen Sprungzielen berücksichtigt werden sollen.

Sie können im Menü zur Entwicklung von Ausgabebeschreibung die Feldbezeichnung über das Feldauswahlmenü einblenden und den Feldaufruf dazu oder nur den Feldaufruf, dann wird keine verbale Benennung des Feldes ausgegeben. Statt dessen können Sie konstante Texte beliebiger Art setzen.

Merke: Der Bezeichnung eines Feldes in der Ausgabe ist unabhängig von der Bezeichnung in der Datenbeschreibung.

Die erste, automatisch erzeugte Ausgabebeschreibung wird mit den Standardwerten für Umbruch, Einrückung und Zeilenwechsel hergestellt. Sie können Ausgabebeschreibungen so ändern, dass die Feldnamen statt der Feldbezeichnung ausgegeben werden. Sie können Feldaufrufe löschen, Kommandos einfügen, Einrückungen ändern, Zeichen zwischen Feldwerten und vor und nach Feldaufrufen ändern und Sie können den Zeilenvorschub bedingt steuern.

Wenn Ihnen die Definition anspruchsvoller Ausgabeformen zu aufwendig ist, dann können Sie uns damit beauftragen. Mit einer von Ihnen erarbeiteten detaillierten Vorlage können wir Ihren Auftrag gern bearbeiten.

Formatierungskommandos**\$\$Tab**

Das Symbol leitet eine allgemeine Tabellenaufbereitung ein. Es ist auch das HTML-Tag <table> möglich.

\$\$Tab rahmen

Das Symbol leitet eine allgemeine Tabellenaufbereitung mit Rahmenlinien um Zeilen und Spalten ein.

\$\$x

Für die Tabelle wird die Breite der ersten Spalte x festgelegt. Wenn die 1. Spalte die Feldbezeichnungen enthält, wird also die Spalte mit den Feldbezeichnungen auf eine feste Grösse in Pixel eingestellt. Die Pixelbreite x der 1. Spalte kann zwischen ca. 10...500 Pixel liegen. Sie müssen die Gestaltung am Bildschirm auf der Basis Ihrer Bildschirmgrösse und Auflösung testen. Die Breite der 1. Spalte legt die Position der Folgezeilen von Feldwerten fest, wenn der Feldaufruf die 2. Spalte der Tabelle bildet.

\$\$0

Mit dem Kommando beenden Sie die Tabelle, es ist auch möglich eine Leerzeile anzugeben. Eine Leerzeile wird durch die Eingabe von CRLF Leertaste CRLF in der Ausgabebeschreibung erzeugt.

~ (Tilde)

Das Zeichen ~ (Tilde) ist der Tabellenspaltentrenner. An diesem Trenner orientiert sich der Beginn der nächsten Spalte.

Beispiel:

\$\$Tab**Verfasser~Titel~Erscheinungsvermerk**

{VF:F}~{ST:F}~{EK:F}

\$\$0

Im Beispiel wird eine 3spaltige Tabelle aus den Feldern VF, ST und EK gebildet. Dabei wird bei fehlenden Feldern ein Platzhalter gesetzt. Sie müssen auf die Breite solcher Tabellen selbst achten. Die Breite ist u.a. abhängig von der gewählten Schriftgrösse. Es wird versucht, eine dynamische Breitenoptimierung analog HTML-Tabellen zu erreichen.

' Apostroph

Das Zeichen ' (Apostroph) dient als Einschliesszeichen für Fettmarkierung und Tabellenanfang. Ein zweites Hochkomma in einer Zeile wirkt wie ein Spaltentrenner. Das Zeichen **muss paarig pro Zeile** und darf maximal zweifach pro Zeile verwendet werden! Eine Textauszeichnung erreichen Sie auch mit den HTML-Tags und , <i> und </i> oder <u> und </u>. Feldwerte können auch mit <h1> und </h1> bis <h7> und </h7>, den HTML-Überschriftenformaten, ausgezeichnet werden.

\$*x

Mit dem Kommando können Aufzählung mit einem dickeren Mittelpunkt (auch Bullit genannt) auf eine Ebene x gesetzt werden. Es sind 9 Ebenen denkbar. Anwenden kann man diese Funktion z.B. um die Aussagen eines Feldes mit Einrückungssymbol untereinander auszugeben. Es können auch mehrere Felder sein, die untereinander, gestaffelt nach Ebenen ausgegeben werden. Die Anzahl der Ebenen ist auf 9 begrenzt. (x=1...9). Das Kommando gilt, bis es wieder ausgeschaltet wird.

\$*0

Das Kommando hebt die Aufzählungsformatierung auf.

\$=xy

Das Kommando tauscht bei der Anzeige das Zeichen x gegen das Zeichen y aus. Als Ersatzsymbol für CRLF/Enter/Return kann ^ angegeben werden. Das Kommando gilt, bis es wieder ausgeschaltet wird.

Beispiel:

Jede Aussage soll auf neuer Zeile stehen: \$=;

Hinweis: Alle \$-Kommandos dürfen nur einzeln auf einer Zeile stehen, also keine Feldaufrufe in gleicher Zeile enthalten, auch keine Konstanten.

\$=zeichenkette1\zeichenkette2

Die Anweisung ersetzt bei der Ausgabe die zeichenkette1 durch die zeichenkette2.

Beispiel:

Jede Aussage soll auf neuer Zeile stehen: \$=; \#13#10

Jedem Semikolon soll ein Leerzeichen folgen: \$=;\#32

\$=0

Das Kommando hebt den Zeichentausch wieder auf.

Sprünge, bedingte Anweisungen, Programmschleifen

GOTO M1

Das ist ein unbedingter Sprung zu einer Marke M1.

Ein Rücksprung ist nicht möglich.

:M1

Allgemein Form der Definition einer Sprungmarke am Zielort des Sprungs. Marken müssen am Zeilenanfang stehen und sie müssen Vorwärtssprünge darstellen. Der Doppelpunkt leitet die Sprungmarke am Zielort ein, nicht aber an der Absprungstelle (siehe GOTO) .

IF-Abfrage

Es gibt mehrere Formen der IF-Abfrage. Hinter dem Feldaufruf einer IF-Abfrage kann nur ein EXIT oder ein GOTO stehen. In Abhängigkeit von der Antwort auf die Sprungbedingung kann also nur abgebrochen oder zu einer Sprungmarke gesprungen werden.

Ist die Sprungbedingung nicht erfüllt, wird mit der Anweisung fortgesetzt, die der IF-Abfrage folgt.

IF {FN} GOTO M1

Abfrage, ob Feld {FN} existiert. Ein leeres Feld existiert im Sinne dieser Abfrage. Ein fehlendes Feld existiert nicht.

IF {FN#} GOTO M1

Abfrage, ob das Feld FN nicht existiert.

IF {FN=zeichenkette} GOTO M1

Abfrage, ob das Feld FN die Zeichenkette zeichenkette enthält. Die Zeichenkette muss in Apostroph eingeschlossen werden, wenn Leerzeichen, geschweifte Klammern oder Apostroph enthalten sind.

IF {FN#;zeichenkette} GOTO M1

Abfrage, ob das Feld FN die Zeichenkette zeichenkette nicht enthält.

IF {FN!zeichenkette} GOTO M1

Abfrage auf die 1. Position der Zeichen xxx im Feld {FN}. Stehen die Zeichen auf der 1. Position, wird der folgende Befehl ausgeführt.

IF {FN=zeichenkette} goto M1

Sprung zur Marke m1, wenn im Feld FN die Zeichenkette zeichenkette enthalten ist.

EXIT

Das Kommando bricht die Aufbereitung der Ausgabebeschreibung sofort ab. z.B.

if {LG=} exit

Wirkung: Wenn das Feld LG existiert, brich ab.

HTML{+}

:HTMLENDE

Die zwischen diesen Anweisung liegenden HTML-Kommandos werden nicht von den MIDOS-Klassik-Druckaufbereitungen (1:1) ausgegeben. Diese Anweisungen erübrigen sich, wenn nur mit der neuen MIDOS-HTML-Druckausgabe gearbeitet wird oder keine HTML-Kommandos verwendet werden.

Reservierte Feldaufrufnamen

RAK

Wenn in der Funktion **RAK-Felder definieren** eine Anpassung der Felder an das RAK-Format vorgenommen wurden, kann mit dem Aufrufnamen das RAK-Format erzeugt werden.

Vor und nach dem Aufruf von *RAK* können weitere Feldaufrufe stehen, so dass auch um weitere Felder ergänzte

Ausgaben erzeugt werden können. Aber diese weiteren Aufrufe und Konstanten dürfen nicht auf der gleichen Zeile stehen, wie der reservierte Aufrufname.

RAKF

wie *RAK*, jedoch wird der Inhalt des Felde TITEL in Fettschrift ausgegeben. Beachten Sie die RAK-Feldanpassung.

DATE

Das aktuelle Systemdatum wird ausgegeben.

TIME

Die aktuelle Uhrzeit wird ausgegeben.

D

Datum und Uhrzeit werden zusammen ausgegeben.

LFDNUM

Die laufende Treffernummer wird ausgegeben. Bei der Tabellenausgabe wird die Treffernummer mit ausgegeben, ohne explizit *LFDNUM* angeben zu müssen.

AKTNUM

Wird in Schleifen benötigt, um die aktuellen Datensatznummer mit der maximalen Datensatznummer zu vergleichen, um die Schleife abzubrechen, wenn der letzte Datensatz verarbeitet wurde.

MAXNUM

Enthält die Anzahl der Datensätze der anzuzeigenden Treffermenge.

REM oder ; am Zeilenanfang

Bemerkung. Die Zeile wird nicht ausgeführt. Auch ein Semikolon am Zeilenanfang führt zum Ausschluss der Zeile.

SUMME

Das Feld enthält den Wert einer Summenbildung.

Einfache HTML-Tags zur Gestaltung

Im Prinzip versteht der interne HTML-Viewer alle Kommandos der HTML Version 3 und teilweise der Version 4. Wir wollen Ihnen aber diese Vielfalt an. Aufbereitungsmöglichkeiten in komplexeren Funktionen zusammenfassen, wie Sie in den Unterfunktionen **Tabelle 1 (markierte Felder)** oder auch in der Funktion **Bibliografie** vorliegen.

Die im folgenden aufgeführten HTML-Tags werden vom HTML-Viewer ausgeführt, gleich ob sie aus dem Text des Dokumentes oder aus der Ausgabebeschreibung stammen und unabhängig davon, ob der HTML-Start gesetzt ist, oder nicht. Unterbunden wird die HTML-Aufbereitung zwischen den HTML-Tags **<code>** und **</code>**.

	Fettmarkierung ein
	Fettmarkierung aus
<i>	Kursiv ein
</i>	Kursiv aus
<u>	Unterstreichung ein
</u>	Unterstreichung aus
<p>	Absatz, grösser als
 1)

	Absatz 1)
<pre>	Vorformatierung beibehalten, CRLF wird ausgeführt 1).
</pre>	Ausschalten von Vorformatierung beibehalten 1).
<h1>	Überschrift Format Schrift sehr gross einschalten 1).
</h1>	Überschrift Format Schrift sehr gross aus ...bis... 1)
<h7>	Überschrift Format Schrift sehr klein einschalten 1).
</h7>	Überschrift Format Schrift sehr klein ausschalten 1).

1) Die Kommandos bringen einen **eigenen Zeilenwechsel** mit, der vor oder nach der Ausführung des Kommandos ausgeführt wird. Das kann mit der Formatierung mit ^ kollidieren.

Link auf Fremddateien

Die Fremddatei muss eine MIDOS-Datenbankdatei sein. In einigen Fällen wird der FILELINK in Form eines HTML-Scripts benötigt. Siehe dazu **Einfügen von HTML-Spezialelementen**.

{FILELINK:[<dateiname>],[<keyfeld>],[<keyword>],[<rückgabefeld>],[<wert bei rückgabe=0>]}

Dieses Kommando ist wie folgt zu interpretieren:

- unter **<dateinamen>** geben Sie den Namen der Datei an, die zu der Ausgabebeschreibung hinzugelinkt werden soll, wenn sie das **<keyfeld>** und den **<keyword>** enthält.
- **<keyfeld>** ist ein Feldname, der in der aufgerufenen Datei existieren sollte, aber nicht muss.
- **<keyword>** ist eine Suchzeichenkette, möglichst mit dem Charakter eines Keys (Schlüssel, Primär- oder Sekundärschlüssels).
- **<rückgabefeld>** ist der Name eines Feldes (muss nicht in der Datenbeschreibung der aktuellen Datenbank existieren, aber in der Ausgabebeschreibung sollte es genannt sein, weil es die Ergebnisse des Links beinhalten wird, wenn er erfolgreich war).
- unter **<wert bei rückgabe=0>** geben Sie eine Zeichenkette an, die in das **<rückgabefeld>** gestellt wird, wenn der Link kein Ergebnis liefert.

Da entsprechend der Syntaxdarstellung jeder einzelne Parameter des FILELINK-Kommandos fehlen kann, bedeutet das, dass praktisch auch eine Verlinkung mit der aktuellen Datenbankdatei selbst möglich ist. Für ausgelassenen Parametern ist das Komma als Trenner zu setzen.

Wenn **kein Keyfeld** genannt ist, bedeutet dies, dass Links hergestellt werden, ohne ein spezielles Feld zu berücksichtigen, das heisst, die Suchzeichenkette wird im ganzen Dokument der hinzugelinkten Datei gesucht.

Wenn kein Keyword genannt ist, bedeutet dies, dass Links hergestellt werden, ohne ein spezielles Keyword zu berücksichtigen, das heisst, **alle Feldwerte** des Keyfeldes der Quelldatei werden als **Suchzeichenkette** in der hinzugelinkten Datei gesucht. Alle getroffenen Dokumente der hinzugelinkten Datei werden zum Ergebnis des Links. Es können sehr grosse Ergebnisse entstehen. Sie sind ggf. so gross, dass sie mit einfachen HTML-Browsern nicht ohne weiteres anzeigbar sind.

Dateien direkt einfügen

Die folgende Kommandoform fügt eine **beliebige Textdatei** direkt an die Position im angezeigten Dokument ein, an der der Aufruf erfolgt. Die Textdatei sollte zeilenweise organisiert sein, wobei die Zeilenlänge kleiner als 30000 Zeichen sein sollte.

{FILELINK:[<dateiname>],text,text}

Die folgende Kommandoform fügt eine **beliebige HTML-Datei** direkt an die Position im angezeigten Dokument ein, an der der Aufruf erfolgt. Die Dateigrösse spielt keine Rolle.

{FILELINK:[dateiname],html,html}

Formeln

Es existieren verschiedene vorgefertigte Formeln und frei definierbare Formeln, die auf Feldaufrufen beruhen. Es können 9 Formeln in einer Ausgabebeschreibung definiert werden. Die allgemeine Form ist:

{FORMELx:[formel]} mit x=[1..9]

Das Wortsymbol FORMEL: innerhalb eines speziellen Formel-Feldaufrufes leitet einen Formelausdruck ein. Es folgen die Feldaufrufe des zur Berechnung heranzuziehenden Feldes, ein Operator

* **Multiplikation,**
 / **Division,**
 + **Addition,**
 - **Subtraktion**

und ggf. ein weiterer Feldaufruf oder eine Rechenkonstante. Sind in den aufgerufenen Feldern Angaben enthalten, die eine Berechnung nicht zulassen, wird versucht, eine **erste Zahl von Links** im Feld zu isolieren und damit die Berechnung auszuführen. Gelingt dies nicht, wird die Berechnung ignoriert.

{FORMEL:{PREIS} * 0.16}

Der Wert des Feldes Preis wird in allen Dokumenten mit 0.16 multipliziert. Das Ergebnis wird an der Position angeordnet, an der der Formelaufwurf erfolgte.

Zur Vereinfachung wird die **Summenbildung** über mehrere Dokumente in Summenfeldern mit den Namen *SUMME x * gespeichert, wobei mit $x=[1.. 9]$ neun Summen gleichzeitig über eine Treffermenge gebildet werden können.

Programmschleifen

Programmschleifen werden mit den Wortsymbolen REPEAT und UNTIL formuliert. Eine Laufvariable ist erforderlich zum Zählen der Schleifendurchläufe und als Endebedingung.

REPEAT(*VAR*=xx)

REM Schleifenbeginn: *VAR* erhält den Wert xx.

UNTIL(*VAR*=yy)

REM Schleifenende: wenn *VAR* = yy ist.

VAR

Enthält den Wert der Laufvariablen, max. 1000 Schleifendurchgänge.

Schleifenbeispiel:

```
REPEAT(*VAR*=1)
/*VAR*/{FN[*VAR*]}
IF {FN[*VAR*]} e x i t
IF {FN[*VAR*]=xxx} goto m1
UNTIL(*VAR*=100)
:m1
{FN}
```

Die Laufvariable wird auf 1 gesetzt. Dann wird je eine Aussage des Feldes FN geholt, solange noch eine gefunden wird, deren Wert xxx ist oder bis die 100 Aussage verarbeitet ist. Die Anweisung /*VAR*/ erhöht die Laufvariable.

HTML-Spezialelemente einfügen

Unter der Funktion **HTML-Spezialelemente einfügen** können Sie:

- **Tabellen erzeugen mit Tabelle1 (markierte Felder) oder Tabelle2 (Felder wie Kurzanzeige),**
- **Überschriften** einblenden,
- **Textboxen** einfügen,
- den **Hintergrund** farblich gestalten,
- **Grafikdateien** direkt **einfügen,**
- Grafikdatei **aus Einzelfeldern** einfügen,
- Grafikdatei **aus Mehrfachfeldern** einfügen,
- **Internet-Links** einfügen,
- **Links** auf **beliebige Dateien** einfügen,

- **Links** auf **Einzelwertfelder** einfügen,
- **Links** auf **Mehrfachfelder** einfügen,
- **Links** auf **Ausgabeformen** setzen,
- **Links** auf Verzeichnis mit **TIF-Dateien** legen,
- **Fremdfelder** direkt einfügen, die aus anderen MIDOS-Datenbanken stammen und
- **Links** auf **Fremddatensätze** herstellen, die aus fremden MIDOS-Datenbanken stammen.

Link auf Ausgabeform

Mit der Funktion **Link auf Ausgabeform** können Sie während der Anzeige von Datensätzen das Ausgabeformat ändern. Die Funktion kann von der IF-Bedingung abhängig gemacht werden.

Damit kann in Abhängigkeit vom Inhalt eines Feldes die Ausgabeform grundsätzlich gewechselt werden, z.B. kann einer Liste eine Tabelle zugefügt werden. Im folgenden Beispiel wurde als Linkbezeichnung *Link auf Ausgabeform 3* eingegeben und die Nummer der Ausgabebeschreibung 3 angegeben, die zu benutzen wäre, wenn der Link angeklickt wird.

```
HTML{+++}  
<CENTER>  
<a href="AFORM3.HTM">Link auf Ausgabeform 3</a>  
</CENTER>  
:HTMLENDE
```


6.2 Kommandoübersicht von MIDOS-UPDATE

Die folgenden Kommandos werden vom Programm MWUPDATE.EXE abgearbeitet und dienen der zyklischen Bearbeitung von Daten. Die Kommandos werden in MJB-Dateien gespeichert.

Alle Kommandos werden mit bis zu 3 Parameterzeichenketten gesteuert, die in der folgenden Erläuterung mit 1.: 2.: und 3.: gekennzeichnet sind. („---“, bedeutet: Parameter nicht erforderlich)

ACOPY

Kopiert eine Datei an einer existierende Zielfile wird angehängt

- 1.: Quelldatei
- 2.: Zielfile
- 3.: ---

ADDFILE

Hängt an eine Datei an eine andere an. Achtung: Zielfile wird verändert!

- 1.: anzuhängende Quelldatei
- 2.: zu ergänzende Zielfile
- 3.: ---

AFORMAT

Erstellt Ausgabeformat ohne Zeilenformatierung

- 1.: Quelldatei mit MIDOS-Satzendezeichen
 - 2.: Zielfile
 - 3.: Ausgabebeschreibungsdatei[x]
- x: 0 ... 20 / Nr. der Ausgangsbeschreibung, Format wie Datei MIAUSART, MITSW:xxx (nur wenn xxx im Satz), OHNESW:xxx (nur wenn xxx nicht im Satz)

ANALYSE

Untersucht Syntax einer MIDOS-Datei

- 1.: MIDOS-Datei
- 2.: Protokoll-Textdatei
- 3.: ---

ANZEIGE

Zeigt beliebige Datei an (mit MWLIST.EXE) und wartet auf Beendigung

- 1.: anzuzeigende Datei
- 2.: -
- 3.: -

APPENDSTRING

Hängt an eine Datei einen String an, wenn dieser noch nicht existiert.

- 1.: Dateiname
- 2.: -
- 3.: Zeichenkette

Existiert die Datei nicht, dann wird sie angelegt und die Zeichenkette ab Position Null geschrieben. Die Zeichenkette kann Steuerzeichen enthalten: #<ASCII-CODE> oder =CRLF

CD

Verzeichnis wechseln

- 1.: Verzeichnisname, das aktiviert werden soll
- 2.: ---
- 3.: NOERROR - kein Abbruch bei Fehler

CHAIN

Führt weitere JOB-Datei aus oder startet Fremdprogramm ohne auf Beendigung zu warten

- 1.: JOB-Datei(*.MJB) oder EXE-Datei
- 2.: von Jobschritt (nur bei MJB-Dateien)
- 3.: bis Jobschritt (nur bei MJB-Dateien)

CHIFF

Datei wird chiffriert

- 1.: Lesbare Quelldatei

- 2.: Chiffrierte Zielfeld
- 3.: Codewort

CLOSE

>>> Beendet das gesamte UPDATE-Programm

- 1.: ---
- 2.: ---
- 3.: ---

COPY

Kopiert eine Datei; eine existierende Zielfeld wird überschrieben

- 1.: Quelldatei
- 2.: Zielfeld
- 3.: ---

DECHIFF

Datei wird dechiffriert

- 1.: Chiffrierte Quelldatei
- 2.: Lesbare Zielfeld
- 3.: Codewort---

DECOMP

Dekomprimiert und/oder kopiert Dateien, die mit COMPRESS gepackt wurden.

- 1.: Quelldatei
- 2.: Zielfeld
- 3.: ---

DELINFO

Löscht das Infofenster

- 1.: ---
- 2.: ---
- 3.: ---

DELSATZ

Löscht MIDOS-Sätze durch selektives Kopieren

- 1.: MIDOS-Quelldatei
- 2.: MIDOS-Zielfeld
- 3.: KEYFELD:xx KEYFILE:yy

KEYFILE enthält zeilenweise (max. 10000) den KEYFELD-Inhalt der Sätze, die nicht kopiert werden sollen

DEL

Löscht eine/mehrere Dateien (auch mit *.xxx)

- 1.: zu löschende Datei
- 2.: ---
- 3.: NOERROR - kein Abbruch bei Fehler

DOPPELAUSSAGEN oder DOPPELAUSSAGEN2

Doppelaussagen: Prüft auf doppelte Aussagen in einem Feld (max 1000 Aussagen a 60 Zeichen)

Doppelaussagen2: Prüft auf doppelte Aussagen in einem Feld (max 250 Aussagen a 250 Zeichen)

- 1.: MIDOS-Quelldatei
- 2.: MIDOS-Zielfeld
- 3.: DOPPELFELD:<Feldname>

TRENNER:<trennzeichen> (nur für Schreiben) "b" reserviert für Blank

DOPPELINDEX

Kopiert doppl.freie MIDOS-Sätze über Indexdatei. Bei Dopplungen wird der in der Pooldatei der zuletzt stehende verwendet. Hinweis: Das Indexfeld darf nur eine Aussage enthalten (keine MIDOS-Trennzeichen).

- 1.: existierende Indexdatei
- 2.: MIDOS-Zielfeld
- 3.:---

DOPPEL

Kopiert nicht doppelte Datensätze

- 1.: Quelldatei
- 2.: Zieldatei
- 3.: Dopplungsprüffelder mit folgender Syntax:

FN1(ab1,anz1);FN2(ab2,anz2) ...

FN - Feldname

ab - Ab-Zeichen im Feld

anz- Zeichenzahl

max Gesamtzeichenzahl: 250, zB.:ST(1,40);VF(1,30)

MITSW:xxx (Doppl.prüfung nur, wenn xxx im Satz)

OHNESW:xxx (Doppl. prüfung - nur, wenn xxx nicht im Satz)

DOSEXEC

Führt ein oder mehrere DOS-Programme aus und testet alle 15 Sec. auf Existenz des DOS-Fensters

- 1.: Programmname (mehrere mit Semikolon trennen)
- 2.: weitere Programme
- 3.: Fenstername/Klasse des DOS-Fensters (Standard: tty)
Beispiel: echo Directory listen;pause;dir;pause

DRIVE

- 1.: Buchstabe des Laufwerkes, das aktiviert werden soll
- 2.: ---
- 3.: NOERROR - kein Abbruch bei Fehler

ENDE

Beendet das UPDATE-Programm mit INFO-Fenster

- 1.: ---
- 2.: ---
- 3.: ---

ERGFELD

Ergänzt in einer MIDOS-Datei ein Feld

- 1.: Quelldatei
- 2.: Zieldatei
- 3.: FN:FELDDINHALT <parameter> BEARBEITUNG:xy KONSTANT:x FÜLLZEICHEN:c

Syntax:

FN - MIDOS-Feldname

FELDDINHALT - Texte/andere MIDOS-Felder mit {} z.B. GESTITEL:{TITEL1} | {TITEL2} | {TITEL3}

Parameter:

mitsw:xxx

ohnesw:xxx - mit/ohne Suchworte

von:xxx,

bis:yyy - nur Datensatznummer xxx – yyy

BEARBEITUNG:xy (Bearbeitungskennung bei Bedarf, Standard:x=0, y=0)

x=0 alle Felder

x=1 nur exist. Felder

x=2 nur nicht exist F.,

y=0/=1 nicht/überschreiben

KONSTANT:x - Feld mindestens x Zeichen

FÜLLZEICHEN:c - Füllzeichen c für "KONSTANT" (Stand:0)

ERGINDEX

Ergänzt den Indexeintrag

- 1.: 1. INDEX-Quelldatei (Typ: *.WTX)
- 2.: 2. Zieldatei (Typ: *.WTX)
- 3.: Textdatei mit folgendem Aufbau:

<INDEXEINTRAG><3 Leerzeichen><Erläuterung>

FELD

Manipuliert Feldinhalte (s. Punkt 7.3), Ausführung von selbsterstellten EXEC-Proceduren

- 1.: Quelldatei
- 2.: Zieldatei
- 3.: Datei mit Feld-Manipulationsanweisungen (Typ: *.TAB)

FRONT

UPDATE-Fenster in Normalform anzeigen

- 1.: ---
- 2.: ---
- 3.: ---

GOTO

Springe zur JobNr. x

- 1.: Jobnummer x oder direkt nach GOTO
 - 2.: ---
 - 3.: ---
- (max. 1000 Schleifendurchläufe)

HIDDEN

UPDATE-Programm nur als Symbol sichtbar

Programm meldet sich nur bei Fehler

- 1.: ---
- 2.: ---
- 3.: ---

HTMLANZEIGE

Zeigt HTML-Datei an (mit M2HTML32.EXE) und wartet auf Beendigung

- 1.: anzuzeigende Datei
- 2.: -
- 3.: -

INDEX oder FASTINDEX

Erstellt INDEX über MIDOS-Dateien

- 1.: Quelldatei mit MIDOS-Satzendezeichen
- 2.: Zieldatei (Typ: *.WTX)
- 3.: Einstellungen:
INDEX:<feld1>;<feld2>; ...
NOANSI (keine ANSI-Umsetzung)
EINZEL (Einzelwörter/sonst MIDOS-Aussagen)
TRENNZEICHEN:xyz (Wortlesen bis zu den Zeichen xyz, "b" reserviert für Blank, hebt EINZEL auf!)
NULLZEICHEN:xyz (Zeichen xyz werden gelöscht)
NUMMER (Satzadresse = lfd. Nummer)
MITFELDNAME (Index mit Ursprungsfeld)
STW:<datei1>;<datei2> ...(Stoppwortliste)
WL:xx (Wortlänge:xx=1..250, sonst 80)
TEXT (Liest Textdatei, 1 Zeile = 1 Satz)
VONZEICHEN:x - Wortgruppe ab Zeichen x einlesen
BISZEICHEN:x - Wortgruppe bis Zeichen x einlesen
SAMMELEINTRAG:YYY - Eintrag im Index
SAMMELWORT:XXX - Sucht beliebiges Wort (Wenn kein Sammeleintrag dann YYY=XXX)
SAMMELLISTE:<filename> - Liste in Textdatei (max. 2000 Einträge a 30 Zeichen)
MITSW:xxx (Index - nur wenn xxx im Satz)
OHNESW:xxx (Index - nur wenn xxx nicht im Satz)
SZ:xx (Satzzahl bis zur Zwischenspeicherung)
WZ:xx (Wortzahl bis zur Zwischenspeicherung)

INDEXTOTEXT

Erstellt eine Textdatei aus einem INDEX

- 1.: Indexdatei (Typ: *.WTX)

2.: Zieldatei (Textdatei)

3.: TREFFER SATZINDEX WORTLÄNGE

TREFFER: Speicherung mit Trefferzahlen

SATZINDEX: Speicherung mit WTX-Indexzahlen

WORTLÄNGE: Speicherung mit Wortlängenzahlen

Die Satzindex-Textdatei beschleunigt die sequentielle Suche in der Wortliste erheblich. Der Name der Textdatei muss hierbei *.SEQ sein

INFO

Zeigt ein Info-Fenster an

1.: Überschrift (max. 40 Zeichen)

2.: Text (max 250 Zeichen)

3.: HALT - Abarbeitung wird angehalten

ISO2709

ISO2709 - Erzeugt aus ISO 2709-Daten MIDOS-Daten

1.: ISO 2709-Datei

2.: MIDOS-Datei

3.: sz=x tz=y elsevier LFDNUM:<feld,zz>

sz - Satzendezeichen

tz - Feldtrennzeichen

vier - elsevier-ISO-Format

LFDNUM:<feldname,abnummer> => laufendes Nummernfeld einfügen

LIX

Bildet Indexhilfsdatei vom Typ ".LIX", große Indizes werden schneller

1.: 1. INDEX-Datei (Typ: *.WTX)

2.: ---

3.: ---

MAKEANSI

DOS-ASCII -> WINDOWS-ANSI Umsetzung

1.: MIDOS-Datei im DOS-ASCII-Format

2.: MIDOS-Datei im WINDOWS-ANSI-Format

MAKEASCII

WINDOWS-ANSI -> DOS-ASCII Umsetzung

1.: MIDOS-Datei im WINDOWS-ANSI-Format

2.: MIDOS-Datei im DOS-ASCII-Format

MAKEDBASE

Umsetzung MIDOS -> dBASE III, dBase-Kopf wird automatisch aus der Datenbeschreibung MISCHABS erzeugt

1.: MIDOS-Quelldatei

2.: DBF-Zieldatei

3.: mitMemo NurExistDbf

Syntax:

mitMemo - erzeugt bei Felder>250 Zeichen Memofelder

NurExistDbf - benutzt nur die Felder einer existierenden DBF-Zieldatei

MAKEHTMLTAB

Umsetzung MIDOS -> HTML-Tabelle

1.: MIDOS-Datei

2.: HTML-Datei

3.: Feld1,Titel1,Para1;Feld2,Titel2,Para2;... DATUM RAHMEN ÜBERSCHRIFT:xxx <parameter>

Parameter:

S - Summenbildung

P - Prozentbildung

N - Numerisch

R - Rechtsbündig

xx% - Spaltenbreite xx=10..90%

xx\$ - Spaltenbreite xx=10..500 Pixel

SUCHPUNKT - 1.Spalte als Suchpunkt (auch SPKT)

bei Feldname=ldfno – Zählnummer, Feldname auch {FN1}
{FN2} ..., bei Feldname {FN!} nur 1. exist. Feld

MAKELINKS

MIDOS-Dateien für Retrievalanwendung mit M2RECH.EXE verlinken

- 1.: MIDOS-Quelldatei
- 2.: MIDOS-Zieldatei
- 3.: FN1;FN2;FN3=FN4 GRAFIK ULINK
FN1... - feldintern zu verlinkende Felder
FN3=FN4 - feldübergreifende Verlinkung, (2. Feldname enthält Link) zB. DOKNR=ERSETZDURCH
GRAFIK - Verwende Grafiklinks mit L1 .. L3.GIF, sonst WINGDINGS-Pfeile
ULINK - Feldübergr. Links als Unterstrich

MAKENUM

Numeriert eine MIDOS-Datei um

- 1.: MIDOS-Quelldatei
- 2.: MIDOS-Zieldatei
- 3.: LFDNUM:<feldname,abnummer,stellen> -
<feldname> : Nummernfeld
<abnummer> : erste Nummer
<stellen> : minimale Stellenzahl

MAN oder STRING

Manipuliert beliebige Text-Dateien über Zeichenkettenaustauschanweisungen

- 1.: Quelldatei (auch mit "**")
- 2.: Zieldatei
- 3.: Manipulationsdateiname SUBDIRS TESTSUCHE
Wenn Quelldatei mit "**" dann Zieldatei ohne Namen, Rückspeicherung in Quelldatei !!! (alt in "*.bak"),
SUBDIRS - Ausführung in allen Unterverzeichnissen
TESTSUCHE - keine Manipulation; nur Anzeige der "**"-Dateien im Protokoll

MAPTOMIDOS

Umsetzung MAP-Format in das MIDOS-Format

- 1.: MAP-Datei
- 2.: MIDOS-Datei
- 3.: -

MD

Erstellt beliebige Verzeichnisstrukturen

- 1.: Verzeichnisstruktur (Pfad), die eingerichtet werden soll
- 2.: ---
- 3.: ---

MIDOSRECH

Kopiert nur das Ergebnis einer sequentiellen Recherche in die Zieldatei.

- 1.: MIDOS-Quelldatei
- 2.: MIDOS-Zieldatei
- 3.: Recherchefrage in der Syntax der MIDOS-Volltextrecherche

MISCHDAT

Mischt MIDOS-Datensätze anhand der Eingabebeschreibungsdatei MISCHABS

- 1.: MIDOS-Quelldatei1
- 2.: MIDOS-Quelldatei2
- 3.: ZIELDATEI KEYFELD:F1;F2;.. multifeld copyall
F1...F10 Schlüsselfelder (max. 10 Felder)
multifeld - alle Datensätze mit gleichem Keyfeld mischen, gleiche Felder kumulieren
copyall - kopiert auch bei nicht gefundenem Keyfeld die Sätze der 1. Quelldatei
Eingabebeschreibungsdatei MISCHABS muss im aktuellem Verzeichnis vorhanden sein !!!

MISCHINDEX

Mischt zwei Indexdateien

- 1.: 1. INDEX-Quelldatei (Typ: *.WTX)
- 2.: 2. INDEX-Quelldatei (Typ: *.WTX)
- 3.: Zieldatei (Typ: *.WTX)

PRUEF

Prüfe MIDOS-Datensätze mit einem Prüfprogramm, Prüfergebnis im Protokoll

1.: MIDOS-Datei

2.: -

3.: Name der Prüfprogrammdatei

Befehle werden beim Erstellen einer neuen Prüfprogrammdatei mit dem Editor (über Doppelklick) angezeigt)

READAUSTAUSCH

Liest Daten im Text-Austausch ein erkennt Felder im Kopf automatisch

1.: Quelldatei im DB-Austauschformat

2.: Zieldatei

3.: ohne Angabe: autom. Erkennung

sonst:FELD:<feld1>;<feld2>;... <parameter>

Parameter:

EZ:x (x:Einschließungszeichen)

TZ:y (y:Trennzeichen)

SZ:abc (abc:Satzendezeichen)

nohead (keine Kopfzeile)

noEZ (keine Einschließungszeichenprüfung)

LEERFELDER werden nicht gelöscht

LFDNUM:<feldname,abnummer> laufendes Nummernfeld einfügen

FELDDATEI:<dateiname> Feldnamen zeilenweise in Datei, wenn Platz nicht reicht

REL

Liest Daten aus festgeblockten Formaten, dBASE-Daten werden automatisch erkannt

1.: Quelldatei im festgeblocktem Format

2.: Zieldatei

3.: <dateiname> LEERFELDER LFDNUM:<feld,zz>

allg. Aufbau der Format-Beschr.datei <dateiname>:

1.Zeile: REL ZL:xx ZZ:yy SL:zz K:tt

REL:Kennung

SL:Satzlänge - Bytezahl/Satz oder

SL:variabel - Satz=Zeile

ZL:Zeilenlänge mit 0D0A

ZZ:Zeilenzahl pro Satz

SL steht alternativ zu ZL,ZZ

K:Kopflänge (Standard:K:0)

2. ... ?.Zeile: Zy,x,l=zk

oder

Pz,l=zk

y:Zeile,x:Spalte,l:Bytezahl

zk:Zeichenkette auch mit ,#xx

z:Byteposition ab Satzanfang

LEERFELDER werden nicht gelöscht

LFDNUM:<feldname,abnummer> laufendes Nummernfeld einfügen

REN

1.: alter Dateiname

2.: neuer Dateiname

3.: NOERROR - kein Abbruch bei Fehler

SATZINDEX

Erstellt einen Satznummernindex über eine MIDOS-Dateien. Notwendig für WTX-Dateien mit dem Parameter NUMMER.

1.: MIDOS-Datei

2.: Zieldatei (Typ: *.SIX)

SCHLAGWORT

Automatische Schlagwortvergabe, Durchsuche SUCHFELDER nach INDEX- und SYNONYM-Einträgen

1.: Quelldatei

2.: Zieldatei

3.: folgende Parameter mit Leerzeichen getrennt:

SUCHFELDER:FN1;FN2

- zu durchsuchende Felder wie Titel, Referat ..

INDEX: <WTX- o. Text-Datei>	- Positivwortliste
ZIELFELD:<FN>	- Ergebnisfeld
SYNONYM: <WTX- o. Text-Datei>	- 2. Positivortliste
LÄNGE:x	- Wortlängenänderung (x:0..10 - Standard:2)
MIN:x	- minimale Wortlänge, die nicht mehr verändert wird - Standard:5)
TRENNER:(+)	- für Wortanalyse, Standard:bl.,:/"* (b=Leerz., += wird ergänzt)
1WORT, 2WORT, 3WORT, 4WORT	- xWORT-Analysen, Standard: 2WORT
ALLE	- bei gefund. Mehrworttermen Suche nach deren Einzelbegriffen
WORTSTAMM:x	- Differenzlänge für Vergleich (x:0..10, 0=volle Identität)
SORTIERT	- Ergebnis wird sortiert
ENGPLURAL	- Wortanalyse mit "S"-Endung
ÄwieA	- Führt Sonderzeichen auf ihre Grundzeichen zurück

SORTAUSSAGEN oder SORTAUSSAGEN2

Sortiert Aussagen eines Feldes

Sortaussagen: max 250 Aussagen a 250 Zeichen

Sortaussagen2: max 1000 Aussagen a 60 Zeichen

1.: MIDOS-Quelldatei

2.: Zieldatei

3.: SORTFELD:<feldname> TRENNER:<trennzeichen>

SORT

Sortiert physisch Datensätze

1.: MIDOS-Quelldatei

2.: Zieldatei

3.: SORTFELD:F1,F11,F12,..;F2,F21,..

F1 .. F5 - hierarchische Sortierung

F11,F12,..;F21,F22 ... - Ersatz-Sortfelder

Trennung hierarchisch, Felder mit Semikolon und Ersatzfelder mit Komma

Grenzen: max 5 hierarchische Felder

max 4 Ersatzfelder

§F1,§F2 ... - Numerische Sortierung

VARIABLE - Sort. mit max 250 Zeichen u. variabler Speicherung

GENAU - genauere Sortierung (200 Zeichen), sonst 100 Zeichen

TEXT/§TEXT - Sortiert Textdatei

ABPOS:x - nur für Textsort.:speichert ab x.Zeichen

MIDOSTRENNER - Filter für MIDOS-Sonderzeichen

LFDNUM:<feldname,abnummer> - laufendes Nummernfeld einfügen

SPLITT

Splittet MIDOS-Datei in kleinere

1.: MIDOS-Quelldatei

2.: MIDOS-Zieldatei (ohne Erweiterung)

3.: UMFANG:xxx SATZ:yyy

Splittung, wenn Zieldateiumfang xxx Byte oder die Satzanzahl yyy der Zieldatei überschritten wird

STOP

Job wird abgebrochen

1.: ---

2.: ---

3.: ---

SYNONYM

Automatische Synonymlistenbildung für Schlagwortvergabe

1.: MIDOS-Quelldatei

2.: Deskriptor-Textliste

3.: folgende Parameter mit Leerzeichen getrennt:

SUCHFELDER:FN1;FN2 - Zu durchsuchende Felder wie Titel, Referat ..

SYNONYM:<name> - Textdatei mit Ergebnis

MIDOS - Ergebnisliste im MIDOS-Format

LÄNGE:x - Wortlängenänderung (x:0..10 - Standard:2)

ÄwieA - Führt Sonderzeichen auf ihre Grundzeichen zurück

MARKDE - markiert Synonyme, die als Deskriptoren existieren mit "**"

TEXT_<progrname>

Textbearbeitungsfunktionen

- 1.: Text-Quelldatei
- 2.: Text-Zieldatei
- 3.: Parameter

Aufruf über folgende Programmnamen:

PROGNAME	Funktion	Parameter
TEXT_ANSI	nach ANSI	---
TEXT_ASCII	nach ASCII	---
TEXT_GROSS	in Gross-Bst.	---
TEXT_KLEIN	in Klein-Bst.	---
TEXT_LÖSCHE_AB	ab SW löschen	Suchwort
TEXT_LÖSCHE_BIS	bis SW löschen	Suchwort
TEXT_LÖSCHE_ANFANG	Anf.zeichen lö.	Zeichen
TEXT_LÖSCHE_ENDE	Endezeichen lö.	Zeichen
TEXT_MITSW	Zeile mit SW	Suchwort
TEXT_OHNESW	Zeile ohne SW	Suchwort
TEXT_DOPPEL	Dopplungsprüf.	GROSS
TEXT_ABGLEICH	2 Dat.vergleichen	2. Quelldatei GROSS NOIDENT
TEXT_MASKE	siehe MASKE	MASKE MITSW= OHNESW= MAKEISO

Erläuterung zur Maske:

MASKE mischt Maskentext mit Variable:

- *Z* - gesamte Zeile
- *Z[vonpos-bispos]* - Teile der Zeile (Zk mit "")
- *L* - Zeilenlänge (rechts formatiert, 6 Zeichen)
- *W* - Wortzahl (rechts formatiert, 3 Zeichen)

WAITWIN

Warten auf die Beendigung eines bereits geladenen Windowsprogrammes

- 1.: Fenstername oder Klasse der Fremdprogrammes
- 2.: ---
- 3.: ---

WINEXEC2

Führt ein Windowsprogramm aus und testet alle 15 Sec. auf Existenz des Fremdprogrammfensters

- 1.: Programmname (mit oder ohne EXE) einschließlich Parameter
- 2.: weitere Parameter
- 3.: Fenstername oder Klasse (s. Hilfsprogramme)

Beispiel: mwlist miausart

WINEXEC

Führt ein Windowsprogramm aus und hält MWUPDATE solange an

- 1.: Programmname (mit oder ohne EXE)
einschließlich Parameter
- 2.: weitere Parameter
- 3.: weitere Parameter

Beispiel: mwlist miausart

WORTPRUEF

Prüfe MIDOS-Datensätze über Prüfliste. Zur anschl. autom. Korrektur wird für den Manipulator die Datei AUTOKORR.MAN mit Ersatzwörtern nach einem Ähnlichkeitsalgorithmus von Dr. Lamprecht erzeugt.

- 1.: MIDOS-Datei
 - 2.: Prüfliste (WTX-Datei)
 - 3.: FELD:xx ähnlich:yy diff:zz exakt
- xx - Prüffeldname
yy - Ähnlichkeitswert für Ersatzwort (50..100), sonst Ersatzwort:"???"
zz - Längendifferenz des Ersatzwortes zum fehlerhaften Wort (0..10)
Standardwerte: ähnlich:70 diff:2
exakt - mit Berücksichtigung der Groß-/Kleinschreibung

WRITEAUSTAUSCH

Umsetzung MIDOS in ein Austauschformat mit Begrenzungszeichen

- 1.: MIDOS-Quelldatei
- 2.: Austausch-Textdatei
- 3.: EZ:x TZ:y SZ:abc nohead
 - EZ:x (x:Einschließungszeichen)
 - TZ:y (y:Trennzeichen)
 - SZ:abc (abc:Satzendezeichen)

nohead (keine Kopfzeile)

Hinweis: Es werden nur die Felder der Eingabebeschreibungsdatei MISCHABS des aktuellen Verzeichnisses berücksichtigt

WRITEDIR

Schreibt Dateinamen in eine Textdatei

- 1.: Verzeichnisname mit Suchmaske
 - 2.: Textdateiname
 - 3.: Schreibmaske SUBDIRS
- In "Schreibmaske" wird *DIR* und *FILE* ersetzt sonst vollständige Pfadangabe
SUBDIRS - mit Unterverzeichnisse

WRITEFILEDATE

Schreibt in eine Textdatei das Änderungsdatum einer Datei

- 1.: Name der Textdatei
- 2.: Dateiname für Änderungsdatum
- 3.:

WRITEFILE

Schreibt in eine Datei Zeichenketten

- 1.: Dateiname
- 2.: Dateiposition
- 3.: Zeichenkette

Existiert die Datei nicht, dann wird sie angelegt und die Zeichenkette ab Position Null geschrieben. Die Zeichenkette kann Steuerzeichen enthalten: #<ASCII-CODE> oder =CRLF

WRITEINFO

Schreibt in die Datei MWWTX.INI Infotext mit aktuellem Datum/Zeit

- 1.: Infotext
- 2.: ---
- 3.: ---

Schlüsselwörter im Infotext:

- *DATE* - aktuelles Datum
- *TIME* - aktuelle Uhrzeit

WTXRECH

INDEX-RECHERCHE in einer WTX-Datei und Kopieren der Treffer in die Zieldatei.

- 1.: WTX-Datei
 - 2.: MIDOS-Zieldatei
 - 3.: Textdatei mit einer Liste der gesuchten Einträge
- ANSI - Zieldatei im ANSI-Format

WTXTOHTML

MIDOS-Index-Dateien werden mit M2rech-Links in HTML-Listen geschrieben

- 1.: WTX-Datei
 - 2.: HTM-Datei
 - 3.: <überschrift> spalten:x grafik
- x: 1...3 Tabellenspalten
grafik: Linksybole mit Grafik-Dateien: L1.GIF,L1.GIF,L1.GIF,BACK.GIF,NEXT.GIF

ZFORMAT

Zeilenformatierung von MIDOS-Sätzen

- 1.: Quelldatei mit MIDOS-Satzendezeichen
 - 2.: Zieldatei
 - 3.: Einstellungen:
- Zeichen/Zeile: ZZ:xx (xx=10..100)
Wortumbruch: WU:x (x=1:Ja, x=0:Nein)

Einrückung: ER:xx (xx=0..30)

7.3 EXEC-Anweisungen für die MIDOS-UPDATE-Feldmanipulation

Das MWUPDATE-Kommando „FELD“ ändert Feldinhalte. Er erwartet als 3. Parameter eine Transformationstabelle mit folgendem Aufbau:

```
<quellfeldname1> <leerzeichen> <zielfeldnameY>
<quellfeldname2> <leerzeichen> <zielfeldnameY>
...
<quellfeldnameN> <leerzeichen> <zielfeldnameZ>
NWENDE
```

NWENDE speichert am Datensatzschluss die MIDOS-Endekennzeichen (&&).

Beispiel:

```
VERF  VERF2
TITEL  TITEL2
EK     EK2
NWENDE
```

oder

```
VERF
TITEL
EK
NWENDE
```

Existiert der Zielfeldname nicht, so wird der Quellfeldname als Zielfeldname verwendet. Weiterhin kann als Quellfeldname EXECx (x=1 ... 99) stehen. Das ist die Bezeichnung eines Bearbeitungsprogramms, das nach „NWENDE“ definiert werden muss.

Praxisbeispiel mit EXEC-Aufrufen:

	<i>Erläuterung</i>
EN EXTNR	Quellfeld = Zielfeld mit Wechsel des Feldnamens
VF VERF	
ST TITEL	
EK	Quellfeld = Zielfeld, Feldname bleibt gleich
DT	
EXEC1 DE2	Zielfeldinhalt wird in EXEC1 erzeugt und in DE2 gespeichert
EXEC2	Zielfeld wird in EXEC3 erzeugt, Feldname muss dort zugewiesen werden
NWENDE	Dokumentende automatisch übertragen
	Leerzeile
EXEC1:	Definition von der EXEC2-Prozedur
CA(DE,1,100,)	Kopiert aus DE die Einträge 1-100 mit Trennzeichen „ “
CC()	danach wird „ “ als Feldtrenner kopiert
CA(DE1,1,100,)	Kopiert aus DE1 die Einträge 1-100 mit Trennzeichen „ “
	Leerzeile
EXEC2:	Definition von der EXEC2-Prozedur
CC(X1:)	Feldname X1: wird gespeichert (weil oben nicht zugewiesen)
CS(RF,1,25)	Kopiert aus dem Feld RF die ersten 25 Zeichen
CC(^)	Das Feldende muss hier erzeugt werden (weil oben Feld nicht zugewiesen)

Erläuterung der EXEC-Kommandos

EXEC-Kommandos sind im Anschluß an die Transformationstabelle nach einer Leerzeile zu definieren. Es kann nur ein Kommando pro Zeile stehen. Hinter einem Kommando kann ein Kommentar stehen. EXEC-Prozeduren sind mit EXEC<nr> einzuleiten und mit einer Leerzeile abzuschließen. Fehlt eine EXEC-Prozedur wird beim Start des Transfers ein Syntaxfehler gemeldet.

Wichtig:

In EXEC-Prozeduren ist der Anwender selbst für die Erzeugung einer MIDOS-gerechten Syntax zuständig, wenn eine gültige MIDOS-Datenbankstruktur erzeugt werden soll. Das betrifft vor allem die Bereitstellung eines CRLF (#13#10 oder ^) am Ende eines Feldes, die Bereitstellung gültiger Dokumentendezeichen (z.B. &&), das Setzen gültiger Aussagentrennzeichen (z.B. |) und die Verwendung gültiger Feldnamen (1 bis 12 Zeichen plus Doppelpunkt).

Im folgenden gilt für die Schreibweise:

- { } steht für andere Schreibweise
- [] steht für wahlweise Angabe
- <> markiert Variable

Reservierte Wortsymbole sind in Grossbuchstaben geschrieben, obwohl Programm sowohl in Groß- als auch Kleinbuchstaben zulässig sind.

Kopieren eines Feldes

COPYALL(<feldname>,[<bedingung>])
{COPY(...)} {CO(...)}

- kopiert das gesamte Feld vom 1. Zeichen nach dem <feldanfangszeichen> (im allgemeinen ":") bis vor <feldende>, im allgemeinen CRLF (X'0D0A', #13#10) oder früher ";"
- <bedingung> wird weiter unten erläutert

Kopieren einer Konstanten

COPYCONST(<zeichenkette>,[&])
{CC(...)}

- kopiert eine Zeichenkette(<zeichenkette>)
- Die <bedingung &> bewirkt, das die Konstante nur kopiert wird, wenn bereits ein Wert (<feld> oder <zeichenkette>) in den Ausgabebereich kopiert wurde.
- Die max. Konstantenlänge beträgt 40 Zeichen. Längere Zeichenketten müssen aus mehreren Teilzeichenketten zusammengesetzt werden.
- Konstanten, die Zeichen aus der Syntax der Kommandos enthalten, sind in Apostroph einzuschliessen. Die Zeichen können auch in ihrem ASCII-Wert angegeben werden.

Beispiel:

CC(' ') --> ein Leerzeichen
CC(#32) --> auch ein Leerzeichen

Kopieren einer Teilzeichenkette

COPYSUBSTRING(<feldname>,<von>,<bis>,[<bedingung>],[*])
{CS(...)}

- kopiert eine Teilzeichenkette aus dem Feld <feldname> von der Positions <von> bis zur Position <bis>, wenn die Angaben Zahlen sind (1...32756)
- kopiert eine Teilzeichenkette aus einem Feld von der Positions <vonzeichenkette> bis zur Position <biszeichenkette>, wenn die von/bis-Angaben Zeichenketten sind und die Position <biszeichenkette> ungleich Null und größer als <vonzeichenkette> ist.
- <bedingung> wird weiter unten erläutert

Beispiel:

CS(EK,1,20) --> von Position 1 bis Position 20
CS(EK,Berlin,') vom 1. Auftreten des Wortes Berlin, bis zur

1. runden Klammer

Die Funktion liefert nur dann ein richtiges Ergebnis, wenn die "(" immer dem Wort "Berlin" steht, andernfalls ist das Ergebnis leer.

Kopieren von Aussagen

COPYAUSSAGE(feldname,von,bis,trennzeichen,[bedingung],[*])
{CA(...)}

- kopiert aus dem Feld <feldname> die Aussagen von einer Aussagennummer bis zu einer Aussagennummer
- <trennzeichen> ist der Aussagentrenner, der in der <quelldatei> als Trenner verwendet wird (Normal: |)

z.B.

COPYAUSSAGE(DE,3,3,|) dritte Aussage kopieren

COPYAUSSAGE(DE,2,2,|) zweite Aussage kopieren

COPYAUSSAGE(DE,1,1,|) erste Aussage kopieren

...

COPYAUSSAGE(DE,4,20,|) Aussage 4 bis 20 kopieren

Übersetzen von Aussagen

TRANSLATE(feldname,von,bis,trennzeichen)
[FILE:<dateiname>]

{TR(...)}

oder

TRANSLATEALL(feldname,von,bis,trennzeichen)
[FILE:<dateiname>]

{TRALL(...)}

- transformiert Aussageninhalte des Feldes <feldname> gemäß einer Translate-Tabelle
- bei TRANSLATEALL (bzw. TRALL) werden nicht in der Tabelle enthaltene <quellaussagen> unverändert als <zielaussage> übertragen
- von, bis definieren, von der wievielten Feldaussage bis zu welcher Feldaussage des Quellfeldes der TRANSLATE-Befehl ausgeführt werden soll.
- Trennzeichen definiert ein Zeichen, das zwischen die übersetzten Feldaussagen gesetzt werden soll.
- Die Tabelle ist mit einer Leerzeile (CRLF + Leertaste + CRLF) zu beenden.
- Liegt die Translate-Tabelle in einer externen Datei vor, muss diese mit

FILE:<dateiname>

unmittelbar in der Folgezeile von TRANSLATE aufgerufen werden.

<dateiname> umfasst entweder die komplette Laufwerks- und Verzeichnisangabe oder nur den Dateinamen. In letzteren Fall muss sich die Datei im Verzeichnis befinden, in dem auch die Transferdatei steht. Nach der Angaben von FILE... muss in jedem Fall eine Leerzeile stehen, bevor ein weiteres Kommando folgen kann.

Beispiel:

EXEC12:
TRANSLATE(CC,1,6,|)
FILE:CCKONK.TXT

CC(^)

Tabellenstruktur:

Die linke Seite der Translate-Tabelle enthält die <quellaussage>, die rechte Seite die <zielaussage>.

Die <quellaussage> und <zielaussage> in der Tabelle können mit Komma oder "^" (ASCII 94) getrennt werden.

Enthält eine Quell- oder Zielaussage ein zu erhaltendes Komma oder ein "^", dann sind diese Zeichen durch ihren ASCII-Wert auszudrücken. Enthält eine Quell- oder Zielaussage nur ein zu erhaltendes Komma, genügt es, die Tabellenelemente mit "^" zu trennen. Werte von <quellaussagen>, die nicht in der Tabelle enthalten sind, werden nicht übersetzt, sondern entfallen (vergl. aber TRANSLATEALL).

Achtung: Eine versehentlich in der Tabelle enthaltene Leerzeile beendet die Tabelle. Ein leerer Wert auf der rechten Tabellenseite führt zur Löschung.

Beispiel:

AAA^AAA Allgemeines
BBB^BBB Philosophisches, Beispiel
XXX^CCC Charakteristisches | AAA Allgemeines

Beachten Sie die letzte Zeile mit einer Besonderheit: Aus einem Code werden zwei Aussagen.

Formatieren von Feldern

FORMAT(feldname,links/rechts,anzahl,füllzeichen)
{FO(...)}

- formatiert das ganze Feld <feldname> links- oder rechtsbündig und gibt es aus
- <links>/<rechts> gibt die Formatierungsrichtung an
- <anzahl> gibt die Länge des formatierten Feldes an
- <füllzeichen> gibt das Zeichen an, daß zum Auffüllen überzähliger Zeichen benutzt wird (wenn Feldlänge kleiner <anzahl>).
- Ist die Feldlänge größer als <anzahl> wird ohne Warnung abgeschnitten. Wo abgeschnitten wird, bestimmt der Parameter links/rechts.
- Es empfiehlt sich, Zahlen mit Vornullen links aufzufüllen und Text mit Leerzeichen rechts.
- Hinweis: Durch das Aneinanderfügen von mehreren formatierten Feldern entsteht ein Grundformat für relationale Datenbanken.

Steuerung des Prozedurablaufs

Bedingte Sprunganweisungen

IF EXIST(<feldname>,[suchwort]) GOTO marke

- bedingte Programmverzweigung: Wenn ein <suchwort> angegeben ist, wird es im Feld <feldname> gesucht, sonst wird nur die Existenz des angegebenen Feldes überprüft. Ist EXIST = true, dann wird mit der ersten Anweisung nach der Marke <marke> fortgefahren - sonst wird als nächste Anweisung die nach dieser IF-Anweisung stehende abgearbeitet. Die Marke darf nur nach dieser Anweisung stehen, sonst wird Abarbeitung der EXEC-Prozedur abgebrochen. Existiert die Marke nicht, wird die EXEC-Prozedur abgebrochen.

IF NOT EXIST(<feldname>,[suchwort]) GOTO marke

- wie oben, es wird jedoch die Nichtexistenz des Feldes bzw. des <suchwortes> im Feld geprüft.

Unbedingte Sprunganweisung

GOTO marke

- unbedingte Verzweigung: Das Programm wird an der Stelle fortgesetzt, die mit der Marke <marke> gekennzeichnet ist. Wird die Marke nicht gefunden oder ist die Marke vor der GOTO-Anweisung, wird die EXEC-Prozedur abgebrochen.

EXIT

- Beendet die Abarbeitung einer EXEC-Prozedur, ohne dass das wirkliche Ende des Programmes erreicht sein muss.

Bedingungen für Kopieranweisungen

Als Bedingung ist eine max. 40 Zeichen lange Zeichenkette <zk> mit Operator einzugeben. Die Bedingung steuert die Ausführung der Kopieranweisungen:

= <zk>	- nur wenn <zk> in der Kopier-Zeichenkette enthalten ist
<> <zk>	- nur wenn <zk> nicht in der Kopier-Zeichenkette enthalten ist
< <zk>	- nur wenn die Kopier-Zeichenkette vor <zk> im Feld steht. 1)
> <zk>	- nur wenn die Kopier-Zeichenkette hinter <zk> im Feld steht 1)
VON<zk>	- nur wenn <zk> in der Kopier-Zeichenkette enthalten ist, kopiert wird ab 1. Zeichen nach <zk> bis Aussagenende. 2)
BIS<zk>	- nur wenn <zk> in der Kopier-Zeichenkette enthalten ist, kopiert wird ab 1. Aussagenzeichen bis ein Zeichen vor <zk> 2)
1) - nur für copysubstring	
2) - nur für copyaussage	

Die Parameter der Kommandos sind durch Kommata zu trennen. Ein <zk>-Parameter darf max. 40 Zeichen lang sein und ist in Hochkomma einzuschließen.

Steuerzeichen sind als ASCII-Code mit "#" einzugeben.

z.B.

```
#39 = Hochkomma ( ' )
^ = Zeilenwechsel (CRLF, #13#10)
#35 = Doppelkreuz ( # )
#13#10 = ^ = (CRLF)
#64 = kommerzielles "A" (@)
```

Zwischenfeld-Befehle

Alle bisherigen Anweisungen transportieren direkt Daten von einem Eingabefeld (meist Quelldatei) zu einem Ausgabefeld (in die Ausgabedatei oder auf den Bildschirm). Die folgenden Anweisungen dagegen arbeiten alle über ein Zwischenfeld. Die Arbeitsweise ist in etwa folgende: Ein oder mehrere Felder werden in das Zwischenfeld geladen, werden dort analysiert und bearbeitet und dann in das Ausgabefeld geschrieben. Die Zwischenfeldbearbeitung wird mit jeweils einer speziellen Anweisung eingeleitet und beendet. Zwischen diesen Anweisungen sind nur die im folgenden beschriebenen Anweisungen zugelassen.

Beginn und Ende der Zwischenfeldbearbeitung

ZFBEGIN(mode)

- Initialisiert das Zwischenfeld <ZF>, wenn mode>0, dann wird der Inhalt des <ZF> der letzten EXEC-Prozedur übernommen. Damit ist der EXEC-Aufruf gemeint, der in der Transformations-Tabelle als letzter vor dem mit mode>0 definierten EXEC-Prozedur steht. Die Reihenfolge der EXEC-Prozedur-Definitionen im Anschluss an die Transformations-Tabelle ist dagegen nicht von Bedeutung.

ZFENDE

- Schreibt die Daten des Zwischenfeldes in das Ausgabefeld

Anzeigen interner Variablen

SHOWVARS

- Zeigt die globalen Variablen VAR1 ... VAR9 an.
(Nur im Testmodus, hält die Ausgabe an, weiter mit Enter.)

SHOWZF

- Zeigt das Zwischenfeld an.
(Nur im Testmodus, hält die Ausgabe an, weiter mit Enter)

Kopieren in das Zwischenfeld

ZFCOPY(<feldname>)
 {ZFCO(<feldname>)}

- Kopiert das Feld <feldname> in das Zwischenfeld

Kopieren von Zeilen

ZFCOPYZEILE(<nummer>,[<trenner>])

- Kopiert die Zeichenkette Nummer <nummer> aus dem Quellsatz, die durch den Trenner <trenner> abgetrennt sind. Optional sind als Trenner die Zeilenwechselzeichen #13#10 (0D0AH) eingestellt.

Kopieren von Aussagen

ZFCOPYAUSSAGE(<feldname>,<von>,<bis>,<trennzeichen>)
 {ZFCA(...)}

- Kopiert die Feldaussage mit der Position <von> bis zur Feldaussage mit der Position <bis> des Feldes <feldname> in das Zwischenfeld.
- <von>,<bis> sind die Aussagennummern. Liegen mehrer Anweisung hintereinander, wird das Zwischenfeld kumuliert.
- <trennzeichen> werden zwischen die Aussagen gesetzt, sobald mehr als eine Aussage im Zwischenfeld landet.

Kopieren von Konstanten

ZFCOPYCONST(<zeichenkette>)
 {ZFCC(...)}

- Kopiert die Zeichenkette <zeichenkette> in das Zwischenfeld. Die Länge der Zeichenkette kann max. 30 Zeichen betragen. Längere Zeichenketten sind aus mehreren ZFCOPYCONST-Anweisungen zusammenzusetzen. ZFCC fügt immer an das Ende des Zwischenfeldes an!

Substituieren von Teilzeichenketten

ZFSUBST(<zeichenkette1>,<zeichenkette2>,[<von>],[<bis>])
 {ZFSU(...)}

- Ersetzt im Zwischenfeld die Zeichenkette <zeichenkette1> durch <zeichenkette2>
- <von>,<bis> sind optional und geben eine Anfangs- und Endeposition an, innerhalb derer die Ersetzung wirken soll

Substitution von Code

ZFCODE(<zeichenkette1>,<zeichenkette2>,[<von>],[<bis>])

- Tauscht im Zwischenfeld die Zeichen der Zeichenkette <zeichenkette1> durch die Zeichen der Zeichenkette <zeichenkette2> aus, die an der gleichen Position in beiden Zeichenketten stehen.
- <von>,<bis> - sind optional und geben die Anfangs- und Endeposition innerhalb derer die Umkodierung wirken soll.

Beispiele:

zfcode(A,a,1,1)

Aus jedem A wird ein a, wenn es auf Position 1 des Zwischenfeldes steht.

zfcode(ABCDEFGHIJKLMNQRSTUUVWXYZ|,abcdefghijklmnopqrstuvwxyz\$,1,var9)

Von Position 1 des Zwischenfeldes bis zum Zwischenfeldende (ausgedrückt durch die Variable var9, die die aktuelle Länge des Zwischenfeldes enthält) werden alle Grossbuchstaben zu Kleinbuchstaben.

Numerische Variable

VAR1 ... VAR8=y

- Verwendung von 8 globalen Variablen z.B. zur Bestimmung der Position von Substrings in der Funktion ZFPOS oder ZFPOS1

VAR9

- Enthält die aktuelle Zwischenfeldlänge, kann nur lesend verwendet werden.

Position einer Zeichenkette ermitteln

VARx=ZFPOS(<zeichenkette>,[<von>][,<bis>])

- Das Zwischenfeld wird nach <zeichenkette> durchsucht, als Ergebnis wird die Position der <zeichenkette> (linkes Zeichen) in VARx (x= 1..8) gespeichert. Null als Ergebnis bedeutet, die Zeichenkette ist nicht im Zwischenfeld enthalten.

- <von>, <bis> sind optional, fehlt die Angabe, wird als Standard Zwischfeldanfang und Zwischenfeldende (wie var9) angenommen.

oder

*VARx=ZFPOS1(<zeichenkette>,[<von>][,<bis>])

- Das Zwischenfeld wird nach den einzelnen Zeichen von Zeichenkette durchsucht. Die Position des ersten Auftretens eines dieser Zeichen wird in VARx (x= 1..8) gespeichert. Ist das Ergebnis 0, bedeutet das, dass keines der Zeichen im Zwischenfeld enthalten ist.

- <von>, <bis> sind optional, fehlt die Angabe, wird als Standard Zwischfeldanfang und Zwischenfeldende (wie var9) angenommen.

Beispiel:

var1=zfpos(0123456789,30,var9)

- ermittelt die Position der ersten Ziffer im Zwischenfeld, die nach der Position 30 steht

var1=zfpos('-',.,:;?!()&\$%"+')

- ermittelt die Position des ersten Sonderzeichens im Zwischenfeld

Löschen von Teilzeichenketten

ZFDELETE(<von>,<bis>)
{ZFDEL(...)}

- Löscht das Zwischenfeld vom Zeichen <von> bis zum Zeichen <bis> einschließlich dieser Zeichen; es wird also die Position des letzten zu löschenden Zeichens angegeben.

Beispiel:

zfdelete(var1,var9)
zfdelete(13,var2)
zfdelete(13,255)

ZFERASE(<von>,<anzahl>)

- Löscht im Zwischenfeld ab dem Zeichen auf Position <von> in der Länge, die durch <anzahl>-Zeichen bestimmt wird.

Zeichenketten einfügen

ZFINS(<zeichenkette>,<ab>)

- Fügt die Zeichenkette <zeichenkette> ab der Position <ab> in das Zwischenfeld ein

Programmverzweigung mit Variablenwerten

IF VARx=y GOTO - bedingte Programmverzweigung in Abhängigkeit
 IF VARx>y GOTO des Wertes y der Variablen VARx (x: 1..9)
 IF VARx<y GOTO
 IF VARx<>y GOTO - <> steht für "ungleich"

Aussagenanzahl ermitteln

VARx=MAXAUSSAGEN(<feldname>)

- Ermittelt die Aussagenanzahl für das genannte Feld (0 (Null) wenn Feld nicht existiert.)

VARx=MAXZFAUSSAGEN(ZF)

- ermittelt die Aussagenanzahl für das Zwischenfeld Aussagentrenner ist Standard; (|).

Programmschleifen

REPEAT <anweisungen> UNTIL

- Die zwischen REPEAT und UNTIL stehenden Anweisungen werden beliebig oft (bis zu 10000 mal) wiederholt. Über IF <bedingung> GOTO MARKE ist ein Abbruch möglich. Die Marke muss sich außerhalb der Schleife befinden
- Der Abbruch bei 10000 ist eine Sicherheitsmassnahme gegen unbeabsichtigte ununterbrechbare Programmschleifen, die durch falsche Endbedingungen entstehen können.

Beispiel:

```
VAR1=0
REPEAT          zählt nur die Variable bis 10
  VAR1=VAR1+1    Abbruchbeispiel
  IF VAR1>10 GOTO M1
UNTIL
:M1
ZFENDE
```

MEMO-Feld

Ein MEMO-Feld ist ein zusätzlicher Pufferspeicher, der über die Verarbeitung eines Dokumentes hinaus mit Daten gefüllt werden kann. Er wird vom Programmierer auf Grund von Bedingungen gezielt gefüllt und entladen. Die maximale Länge ist 500000 Zeichen.

Mit den im folgenden beschriebenen Funktionen wird möglich:

- Text (Aussagen) in das Zwischenfeld kopieren, das Zwischenfeld bearbeiten mit den Zwischenfeldkommandos, das Zwischenfeld in das MEMO (Memory-Feld) kopieren und ggf. kumulieren,
- das Zwischenfeld zu löschen und dann neuen Text in das Zwischenfeld zu laden, zu bearbeiten usw...
- irgendwann das angesammelte MEMO-Feld in das Zwischenfeld zurück zu kopieren und das ganze mit ZFENDE auszugeben.

ZFADDMEMO

- Kopiert das Zwischenfeld ZF vollständig in das MEMO-Feld; Wenn das MEMO-Feld bereits Werte enthält, wird der Inhalt des Zwischenfeldes kumulierend angefügt.

ZFCLEAR

- Löscht das Zwischenfeld ZF vollständig. VAR9 wird auf 0 gesetzt.

ZFRESTOREMEMO

- Kopiert den Inhalt des MEMO-Feldes in das Zwischenfeld. Das Zwischenfeld wird dabei vollständig überschrieben. Es sollte aber dennoch mit ZFCLEAR vor dem Rückspeichern gelöscht werden. Der MEMO-Inhalt wird nur gespeichert, wenn vor ZFENDE dieser Befehl steht.

ZFMEMOCLEAR

- Löscht MEMO vollständig. Der MEMO-Inhalt wird nur mit diesem Befehl gelöscht, bleibt also auch datensatzübergreifend erhalten, solange der Befehl ZFMEMOCLEAR nicht ausgeführt wird. Wenn EXEC-Prozeduren im Fehlerfall verlassen werden, ohne das MEMO-Feld zu löschen, wird das MEMO-Feld ggf. über mehrere Datensätze hinweg gefüllt.

Wenn mit einem MEMO-Feld gearbeitet wird, kann es sinnvoll sein, dieses am Anfang der EXEC-Prozedur zu löschen. MEMO-Felder bleiben bestehen und werden kumulativ gefüllt, bis sie explizit gelöscht werden.

Komplexbeispiel:

Alle Deskriptoren des Feldes DESKR, die eine REGISTERKENNUNG # haben, sollen nicht ausgegeben werden.

DESKR:Heisluftballon; #Wetterballon; #Luftdruckmessung; Drift

```

exec12:
zfbegin
if not exist(DESKR) goto f1
zfmemoclear
var1=zfmaxaussagen(DESKR)
var2=0
repeat
zfclear
var2=var2+1
if var2>var1 goto raus
zfcopyaussage(DESKR,var2,var2,;)
var3=zfpos1(#35,1,1)
if var3=1 goto schleife
zfcc(;)
zfaddmemo
:schleife
until
:raus
zfclear
zfrestorememo
zfmemoclear
zftrimm(; #32,; #32)
zfsu(, #32#124##32)
if var9<1 goto f2
zfins(DESKR:,1)
zfcc(^)
zfende
exit
:f2
cc('DESKR:FEHLER=ALLE DESKR MIT REGISTERKENNUNG')
cc(^)
exit
:f1

```

Ergebnis: DESKR:Heisluftballon; Drift