

fiasco

COLLABORATORS

	<i>TITLE :</i> fiasco	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		February 12, 2023

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	fiasco	1
1.1	Fiasco.guide	1
1.2	Einführung	2
1.3	Voraussetzungen	2
1.4	Installation	2
1.5	Der GZZ-Modus	3
1.6	features	3
1.7	giftware	4
1.8	disclaimer	4
1.9	filelist	4
1.10	Quick Start	6
1.11	Erstellen einer Datenbank	6
1.12	Suchen in einer Datenbank	7
1.13	Suchmuster	8
1.14	Unscharfe Suche	8
1.15	Suchen mit ARexx	8
1.16	Zählen	8
1.17	Ersetzen	9
1.18	Filter	9
1.19	Alle Suchmuster	9
1.20	Relationen	10
1.21	Relations-Checkliste	11
1.22	Feldtypen	11
1.23	Standard-Attribute	12
1.24	Der String-Feldtyp	13
1.25	Der Integer-Feldtyp	13
1.26	Der Float-Feldtyp	14
1.27	Der Boolean-Feldtyp	14
1.28	Der Cycle-Feldtyp	15
1.29	Der Slider-Feldtyp	15

1.30	Der Datums-Feldtyp	16
1.31	Der Time-Feldtyp	17
1.32	Der Extern-Feldtyp	17
1.33	Der Datatypes-Feldtyp	18
1.34	Der Text-Feldtyp	18
1.35	Grafische Benutzer Oberfläche (GUI)	19
1.36	Service Window	20
1.37	Neu (Add)	20
1.38	Löschen (Delete)	21
1.39	<	21
1.40	<	21
1.41	>	21
1.42	>	22
1.43	Aktives Projekt	22
1.44	Status	22
1.45	Feldtyp	22
1.46	Maske	23
1.47	list	23
1.48	Records	23
1.49	Unterlegung	24
1.50	Maskendehnung	24
1.51	Record-Modus	24
1.52	Masken-Modus	24
1.53	Menus	24
1.54	Projekt/Neu (Project/New)	28
1.55	Projekt/Leeren (Project/Erase)	28
1.56	Projekt/Öffnen (Project/Open...)	28
1.57	Projekt/Optionen... (Project/Options...)	28
1.58	Projekt/Statistik... (Project/Statistic...)	29
1.59	Projekt/Relationen neuladen (Project/Reload Rels)	29
1.60	Projekt/Speichern (Project/Save)	29
1.61	Projekt/Speichern als... (Project/Save As...)	30
1.62	Projekt/Über Fiasco... (Project/About...)	30
1.63	Projekt/Beenden (Project/Quit)	30
1.64	Records/Hinzufügen (Records/Add Record)	30
1.65	Records/Duplizieren (Records/Duplicate Record)	31
1.66	Records/Entfernen (Records/Remove Record)	31
1.67	Records/Voriger (Records/Previous)	31
1.68	Records/Nächster (Records/Next)	32

1.69	Records/Erster (Records/First Record)	33
1.70	Records/Letzter (Records/Last Record)	33
1.71	Records/Gehe zu... (Records/Goto...)	33
1.72	Feldtyp	34
1.73	Felder/Feld Hinzufügen... (Fields/Add Field...)	35
1.74	Felder/Feld Entfernen (Fields/Remove Field)	35
1.75	Felder/Feld Ändern... (Fields/Edit Field...)	36
1.76	Felder/Relationen ändern... (Fields/Edit Relations...)	36
1.77	Liste/Spalte verbergen (List/Hide column)	37
1.78	Liste/Spalte sichtbar machen... (List/Show column...)	37
1.79	Liste/Alle Spalten sichtbar (List/Show all columns)	37
1.80	Liste/Liste neu berechnen (List/Recalc List)	38
1.81	Vergleichen/Suchen... (Compare/Find...)	38
1.82	Vergleichen/Weitersuchen (Compare/Find next)	38
1.83	Vergleichen/rückwärts Suchen (Compare/Find previous)	39
1.84	Vergleichen/Ersetzen... (Compare/Replace...)	39
1.85	Vergleichen/Zählen... (Compare/Count...)	39
1.86	Vergleichen/Sortieren... (Compare/Sort...)	40
1.87	Vergleichen/Filter Ändern... (Compare/Edit Filter...)	40
1.88	Vergleichen/Filter An? (Compare/Use Filter?)	40
1.89	Kontrolle/Record-Modus (Control/Record Mode)	41
1.90	Kontrolle/Masken-Modus (Control/Mask Mode)	41
1.91	ServiceFenster (ServiceWindow)	41
1.92	ListenFenster (ListWindow)	42
1.93	ARexx-Debug	42
1.94	Einstellungen/Icons Erzeugen? (Settings/Create Icons?)	42
1.95	Einstellungen/Backups Erzeugen? (Settings/Create Backups?)	42
1.96	Einstellungen/Relationen schreiben? (Settings/Write Relations?)	43
1.97	Einstellungen/Relationen aktualisieren? (Settings/Update Rels?)	43
1.98	GZZ-Fenster? (GZZ Windows?)	43
1.99	Einstellungen/Sicherheits-Requester? (Settings/Security-Reqs?)	43
1.100	Einstellungen/Auto-Öffnen ServiceWin? (Settings/Auto-Open ServiceWin?)	44
1.101	Einstellungen/ServiceWin dynamisch? (Settings/Dynamic ServiceWin?)	44
1.102	Einstellungen/Schrift auswählen... (Settings/Choose Font...)	44
1.103	Einstellungen/Einstellungen speichern (Settings/Save Settings)	44
1.104	Einstellungen/Einstellungen speichern als... (Settings/Save Settings as...)	44
1.105	Einstellungen/Einstellungen laden... (Settings/Load Settings...)	44
1.106	Benutzer/Ändern... (User/Edit...)	45
1.107	Alle Requester	45

1.108Der Feldrequester	45
1.109Der Suchrequester	46
1.110Der Ersetzen-Requester	47
1.111Der Zähl-Requester	48
1.112Der Sortier-Requester	49
1.113Der Filter-Requester	49
1.114Der Usermenu-Requester	50
1.115Der Options-Requester	51
1.116Goto-Requester	51
1.117Der Relations-Requester	52
1.118Spalte anzeigen-Requester	53
1.119arexx	53
1.120Allgemeines zu ARexx mit Fiasco	54
1.121Index der ARexx Kommandos	55
1.122Index der GUI-Abhängigen ARexx-Kommandos	56
1.123Index der GUI-Unabhängigen ARexx-Kommandos	58
1.124f_openservicewin	59
1.125f_closeservicewin	60
1.126f_newproject	60
1.127f_openprojectreq	61
1.128f_openproject	61
1.129f_saveproject	62
1.130f_saveprojectreq	62
1.131f_aboutreq	63
1.132f_optionsreq	63
1.133f_setdeffontreq	63
1.134f_findfirst	64
1.135f_findnext	65
1.136f_findprev	66
1.137f_findreq	66
1.138f_setsearchpat	67
1.139f_setsearchfield	67
1.140f_addfieldreq	68
1.141f_gotoprevrec	68
1.142f_gotonextrec	69
1.143f_gotofirstrec	69
1.144f_gotolastrec	70
1.145f_quit	70
1.146f_gotorec	71

1.147f_gotorecreq	72
1.148f_setfieldcont	72
1.149f_getfieldcont	73
1.150f_addrrecord	73
1.151f_remrecord	74
1.152f_setmode	74
1.153f_locate	75
1.154f_countrecs	75
1.155f_sortreq	76
1.156f_countreq	76
1.157f_openlist	77
1.158f_closelist	77
1.159f_filterreq	78
1.160f_duprec	78
1.161f_getrecnum	78
1.162f_savesettings	79
1.163f_clearproject	79
1.164f_getprojname	80
1.165f_getprojfullname	81
1.166f_selectproj	81
1.167f_usercommand	82
1.168f_isvirgin	82
1.169f_makevirgin	83
1.170f_	83
1.171Beispiel-Projekte	83
1.172Adress-Verzeichnis	84
1.173Datatypes Demo	84
1.174Stammbaum	84
1.175Videos	85
1.176Anpassung an Sprachen	85
1.177Wie man mich erreichen kann	85
1.178bugs	85
1.179todo	86
1.180index	87

Chapter 1

fiasco

1.1 Fiasco.guide

Fiasco RELEASE 1.0

von Nils Bandener

Einführung

Voraussetzungen

Installation

Features

Quick-Start

Erstellen einer Datenbank

Suchen in einer Datenbank

Relationen

GUI

Feldtypen

Liste

ARexx

Beispiel-Projekte

Anpassung an andere Sprachen

Index

Rechtliches

Giftware
meine Adresse
Bugs
Was es noch zu tun gibt
Copyright, Disclaimer usw.

1.2 Einführung

Fiasco ist ein neuer Kandidat unter der Masse von Datenbanken für den Amiga. Ursprünglich wollte ich nur ein kleines Programm schreiben, das einem ein paar Vokabeln abhört, die man vorher eingegeben hat. Später kam dann noch die Möglichkeit hinzu, mehr als nur zwei Felder, also Frage und Antwort, zu erstellen. Ab einem gewissen Entwicklungsstadium war dieses Programm einer Datenbank schon so nahe, daß es nur ein paar Änderungen bedurfte, um daraus schon eine Datenbank zu machen. Seitdem (Januar 95) entwickle ich an diesem Meisterstück der Programmierkunst. ;-)

In den Grundlagen unterscheidet sich Fiasco kaum von anderen Datenbanken. Zwar unterstützt Fiasco keine hierarchischen Datenbankstrukturen (wie z.B. AmigaBase), hat dafür aber einen kleinen Teil für Relationen (Ich habe bisher nur von relationalen Datenbanken gelesen, so kann ich nicht 100%ig garantieren, daß ich das Prinzip richtig implementiert habe. Auf jeden Fall macht Fiasco sowas ähnliches :-). Außerdem kann Fiasco über AREXX gesteuert werden. So können außerdem Programme einzelnen Feldern zugeordnet werden.

Die Maske von Fiasco wird nicht durch eine Grafikdatei definiert, sondern mit internen Grafiken erstellt. So können Fiasco-Masken mit jedem nicht-proportionalen Font benutzt werden. Die Feldtypen, die in einer **Maske** benutzt werden können, sind breit gesät. Am besten gefällt mir der Datatypes-Feldtyp, mit dem man Grafiken, Animationen, Texte usw. direkt in die Maske einbauen kann.

Neben der Maske kann man die Daten in der **Liste** anzeigen. Diese ist wie die Maske frei konfigurierbar. Die Daten können in der Liste leider noch nicht verändert werden.

Das **Suchsystem** von Fiasco unterstützt unscharfe Suche und Platzhalter. Unscharfe Suche bedeutet, daß nicht nach Gleichheit, sondern nach Ähnlichkeit getestet wird. Die Toleranzschwelle, ab der Fiasco einen Eintrag als ähnlich ansieht, kann frei eingestellt werden. Die Platzhalter ähneln den AmigaDOS- Pattern-Matching-Symbolen; zur Zeit werden #? (eine unbestimmte Zahl unbekannter Zeichen) und ? (ein unbekanntes Zeichen) unterstützt.

Außerdem stehen Sortier-, Filter-, und Zählfunktionen zur Verfügung, die allesamt mit dem Suchsystem verwandt sind und so - kennt man einmal das Suchsystem - keine größeren Probleme darstellen sollten.

1.3 Voraussetzungen

Um Fiasco starten zu können braucht man mindestens einen Amiga mit mindestens 1 MB RAM (512 KB könnte auch gehen, würde aber bald knapp werden) und mindestens OS 2.04 (37.175). Ab OS 2.1 (38.x) wird die Locale.library unterstützt. Der **Datatypes-Feldtyp** ist erst ab Amiga OS 3.0 (39.x) verfügbar. Die Hilfs-Funktion benötigt die amigaguide.library version 34 oder höher. Ab Amiga OS 3.0 ist diese im Lieferumfang inbegriffen, falls man noch 2.0 oder 2.1 hat, kann man sie z.B. auf der Fish-Disk 9?? finden. (Aber da Sie ja inzwischen dieses Dokument lesen, kann ich wohl davon ausgehen, daß sie AmigaGuide bereits haben.) Obwohl Fiasco grundsätzlich auf allen Prozessoren läuft, kann bei bestimmten Situationen ein simpler 68000er leicht in die Knie gezwungen werden (besonders bei der Benutzung von mehreren **GZZ-Fenstern**)

1.4 Installation

Klicken Sie einfach zweimal auf das Installer Icon Ihrer Lieblingssprache im Verzeichnis "Install". Sie werden dann instruiert, was sie zu tun haben.

Falls Sie den Commodore Installer nicht besitzen, brauchen sie das "Fiasco" Verzeichnis nur irgendwo hin kopieren. Falls Sie wollen, können Sie die Catalogs nach locale:catalogs kopieren, was aber nicht notwendig ist. Weiterhin können Sie die unbenutzten Sprachen in "Documentation" und "Examples" löschen und die verbleibenden Dateien in die übergeordneten Verzeichnisse schieben. Die Dateien in "Development" und "Install" werden für den normalen Gebrauch nicht benötigt.

1.5 Der GZZ-Modus

Ist dieser Modus unter Settings/Window Mode aktiviert, werden alle Projekt-Fenster im sogenannten "Gimme-Zero-Zero"-Modus geöffnet. Bei solchen Fenstern wird eine separate Layer für die Fenster-Balken erzeugt. Dies verhindert, daß Gadtools mit seinen Grafiken die Fenster-Balken zerstört. Der Nachteil ist, daß dieser Modus sehr Ressourcen-Intensiv ist und so die kleineren Modelle leicht in die Knie zwingen kann.

1.6 features

Das kann Fiasco:

- o Mehrere Projekte können gleichzeitig im Speicher gehalten werden. Die Anzahl ist nur durch den freien Systemspeicher begrenzt.
 - o Masken verhalten sich genau so wie andere GUIs
 - o Masken, Listen und Requester sind voll Fontsensitiv.
 - o Wahre Feldtypen-Vielfalt. Neben String, Integer und Float gibt es noch Cycle, Boolean, Slider, Datum, Zeit, Extern und Datatypes.
 - o Unterstützung von
Datatypes-Feldern
. Somit können auch
Bilder, Texte usw. direkt in die Maske integriert werden.
 - o ARexx-Port, mit dem das Programm gesteuert oder erweitert werden kann. Der ARexx-Port wird auch benutzt, um einzelnen Feldern Programme zuzuordnen.
 - o Frei konfigurierbares "User-Menu", von dem CLI-Programme und ARexx-Scripts gestartet werden können.
 - o Die Suchfunktion erlaubt sowohl Suchmuster als auch "unscharfe Suche"
 - o Sehr flexibles Listen-Fenster, in dem Einträge versteckt und in der Position und Größe verändert werden können.
 - o Einfaches Relation-Handling
-

1.7 giftware

Fiasco ist Giftware, das bedeutet, daß jeder Benutzer von Fiasco mir eine kleine Anerkennung für den Programmieraufwand zukommen lassen sollte. Dies kann Geld, irgendein kleines Geschenk (CD-Roms, Bücher, etc.) oder auch nur eine Postkarte sein (Oder auch nichts, wenn man meint, sein Geld für Besseres gebrauchen zu können).

Hier ist nochmal meine Adresse:

Nils Bandener
 Dekanatsgasse 4
 34369 Hofgeismar

1.8 disclaimer

Nutzungsbedigungen

Das Programm "Fiasco" und zugehörige Daten werden "wie sie sind" zur Verfügung gestellt. Der Autor übernimmt keinerlei Gewährleistung für Fehlerfreiheit, Genauigkeit, Verlässlichkeit oder Freiheit von sonstigen Mängeln der oben genannten Programme und Daten, weder ausdrücklich, noch stillschweigend. In keinem Falle kann ich für irgendwelche Schäden, die von diesem Programm verursacht wurden verantwortlich gemacht werden.

"Fiasco" ist nicht Public Domain, ich behalte mir jegliche Urheberrechte vor. (Fiasco Copyright © 1995 Nils Bandener)

"Fiasco" darf frei verbreitet werden, solange folgende Bedingungen erfüllt sind:

- Das Programm-Paket muß komplett sein. Fiasco besteht in der Release 1.0 aus den Dateien, die hier aufgelistet sind.
- Fiasco darf ohne schriftliche Genehmigung des Autors nicht für kommerzielle Zwecke verkauft werden. Dies schließt den Vertrieb von Fiasco zu überhöhten Preisen ein. Es dürfen lediglich Kosten für Medien und Kopieren erhoben werden. Der Vertrieb auf CD-Roms ist gestattet, solange der Preis nicht den Preis der "Fresh Fish"-CD-Rom von Fred Fish nicht übersteigt.

1.9 filelist

```
Fiasco_1.0 (dir)
  Development (dir)
    fiasco.cd          fiasco.cd.info
    fiasco.ct          fiasco.ct.info
    Fiasco_NewStrings Fiasco_NewStrings.info
  Databases (dir)
    English (dir)
    PD-Disks (dir)
```

```

        Disks.fdb
        myrequestfile
        ReadFish.rexx.info
    FamilyTree (dir)
        families.fdb
        persons.fdb
    Addresses (dir)
        Addresses.fdb
        Countries.fdb
    Videos (dir)
        CalcLen.rexx
        Films.fdb.info
        Tapes.fdb.info
    DatatypesDemo (dir)
        AmigaWorld.ilbm
        DatatypesDemo.fdb
        Hallelujah.8svx
Addresses.info
FamilyTree.info
Videos.info
Deutsch (dir)
    PD-Disks (dir)
        Disks.fdb
        myrequestfile
        ReadFish.rexx.info
    FamilyTree (dir)
        families.fdb
        persons.fdb
    Addresses (dir)
        Addresses.fdb
        Countries.fdb
    Videos (dir)
        CalcLen.rexx
        Films.fdb.info
        Tapes.fdb.info
    DatatypesDemo (dir)
        AmigaWorld.ilbm
        DatatypesDemo.fdb
        Hallelujah.8svx
Addresses.info
FamilyTree.info
Videos.info
Deutsch.info
Catalogs (dir)
    Deutsch (dir)
        fiasco.catalog
Documentation (dir)
    Deutsch (dir)
        fiasco.guide
    English (dir)
        fiasco.guide
Deutsch.info
Install (dir)
    Deutsch.info
    Install
Databases.info
Documentation.info
        Disks.fdb.info
        ReadFish.rexx
        families.fdb.info
        persons.fdb.info
        Addresses.fdb.info
        Countries.fdb.info
        Films.fdb
        Tapes.fdb
        AmigaWorld.ilbm.info
        DatatypesDemo.fdb.info
    DatatypesDemo.info
    PD-Disks.info
        Disks.fdb.info
        ReadFish.rexx
        families.fdb.info
        persons.fdb.info
        Addresses.fdb.info
        Countries.fdb.info
        Films.fdb
        Tapes.fdb
        AmigaWorld.ilbm.info
        DatatypesDemo.fdb.info
    DatatypesDemo.info
    PD-Disks.info
    English.info
        fiasco.guide.info
        fiasco.guide.info
    English.info
    English.info
    Development.info
    Fiasco

```

Fiasco.info

Install.info

1.10 Quick Start

Dies sind die wichtigsten Dinge, die man wissen muß, um mit Fiasco zu arbeiten:

- o Das Programm startet man am besten über das Programm- oder ein Projekt-Icon.
- o Es gibt zwei Arbeits-Modi in Fiasco, einen, in dem man die Records verändern kann, und einen anderen, in dem man die Maske verändern kann. Man steuert sie über
 - Kontrolle/RecordModus
 - bzw.
 - Kontrolle/MaskenModus
 - .
- o Das Service-Fenster erleichtert das Arbeiten mit Fiasco, wenn man sich noch nicht mit Shortcuts u.ä. auskennt. Man öffnet es über
 - Kontrolle/ServiceFenster
 - . ACHTUNG: Die Funktionen der Gadgets variieren in den Modi.
- o Die Liste, die man über Kontrolle/ListFenster erreicht, kann verändert werden, indem man in die Titeln der Liste klickt. Klickt man einen Titel einmal an, so wird er aktiviert. Über das Menu "List" kann man nun diverse Sachen mit der Spalte machen. Wenn man am rechten Rand eines Titels klickt, und die Maustaste gedrückt hält, kann man die Breite einer Spalte variieren. Falls man in den restlichen Raum des Titels klickt, und die Maustaste gedrückt hält, kann man die Position der Spalte verändern.
- o Bestimmte Einstellungen zum Projekt, z.B. zum Masken-Layout kann man über den Menüpunkt
 - Projekt/Optionen
 - einstellen.
- o Bei sonstigen Problemen kann man einfach das Menu anwählen und bevor man die rechte Maustaste losläßt, die Help-Taste drücken.

1.11 Erstellen einer Datenbank

Eine Datenbank besteht aus zwei Komponenten: Einmal die Daten an sich, die in Records aufgeteilt werden. Die Art der Daten wird im zweiten Teil definiert, der Maske. Eine Maske besteht aus einer beliebigen Anzahl von Feldern, die durch ihre Typen bestimmen, welche Art von Daten sie speichern sollen.

Um eine Maske zu erstellen muß man zuerst in den Masken-Modus, indem man den Menüpunkt **Kontrolle/Masken-Modus** aktiviert. Nun erscheint in der Maske ein Cursor, mit dem man die Position des Feldes festlegen kann, das man erstellen will. Den Typ des Feldes, das erzeugt werden soll, kann man über das Menu **Felder/Typ** festlegen. Die verschiedenen Feldtypen werden im Kapitel **Feldtypen** erklärt. Weiterhin läßt der aktuelle Feldtyp sich noch über das **Service-Fenster** und per Tastaturshortcuts einstellen (siehe **Menu**)

Nach diesen Einstellungen kann man mit dem Menüpunkt **Feld Hinzufügen...** ein neues Feld erstellen. Zuerst öffnet sich der **Feldrequester** . Die einzelnen Gadgets im Feldrequester sind vom Feldtyp abhängig und bei den jeweiligen **Typdokumentationen** erklärt. Es reicht übrigens nicht, nur Ok anzuklicken. Bestimmte Attribute, wie z.B. die Id müssen vom Benutzer immer neu angegeben werden. Sollten alle Attribute gültig sein, schließt sich der Requester und das jeweilige Feld erscheint in der Maske.

Nachträglich können im Nachhinein alle Attribute des Feldes bis auf den Feldtypen an sich geändert werden. Die Position läßt sich einfach mit der Maus verändern. Wenn man ein Feld auswählt und die Maustaste gedrückt hält erscheint einmal ein gefülltes Rechteck, das mit der Maus mitgeht und ein anderes, das die genaue neue Position zeigt. Läßt man die linke Maustaste los, wird das Feld an der neuen Stelle abgelegt, soweit keine anderen Felder im Weg sind.

Der Feldrequester läßt sich auch wieder aufrufen, indem man entweder auf ein Feld doppelt klickt, oder ein Feld aktiviert und den Menüpunkt **Felder/Feld ändern...** anwählt. Hier lassen sich wiederum alle Parameter frei verändern. Der Id-Wert sollte jedoch nur mit Vorsicht verändert werden, da andere Fiasco-Projekte oder ARexx-Scripts unter Umständen noch die alten Werte eingestellt haben und die Felder nicht mehr finden. Falls man die maximale Zeichenanzahl von String, Extern oder Datatypes-Feldern verkleinert, wird zuerst geprüft, ob Daten eventuell verloren gehen würden und eine Warnung ausgegeben.

Über den Menüpunkt **Felder/Feld entfernen** lassen sich Felder wieder entfernen. Vorsicht: Falls die Option **Sicherheits-Requester** nicht aktiviert ist, werden alle in diesem Feld gespeicherten Daten sofort aus dem RAM gelöscht. Beim nächsten Abspeichern werden auch die auf der Diskette verbliebenen Daten dieses Feldes vernichtet.

Im **Options-Requester** lassen sich weitere Parameter für das aktuelle Projekt, wie **Maskendehnung** , Name des Autors usw. angeben.

Der Menüpunkt **Felder/Relationen ändern** arbeitet ähnlich wie **Feld ändern** . Hier kann man jedoch Parameter für **Relationen** einstellen.

Ist die Maske vollständig kann man wieder in den Record-Modus zurückgehen (**Kontrolle/Record-Modus**) und die eigentlichen Daten erstellen. Mit dem Menüpunkt **Records/Hinzufügen** kann man neue Records erstellen, in denen am Anfang nur die Startwerte enthalten sind. Die Felder lassen sich nun wie normale Gadgets für die Dateneingabe benutzen.

1.12 Suchen in einer Datenbank

Die GUI-Schnittstelle zur Standard-Suchfunktion von Fiasco bildet der **Such-Requester** , den man über den Menüpunkt **Vergl./Suchen** erreicht. Hier kann man auswählen, auf welches Feld die Suche bezogen sein soll, ein Suchmuster angeben und Parameter zur unscharfen Suche angeben. Das Suchmuster kann einem gewöhnlichen Eintrag entsprechen. Bei den String-, Integer-, Float-, Date- und Timefeldtypen entspricht dies genau dem Inhalt des eigentlichen Felds. Falls sich die Suche auf ein Boolean-Feld bezieht, so kann man entweder "TRUE" oder "1" eingeben, falls man nach einem aktiven Feld suchen will, oder "FALSE" oder "0", falls man nach einem inaktiven Feld suchen will. Cycle-Felder lassen als Eingabe entweder eine Auswahlmöglichkeit als Klartext zu, oder die Nummer einer Auswahlmöglichkeit (von 0 aufwärts). Slider-Felder lassen wiederum nur eine Zahl an sich zu. Extern- und Data- typesfelder können nur über den Filenamen gesucht werden. Eine Suche, die sich auf den Inhalt der Dateien bezieht, ist über die eingebaute Suchfunktion nicht möglich.

Mit den Gadgets am unteren Rand kann man die Suche starten. Wenn ein passender Eintrag gefunden wird, wird der entsprechende Record automatisch aktiviert. Die Suche läßt sich hinterher ohne Umweg über den Suchrequester mit den Menüpunkten **Vergl./weiterrufen** und **Vergl./rückwärts suchen** fortsetzen.

Suchmuster

unscharfe Suche

Suchen mit ARexx

Zählen

Ersetzen

Filter

1.13 Suchmuster

Neben Eingaben im Klartext sind auch bei bestimmten Feldtypen Suchmuster zulässig. Der Stringfeldtyp erlaubt die Benutzung von AmigaDOS-ähnlichen Mustern. ? steht für ein unbekanntes Zeichen. "?iasco" würde also nach Aiasco, Biasco, Ciasco, Iiasco, @iasco, usw. suchen. "?????" würde nach Einträgen suchen, die 4 Buchstaben lang sind. #? steht für null oder mehr unbekannte Zeichen. A#? könnte also "Amiga", "Afrika", "A", "ABCD" bedeuten. ?#? sucht nach allen Einträgen, die nicht leer sind. Es ist nur ein "#?" in einem Suchmuster erlaubt. Wie auch unter AmigaDOS können auch diese Muster "escaped" werden, wenn man nach Einträgen suchen will, die genau den Mustern entsprechen. Um dies zu tun, muß man den jeweiligen Zeichen ein ^ (ALT-Ä) voranstellen.

Integer- und Sliderfelder unterstützen auch zusätzliche Muster. Folgende sind möglich: > x, < x, >= x, <= x, != x. "x" steht hier für eine beliebige ganze Zahl. ">" sucht nur nach Zahlen größer als x, ">=" nur nach Zahlen größer oder gleich x, "<" nur nach Zahlen kleiner als x und "<=" nur nach Zahlen kleiner oder gleich x. "!=" sucht nur nach Zahlen, die ungleich x sind. Ein Muster wie "==" , also "gleich" gibt es nicht, da dies durch den Wert ohne jedes Muster dargestellt wird.

Die von einem Feldtyp unterstützten Muster sind auch nochmal bei den Dokumentationen der jeweiligen **Feldtypen** aufgeführt.

Außerdem ist eine **Zusammenfassung aller Suchmuster** verfügbar.

1.14 Unscharfe Suche

Die unscharfe Suche bietet die Möglichkeit, nach Strings zu suchen, wobei nicht nur nach gleichen Einträgen gesucht wird, sondern nach ähnlichen Einträgen. Wie tolerant die Funktion dabei ist, kann man unter "faktor" einstellen. 0 entspricht der niedrigsten Toleranz, dies gleicht der normalen Suche, ist jedoch etwas langsamer. 100 bedeutet, daß alle Einträge als "ähnlich" angesehen werden. (Besser: "sollte bedeuten", aber die Funktion ist bei 100 dennoch sehr tolerant)

Die **Zähl-Funktion** eignet sich hervorragend, um mit der Unscharfen Suche etwas zu experimentieren.

1.15 Suchen mit ARexx

Auch mit **ARexx** können Suchen durchgeführt werden. Das Kommando **F_SetSearchPat** setzt das Suchmuster (im Requester unter Pattern). Mit **F_SetSearchField** kann man das Feld, auf das sich die Suche bezieht festlegen (im Requester unter Field). Nun kann man mit **F_FindNext** bzw. **F_FindPrev** nach Übereinstimmungen suchen. Im Gegensatz zu der GUI-Suchfunktion wird nun jedoch nicht der jeweilige Record aktiviert, sondern die Nummer des Records in der Variablen RESULT zurückgegeben. Mit der kann man nun machen, was man will, z.B. mit dem Kommando **F_GotoRec** den Record aktivieren.

Die Einstellungen, die über die Kommandos **F_SetSearchPat** und **F_SetSearchField** gemacht wurden, werden übrigens hinterher auch bei der GUI-Suche weiterbenutzt.

Über **ARexx** kann man so auch eine Suchfunktion erstellen, die auch mehrere Felder unterstützt.

1.16 Zählen

Über den Menüpunkt **Vergl./Zählen** öffnet man einen dem Suchrequester sehr ähnlichen Requester, der sich genaugenommen nur durch Titel, Größe und die Gadgets am unteren Rand unterscheidet. Hier kann man wiederum Suchmuster, Bezugsfeld und die Toleranz für die unscharfe Suche angeben. Falls man dann "Ok" wählt, werden die Übereinstimmungen gezählt und das Ergebnis angezeigt. Natürlich werden auch hier Muster und Toleranz mit einbezogen und beeinflussen so das Ergebnis. So kann man sich auch mal das Verhalten des **Toleranz-Faktors** genauer betrachten.

1.17 Ersetzen

Compare/Replace... erlaubt es, bestimmte Werte durch einen anderen zu ersetzen. Auch hier sind Suchmuster möglich, es wird jedoch immer nur ein Wert eingesetzt. Das Gadget mit dem Titel "Replacement" nimmt den Wert auf, der neu eingesetzt werden soll. Falls das Gadget "Confirm" aktiviert ist, wird ihnen immer der jeweilige Record gezeigt und Sie werden gefragt, ob sie wirklich den alten Wert ersetzen wollen.

Achtung: Mit einem falschen Muster (z.B. #?) kann man schnell viele Einträge löschen bzw. unbrauchbar machen!

1.18 Filter

Compare/Filter öffnet auch den Such-Requester, in dem man auch wieder die altbekannten Parameter angeben kann bzw. muß. Bestätigt man diesen Request mit "Ok", so ist die Datenbank scheinbar geschrumpft. Sie würden sicherlich bald herausfinden, daß nun nur noch die Records angezeigt werden, die den Suchparametern entsprechen. Die Funktionen **Records/Nächster**, **Records/Voriger** oder deren Äquivalente verhalten sich nun die "Weitersuchen" bzw. "Rückwärts suchen".

Der Filter wird gleich nach dem Bestätigen des Requesters erstellt. Das bedeutet, daß neu hinzugefügte Records auch angezeigt werden, auch wenn sie nicht mit dem Filter übereinstimmen. Um dies zu erreichen muß man einfach den Requester noch mal aufrufen und bestätigen.

Über den Menüpunkt **Vergleichen/Filter An?** läßt sich der Filter zeitweise aktivieren und deaktivieren.

1.19 Alle Suchmuster

	Muster	Unterstützte Feldtypen	Funktion
--kein Muster--	Alle Typen		Exakte Übereinstimmung
#?	String , Extern , Datatypes		Eine unbekannte Zeichenkette undefinierter Länge. Kann nur einmal in einem Text vorkommen.
?	String , Extern , Datatypes		Ein unbekanntes Zeichen.
> x	Integer , Slider		Eine Zahl, die größer als x ist.
< x	Integer ,		

```

Slider
    Eine Zahl, die kleiner als x ist.

>= x
Integer
,
Slider
    Eine Zahl, die größer oder gleich x ist.

<= x
Integer
,
Slider
    Eine Zahl, die kleiner oder gleich x ist.

!=
Integer
,
Slider
    Eine Zahl, die ungleich x ist.

```

Genauere Beschreibungen zu den Mustern finden sich bei den Dokumentationen der einzelnen

Feldtypen
 und bei der allgemeinen Beschreibung der
 Suchfunktionen

1.20 Relationen

Gleich vorweg: Ich habe noch nie mit einer anderen Datenbank gearbeitet, die Relationen unterstützt. So weiß ich nicht genau, ob meine Implementation genau mit der ursprünglichen Idee übereinstimmt. Hauptsache ist jedoch: Sie funktioniert!

Relationen sind Felder, deren Inhalt nicht in der Datei des Projektes, aus dem die Relationen stammen, sondern in einer weiteren Projekt-Datei zu finden sind. Um zu wissen, aus welchem Record die Daten entnommen werden sollen, müssen beide Projekte ein Feld mit einem eindeutigen Identifikations-Inhalt haben, das als Schlüssel (Key) bezeichnet wird.

Um nun in Fiasco Relationen nutzen zu können, muß man zuerst in ein Projekt erzeugen, das die Quelle für ein anderes Projekt darstellt, das dann daraus die Daten liest. Dieses Projekt wird später "dort" genannt. In diesem Quell-Projekt müssen dann mindestens zwei Felder erstellt werden, eins für die eigentlichen Daten, das andere für den Schlüssel. Das Feld für den Schlüssel sollte vom Typ Integer sein, da dies am schnellsten ist und von Fiasco automatisch mit Schlüsselns versorgt werden kann. Für besondere Fälle erlaubt Fiasco jedoch auch, jeden anderen Feldtypen zu verwenden.

Um die Option zu aktivieren, die dem Schlüssel-Feld automatisch einen eindeutigen Schlüssel zuweist, muß man im **Feld-Requester** das Gadget "Eindeutiger Schlüssel" aktivieren. Bitte beachten Sie, daß der Schlüssel nur beim Erstellen eines Records erzeugt wird! Falls schon Records existierten, bevor diese Option aktiviert wurde, behalten diese ihren alten, und möglicherweise nicht eindeutigen Inhalt. Außerdem kann man später den Inhalt dieses Feldes beliebig verändern, ohne daß irgendeine Überprüfung vorgenommen wird.

Der Typ des zweiten Feldes ist relativ egal. Man sollte sich nur bei den Datentypen String, Extern und Datatypes das Attribut "Maximale Zeichen" gut merken, da dieses im Feld des zweiten Projektes genau übereinstimmen muß.

Es wäre sinnvoll, wenn dieses Projekt bereits einige Records mit etwas Inhalt enthalten würde, damit nach dem Aktivieren der Relation sofort ein Effekt sichtbar wird.

Dieses Projekt muß jetzt abgespeichert werden.

Das zweite Projekt muß wiederum aus zwei Feldern bestehen, die in Typ und bei den Typen String, Extern und Datatypes im Attribut "Maximale Zeichen" übereinstimmen. Das Feld für den Schlüssel sollte jedoch nicht wieder "Eindeutiger Schlüssel" aktiviert haben, da dies keinen größeren Sinn machen würde.

Bevor nun die Relation aktiviert werden kann, sollte man das Projekt im selben Verzeichnis wie das erste Abspeichern, damit im Relations-Requester keine absoluten, sondern relative - oder in diesem Fall keine Pfade - benutzt werden können.

Um nun die Relation erstellen zu können, muß das Feld, das nicht den Schlüssel beinhalten soll, aktivieren und dazu den **Relations-Requester** aufrufen (mit **Felder/Relationen Ändern** . Hier sollte man zuerst den Schlüssel "hier" im Listview oben links auswählen. Danach muß die Relations-Datei mit dem Filerequester-Gadget in der Mitte ausgewählt werden. Der Requester sollte sich bereits im richtigen Verzeichnis befinden. Nachdem die Datei ausgewählt wurde, werden in den unteren beiden Listviews die möglichen Schlüssel und eigentlichen Felder "dort" angezeigt. Hier sollte man auch die richtigen Felder auswählen und anschließend "Ok" anwählen. Hat man alles richtig gemacht werden jetzt die Relationen zu den bereits bestehenden Records geladen.

Das wars!

Es gibt übrigens noch eine **Relationen-Checkliste** , die diese Schritte kurz zusammenfaßt.

1.21 Relations-Checkliste

- o Schlüsselfeld "dort" erstellen. Eventuell "Eindeutiger Schlüssel" aktivieren.
- o Eigentliches Feld "dort" erstellen. Bei String, Extern, Datatypes "Maximale Zeichen" merken!
- o Projekt speichern.

- o Schlüsselfeld "hier" erstellen. Typ muß mit Schlüsselfeld "dort" übereinstimmen.
- o Eigentliches Feld "hier" erstellen. Typ und "Maximale Zeichen" müssen übereinstimmen!
- o Projekt speichern.

- o Relations-Requester für Eigentliches Feld "hier" aufrufen.

- o Schlüssel "hier" auswählen

- o Relations-Datei auswählen

- o Schlüssel und eigentl. Feld dort auswählen. Falls nicht die richtigen Felder angezeigt werden, Typ und bei String, Extern od. Datatypes "Maximale Zeichen" überprüfen.

- o "Ok" anwählen

1.22 Feldtypen

Felder sind die eigentlichen Speicher für Daten. Dazu reichen \leftrightarrow eigentlich die Grundtypen "String" und "Zahl". Alle anderen Typen sind mehr oder weniger Modifikationen dieser Grundtypen, die die Arbeit mit der Datenbank erleichtern.

Fiasco unterstützt zur Zeit folgende Feldtypen:

String
Integer
Float
Boolean
Cycle
Slider
Date
Time
Extern
Datatypes
Text

1.23 Standard-Attribute

Diese Attribute werden normalerweise von einem Feldtyp unterstützt ↔
:

- Id
Hier kann man einen Identitäts-String angeben, mit dem man die Felder z.B. in ARexx-Scripts erkennen kann. Dieser String sollte nur einmal in einem Projekt vorkommen, überprüfungen daraufhin werden jedoch (noch) nicht ausgeführt. Sollte über eine nicht "einmalige" Id mit einem ARexx-Script versucht werden ein Feld anzusprechen, ist das Verhalten des Programms nicht definiert.
 - Breite (Width)
bestimmt die Breite in Zeichen, mit der das Feld in der Maske erscheint.
 - Startwert (Init Cont)
Hier kann man einen Wert angeben, der beim Neuerstellen eines Records in dieses Feld kopiert wird.
 - Alten Wert benutzen (Use old value)
Wenn man einen Record neu erstellt wird der Wert des alten Records übernommen. Wenn "Use old value" aktiv ist, wird Init Cont ignoriert. Dies ist insbesondere beim Eingeben vieler ähnlicher Datensätze nützlich.
 - Script
Hier läßt sich ein ARexx-Script angeben, das aktiviert wird, wenn ein neuer Record erstellt wird, bzw. der Inhalt des jeweiligen Feldes geändert wird. Abhängig vom Inhalt des Scripts können Init Cont und Use old value von ihrem eigentlichen Sinn abweichen. Auf das Script wird relativ von dem Verzeichnis zugegriffen, in dem das Projekt liegt.
-

Durch

Dehnungswerte
ungleich 0 können die Dimensions-Attribute verzerrt

werden.

1.24 Der String-Feldtyp

In einem String-Feld lassen sich Zeichenketten begrenzter Länge ↔
eingeben.

Neben den

Standard-Attributen
werden folgende Attribute unterstützt:

- Maximale Zeichen (Max Chars)
bestimmt die maximale Anzahl von Zeichen, die in das Feld eingegeben werden dürfen. Dieses Attribut wirkt sich direkt auf die Größe der Datei aus.

Suchäquivalent:

entspricht dem Feldinhalt.

Unterstützte Suchmuster:

- ? = Ein unbekanntes Zeichen.
- #? = Null oder mehr unbekanntes Zeichen.

1.25 Der Integer-Feldtyp

In einem Integer-Feld lassen sich Ganzzahlen im Bereich von ↔
-2.147.483.348 bis

2.147.483.347 eingeben.

Neben den

Standard-Attributen
werden folgende Attribute unterstützt:

- Maximale Zeichen (Max Chars)
bestimmt die maximale Anzahl von Zeichen, die in das Feld eingegeben werden dürfen.
- Eindeutiger Schlüssel (Gimme unique Key)
legt in diesem Feld bei Neuerstellung eines Records eine -für diese Datenbank-einmalige Identifikations-Nummer ab. Wenn dieses Attribut aktiv ist, werden "Init Cont" und "use old value" ignoriert.

Suchäquivalent:

entspricht dem Feldinhalt

Unterstützte Suchmuster:

- > - größer als
- < - kleiner als
- >= - größer oder gleich
- <= - kleiner oder gleich
- != - ungleich

1.26 Der Float-Feldtyp

In ein Float-Feld können reelle Zahlen eingegeben werden.

Neben den

Standard-Attributen
werden folgende Attribute unterstützt:

- Präzision (Precision)
Anzahl der darzustellenden Nachkommastellen.

Suchäquivalent:

entspricht dem Feldinhalt

Bemerkung: Die Genauigkeit von Float läßt noch zu wünschen übrig. Es empfiehlt sich, statt Float

String
zu benutzen. ARexx kann damit
genausogut rechnen, wenn korrekte Zahlen angegeben werden.

1.27 Der Boolean-Feldtyp

Ein Boolean-Feldtyp nimmt Wahrheitswerte auf, d.h. entweder Ja ↔
oder Nein.

Ein Boolean-Feld erscheint im Maskenmodus als ein "Checkbox-Gadget".

Von den

Standard-Attributen
werden folgende Attribute nicht unterstützt:

- Höhe (ist immer 1)
- Breite (ist immer 3)

Suchäquivalent:

TRUE oder 1 - Gesetzter Haken
FALSE oder 0 - Kein Haken

Unter Amiga OS 2.0 kann das Aussehen dieses Feldes leicht kaputt wirken, da

die Grafiken noch nicht skalierbar sind. Unter Amiga OS 3.0 oder höher werden die Grafiken korrekt an die Größe der Schrift angepaßt.

1.28 Der Cycle-Feldtyp

Der Cycle-Feldtyp bietet mehrere Auswahlmöglichkeiten aus einer selbst definierten Liste. Der Größte Vorteil beim Cycle-Feldtyp ist die Speichersparnis. Es werden maximal 65536 Auswahlmöglichkeiten unterstützt. (Ich hoffe, daß das reicht ;-)
Ein Cycle-Feld erscheint in der Maske als ein "Cycle-Gadget" (wie der Name schon sagt).

Neben den

Standard-Attributen
werden folgende Attribute unterstützt:

- Labels

Eine Liste aller Auswahlmöglichkeiten. Ein Eintrag ist Pflicht, ab zweien wird das Cycle-Feld sinnvoll.

"Init Cont" ist in dem Fall eines Cycle-Felds eine Nummer, die sich auf die Auswahlmöglichkeiten bezieht. 0 entspricht dabei dem ersten Eintrag, 1 dem zweiten, und so weiter.

Suchäquivalent:

entweder die Nummer des Eintrags a la "Init Cont",
oder der Eintrag im Klartext (nicht vertippen!)

1.29 Der Slider-Feldtyp

Ein Slider-Feld ist eine Abwandlung des Integer-Typs. Mit ihm können Ganzzahlen grafisch dargestellt und verändert werden. Der Bereich der Zahlen ist auf -32.768 bis 32.767 begrenzt, kann jedoch durch mehrere Attribute beeinflußt werden.

Neben den

Standard-Attributen
werden folgende Attribute unterstützt:

- Min. Wert (Min. Value)

gibt den niedrigsten Wert der dargestellt wird an. Dieser Wert entspricht der Stellung des "Knobs" am linken bzw. am oberen Rand des Feldes.

- Max. Wert (Max. Value)

gibt den höchsten Wert der dargestellt wird an. Dieser Wert entspricht der

Stellung des "Knobs" am rechten bzw. am unteren Rand des Feldes.

- Format

ist ein Formatstring im Stil der Programmiersprache C. Er baut sich wie folgt auf:

```
%[-][0][ZahlenFeld][.Maximum][l]Format
```

- -: Die Zahl wird im ZahlenFeld nach links ausgerichtet, normalerweise wird sie nach rechts ausgerichtet.
- 0: Das ZahlenFeld wird mit dem Zeichen '0' ausgefüllt. Bsp.: 1 -> 001
- ZahlenFeld: Die minimale Feldbreite
- Maximum: nur für Strings, spielt hier keine Rolle.
- l: Zeigt an, daß die Zahl 32-bit breit ist, muß hier immer angegeben werden.
- Format:
 - %b - BCPL-String, spielt hier keine Rolle.
 - %s - C-String, spielt hier keine Rolle.
 - %c - Char, das ASCII-Zeichen für die Zahl wird ausgegeben.
 - %d - Die Zahl wird ausgegeben.
 - %u - Die Zahl wird vorzeichenlos ausgegeben.
 - %x - Die Zahl wird im Hexadezimal-Format ausgegeben.

Die Formatierung erfolgt mit der Exec-Funktion RawDoFmt().

- MaxFormatLänge (MaxFormatLen)

gibt die maximale Länge der Format-Ausgabe an, dieser Bereich wird bei der Breite angerechnet, d.h. je größer MaxFormatLen, desto kleiner das eigentliche Feld.

Suchäquivalent:

Die eigentliche Zahl

Unterstützte Suchmuster:

- > - größer als
- < - kleiner als
- >= - größer oder gleich
- <= - kleiner oder gleich
- != - ungleich

1.30 Der Datums-Feldtyp

In ein Datums-Feld kann ein Datum eingegeben werden.

Neben den

Standard-Attributen

werden folgende Attribute unterstützt:

- Benutze aktuelles Datum (use current Date)

Beim Neuerstellen eines Records wird das aktuelle Datum in dieses Feld eingefügt.

Suchäquivalent:

entspricht dem Feldinhalt

Das Datum wird z.Zt. nur im deutschen Format (TT.MM.YYYY) gelesen und geschrieben. Es wird keine Überprüfung der Werte vorgenommen, so sind z.Zt. Werte wie 65.20.3687 möglich.

1.31 Der Time-Feldtyp

In ein Zeit-Feld kann eine Uhrzeit eingegeben werden.

Neben den

Standard-Attributen
werden folgende Attribute unterstützt:

- Benutze aktuelle Zeit (use current Time)
Beim neuerstellen eines Records wird die aktuelle Zeit in dieses Feld eingefügt.

Suchäquivalent:

entspricht dem Feldinhalt

Die Zeit wird z.Zt. nur mit Sekunden ausgegeben (HH:MM:SS). AM und PM wird nicht unterstützt. Außerdem wird keine Wertüberprüfung vorgenommen, so sind z.Zt. Werte wie 55:66:99 möglich.

1.32 Der Extern-Feldtyp

Ein Extern-Feld nimmt einen Dateinamen auf, der auf Wunsch an ein \leftrightarrow benutzerdefiniertes Programm als Argument weitergeleitet wird. So kann man weitere Daten zu einem Record definieren und abrufen.

Neben den

Standard-Attributen
werden folgende Attribute unterstützt:

- Kommando (Command)
ist der Name eines Programms, mit dem die Daten verarbeitet werden können. Die Zeichenfolge "%s" wird bei Aufruf durch den Inhalt des Feldes ersetzt. Läßt man "%s" weg, so wird kein Argument übergeben.
(z.B. "C:ED %s")
 - Stack
Gibt die Größe des Stacks für das Kommando an.
-

- Maximale Zeichen (Max Chars)
bestimmt die maximale Länge des Dateinamens. Dieses Attribut wirkt sich direkt auf die Dateigröße aus.

Suchäquivalent:

Entspricht dem Dateinamen; Inhaltsbezogene Suche ist nicht möglich

Die Programme werden mit dem AmigaDOS-Funktion System() aufgerufen. Für Aus- oder Eingabeoperationen wird ein Konsolen-Fenster geöffnet.

1.33 Der Datatypes-Feldtyp

Ein Datatypes-Feld ist Extern-Feldern ähnlich. Der Unterschied ist \leftrightarrow , daß die Datatypes-Library benutzt wird (und somit kann man dieses Feld erst ab OS 3.0 benutzen) und daß die Daten direkt in der Maske angezeigt werden. Durch die Datatypes-Library ist dieses Feld auch universell einsetzbar und frei erweiterbar.

Neben den

Standard-Attributen werden folgende Attribute unterstützt:

- Maximale Zeichen (Max Chars)
bestimmt die maximale Länge des Dateinamens. Dieses Attribut wirkt sich direkt auf die Dateigröße aus.
- Scroll-Leisten (Scrollbars)
Sollen Scroller zur Verfügung gestellt werden, mit denen der Inhalt des Feldes verschoben werden kann? Ohne Scroller kann immer nur die linke obere Ecke der Daten betrachtet werden. (Dies ist nicht ganz wahr. Manche Datentypen unterstützen das verschieben ihres Inhaltes, wenn man in den Bereich der Anzeige klickt und dabei in die Maus in die Richtung des verborgenen Inhalts zieht. Ein Beispiel ist der Bild Datentyp)

Suchäquivalent:

Entspricht dem Dateinamen; Inhaltsbezogene Suche ist nicht möglich

Der AmigaGuide- und der Animations-Datentyp scheinen mit relativ kleinen Feldern nicht zurechtzukommen. Weil die Daten bei jedem Wechsel des Records neu geladen werden müssen, kann sich dieser Vorgang verlangsamen.

1.34 Der Text-Feldtyp

Der Text-Feldtyp ist eigentlich gar kein echtes Feld, er dient \leftrightarrow lediglich zur Beschriftung der Maske.

Es werden folgende Attribute unterstützt:

- Text
ist der Text, der in die Maske geschrieben wird.
- Pen
gibt die Farbe an, in der der Text geschrieben werden soll.
Es sind möglich: Normal (normalerweise Schwarz), Highlight (normalerweise Weiß). Die Farben lassen sich über den Palette-Preferences-Editor beeinflussen.
- Bold
Ist dieses Attribut aktiv, wird der Text Fett geschrieben
- Italics
Läßt den Text kursiv schreiben
- Underlined
Läßt den Text unterstreichen.

Es werden keine
Standard-Attribute
unterstützt.

Suchäquivalent:

Nach einem Text-Feld kann aufgrund fehlenden Inhalts nicht gesucht werden.

1.35 Grafische Benutzer Oberfläche (GUI)

Das einzige, womit Fiasco sich normalerweise nach dem Start präsentiert, ist ein leeres Fenster. Durch Pull-Down-Menus kann man nun dem Programm seine Wünsche mitteilen. Für diejenigen, die Menus nicht so mögen, können über den Menüpunkt Kontrolle/ServiceFenster ein weiteres Fenster öffnen, mit dem sich die meisten Aktionen auch ohne Menu ausführen lassen. Die dritte Möglichkeit, mit Fiasco zu kommunizieren, sind Tastatur-Shortcuts.

Menus und Shortcuts

Service-Window

Requester

Mit der Maus direkt läßt sich im Masken-Modus der Cursor positionieren, und Felder verschieben. Mit einem Doppelklick auf ein Feld wird der

Feld-Requester

geöffnet, mit dem sich die Feld-Attribute verändern lassen.

Dies entspricht dem Menüpunkt
Feld/Ändern...

.

Fiasco unterstützt Menuhelp. Das bedeutet, daß wenn man über einem Menüitem die Help-Taste drückt, eine Beschreibung der Funktion in einem separatem AmigaGuide-Fenster angezeigt wird. Dazu muß man jedoch die amigaguide.library besitzen und den Guide unter dem Namen Fiasco.guide in dem Verzeichnis liegen haben, in dem Fiasco selbst liegt (Progdir:).

Die Requester, die Fiasco benutzt, haben alle eine grundlegende Struktur. Am unteren Rand befinden sich die Gadgets, mit denen man den Requester bestätigt. Meistens sind dies "Ok", was die Einstellungen im Requester ausführt und "Cancel", was die Einstellungen im Requester nicht gültig macht. "Ok" kann mit der Return-Taste abgekürzt werden und "Cancel" kann mit "Esc" abgekürzt werden. Das Closegadget oben links tut das selbe wie "Cancel" oder "Esc".

1.36 Service Window

Das Service Window läßt sich über den Menüpunkt
Kontrolle/ServiceFenster
öffnen bzw. schließen. Es sieht ungefähr so aus:

```

Add

Del

|<

<

>

>|

<Dateiname>

<Status>

<Feldtyp>

```

1.37 Neu (Add)

Falls das aktuelle Projekt sich im Record-Modus befindet, wird ↔ ein neuer Record erstellt. Ist der Modus der Masken-Modus, wird ein neues Feld des eingestellten Feld-Typs an der aktuellen Cursorposition erstellt.

Äquivalent zu dem Menüpunkt:

Records/Hinzufügen
bzw.

Felder/Feld hinzufügen...

1.38 Löschen (Delete)

Falls das aktuelle Projekt im Record-Modus ist, wird der \leftrightarrow
aktuelle Record
gelöscht. Ist der Masken-Modus aktiv, wird das aktive Feld gelöscht (falls keins
aktiv ist, wird auch keins gelöscht). Achtung: Das Löschen geschieht normaler-
weise ohne Sicherheitsabfrage!

Äquivalent zu dem Menüpunkt:

Records/Löschen
bzw.

Felder/Feld Entfernen

1.39 |<

Falls das aktuelle Projekt im Record-Modus ist, wird der \leftrightarrow
Record Nr.1
aktiviert.

Äquivalent zu dem Menüpunkt:

Records/Erster

1.40 <

Falls das aktuelle Projekt im Record-Modus ist, wird der dem \leftrightarrow
aktiven
Record vorausgehende Record aktiviert.

Äquivalent zu dem Menüpunkt:

Records/Voriger

1.41 >

Falls das aktuelle Projekt im Record-Modus ist, wird der dem \leftrightarrow
aktiven
Record folgende Record aktiviert.

Äquivalent zu dem Menüpunkt:

Records/Nächster

1.42 >|

Falls das aktuelle Projekt im Record-Modus ist, wird der letzte Record ←
aktiviert.

Äquivalent zu dem Menüpunkt:

Records/Letzter

1.43 Aktives Projekt

Hier wird der Name des aktuellen Projektes angezeigt.
Achtung: Sollten sich zwei Projekte nur im Pfad, nicht im Namen voneinander unterscheiden, wird beidesmal dasselbe angezeigt.

Ein anderes Projekt aktiviert man, indem man das jeweilige Fenster aktiviert.

1.44 Status

Hier werden Statusinformationen zum aktiven Projekt angezeigt.

Im Record-Modus ist dies:

<Nummer des aktiven Records>/<Anzahl der Records>

Die Zahlen können variieren, falls ein

Filter

aktiv ist.

Im Mask-Modus wird folgendes angezeigt:

X: <X-Position des Cursors>, Y: <Y-Position des Cursors>

1.45 Feldtyp

Hier stellt man ein, was für ein Feldtyp beim nächsten Erzeugen eines Feldes erzeugt wird. Äquivalent mit "Felder/Feldtyp".

1.46 Maske

Die Maske ist die Darstellungsform, mit der Fiasco am meisten arbeitet. In einer Maske kann im Gegensatz zur **Liste** nur ein Record gleichzeitig angezeigt werden. Dafür ist diese Anzeige übersichtlicher und geordneter (falls man es richtig gemacht hat ;-), weshalb diese Darstellungsform auch manchmal als "Formular" bezeichnet wird.

Die Maske setzt sich aus sogenannten Feldern zusammen, die verschiedene Typen und somit verschiedene Erscheinungsformen haben. Eine komplette Beschreibung aller Feldtypen finden Sie **hier** !

Bei normalen Amiga-Programmen würde man die "Felder" als Gadgets bezeichnen und de facto sind sie sogar welche. Fiasco benutzt für die Maske intensiv Gadgets der gadtools.library, weshalb man sich schnell an die Fiasco-Masken gewöhnen können sollte.

Fiasco-Masken passen sich automatisch an einen beliebigen Zeichensatz an, einzige Voraussetzung ist, daß er nicht proportional ist (z.B. topaz, courier).

Um eine Maske in Fiasco zu gestalten, muß man über den Menüpunkt **MaskenModus** in den Mask-Modus überwechseln. Nun kann man mit der Maus existierende Felder verschieben oder über das "Felder"-Menu verschiedene andere Manipulationen ausführen. Mehr dazu **hier** !

1.47 list

Über den Menüpunkt **Kontrolle/ListWindow** kann man ein Fenster mit einer Listendarstellung des aktuellen Projektes öffnen. In der Liste wird ein Record durch eine Zeile repräsentiert. Die Felder des Records sind durch Spalten gekennzeichnet. Die erste Zeile der Liste beinhaltet die Ids der jeweiligen Felder. Falls das Fenster nicht groß genug ist, um die gesamte Liste anzuzeigen, kann man die Scroll-Balken im rechten und unteren Fensterrand benutzen, um sich durch die Liste hindurchzuarbeiten. Die Zeile des aktuellen Records wird durch eine **Unterlegung** deutlich gemacht.

Sie können auch mit der Liste Records auswählen. Sie müssen lediglich den jeweiligen Record anklicken und schon wird er aktiviert. Änderungen im Record können Sie jedoch nur in der Maske vornehmen.

Falls ein **Filter** aktiv ist, werden nur noch die Einträge angezeigt, die mit dem Filter-Muster übereinstimmen.

Normalerweise werden für das Layout der Liste Position und Dimensionen der jeweiligen Masken-Felder für die Liste umgesetzt. Sie haben jedoch auch die Möglichkeit, hier einzugreifen. Die Breite der Spalten können Sie verändern, indem Sie in der Titelzeile, in der die Ids angezeigt werden, an der rechten Kante die linke Maustaste zu drücken. Nun erscheint eine Linie an der Mausposition, die Sie nun mit der Maus nach links oder rechts ziehen können. Dort, wo Sie sie fallen lassen, d.h. die linke Maustaste loslassen, befindet sich künftig der rechte Rand des Feldes. Falls Felder hiervon überschritten würden, werden Sie verschoben.

Die Position einer Spalten kann man auch verändern. Dazu klickt man über der Mitte des Spalten-Kopfes und hält die Maustaste gedrückt. Nun kann man die Spalte in der Liste an eine beliebige Stelle verschieben. Läßt man die Maustaste wieder los, so wird die Spalte bestmöglich zwischen den benachbarten Feldern plaziert.

Will man Spalten ganz verschwinden lassen, so macht man das, indem man mit der linken Maustaste den Kopf einer Spalte aktiviert. Nun kann man über den Menüpunkt **Liste/Spalte verbergen** diese Spalte aus der Liste verschwinden lassen. Will man sie wieder haben, muß man **Liste/Spalte anzeigen** auswählen. Daraufhin erscheint ein Requester mit allen Spalten, die über "Hide column" entfernt wurden. Wenn man die gewünschte ausgewählt hat, wird sie möglichst nahe an der alten Position wieder eingefügt.

Der Menüpunkt **Liste/Liste neu berechnen** bewirkt, daß der Aufbau der Liste völlig neu berechnet wird. Man kann ihn mit dem Menüpunkt "Inhalt aufräumen" der Workbench vergleichen. Verborgene Spalten werden jedoch nicht sichtbar gemacht.

1.48 Records

Records stellen in einer Datenbank - um den Kartei-Vergleich wieder mal heranzuziehen - die Karteikarten einer Kartei dar. Ein Record ist also eine Sammlung verschiedener Aspekte zu einem Oberbegriff; z.B. zu einer Person Name, Adresse usw. In der **Maske** wird immer nur ein Record auf einmal angezeigt, In der **Liste** werden Records als Zeilen dargestellt.

1.49 Unterlegung

Wenn Sie AmigaOS 3.0 oder höher benutzen, können Sie diese Farben mit dem Palette-Preferences-Editor manipulieren. Den "FILLPEN", der für die Unterlegung benutzt wird, können sie über den Eintrag "Titelbalken aktiver Fenster" (der Name ist leicht irreführend ;-)) ändern und den "FILLTEXTPEN", der für den darüber liegenden Text benutzt wird, über "Titeltexte aktiver Fenster".

Wenn Sie noch AmigaOS 2.x benutzen sollten, wird normalerweise als "FILLPEN" für die Unterlegung Farbe 3 (also Blau) und für "FILLTEXTPEN" Farbe 1 (also Schwarz) benutzt. Es gibt jedoch einige Publicscreen-Utilities in der PD (z.B. PSM), die es auch unter 2.x erlauben, die Pens auf eigenen Publicscreens zu manipulieren.

Der Text-Feldtyp nutzt übrigens noch einen zusätzlichen Pen, den "HIGHLIGHTTEXTPEN". Dieser wird benutzt wenn man im **Feld-Requester** "Hiligh" aktiviert.

1.50 Maskendehnung

Normalerweise sitzen die Felder in einer Fiasco-Maske sich dicht auf der Pelle. Dies ist a) nicht schön anzusehen, b) haben eigentliche alle anderen "normalen" GUIs ein paar Pixel zwischen den Gadgets Platz. Man könnte zwar jeweils eine Zeile zwischen den Feldern frei lassen, dies würde jedoch bald zu Platzproblemen führen. Deswegen gibt Fiasco die Möglichkeit, zwischen den einzelnen Zeilen und Spalten pixelweise Raum zu lassen.

Diese Werte können im **Options-Requester** in den Gadgets Stretch X / Y angegeben werden.

Ein Problem bei dieser Methode die Maske zu "dehnen" ist, daß auch die Felder größer werden als in den **Feld-Requestern** angegeben wurde. Dies fällt insbesondere in der horizontalen Richtung auf, da die meisten Fiasco-Felder sich nur in diese Richtung ausdehnen. Bei String-Feldern ist es so z.B. möglich, daß der Text, der in eingegeben werden kann nur bis z.T. das Feld füllen kann, da Max Chars geringer als der wahre Width-Wert ist. Das größte Problem ergibt sich bei Text-Feldern, die normalerweise nur so breit sind, wie der Text es erfordert. Nun sind sie aber breiter und der Text muß zentriert werden.

Um diese Probleme zu vermeiden, empfehle ich, daß als X-Wert Null angegeben wird und in dieser Richtung weiterhin einzelne Spalten als Trennung benutzt werden. In Y-Richtung liefert der 4 die besten Ergebnisse.

1.51 Record-Modus

In diesem Modus kann man Records hinzufügen, löschen und deren Inhalt verändern. Dieser Modus kann über den Menüpunkt **Kontrolle/Record-Modus** aktiviert werden. Das Gegenteil des Record-Modus ist der **Masken-Modus**.

1.52 Masken-Modus

In diesem Modus hat man die Möglichkeit, die Maske zu verändern, d.h. Felder neu zu erstellen, zu löschen und zu verschieben. Außerdem können hier Relationen erstellt und verändert werden. Dieser Modus kann über den Menüpunkt **Kontrolle/Masken-Modus** aktiviert werden. Das Gegenteil des Masken-Modus ist der **Record-Modus**.

1.53 Menus

Fiasco bietet diese Pull-Down-Menus:
(von links nach rechts; Menüpunkte, die mit einem '/' markiert sind, lassen sich aktivieren bzw. deaktivieren)

Deutscher Name	Engl.	Tastaturshortcut
----------------	-------	------------------

Projekt	(Project)	
	Neu	(New)
	A N	
	Leeren	(Erase)
	A Z	
	Öffnen...	(Open...)
	A O	
	Optionen...	(Options...)
	A \$	
	Statistik...	(Statistic...)
	Relationen neuladen	(Reload Rel)
	A !	
	Speichern	(Save)
	A S	
	Speichern als..	(Save as...)
	A A	
	Über Fiasco...	(About...)
	A ?	
		Quit
	A Q	
Records	(Records)	
	Hinzufügen	(Add Record)
	A +	
	Duplizieren	(Duplicate Record)
	A 2	
	Löschen	(Remove Record)
	A -	
	Voriger	(Previous)
	Cursor Up	
	Nächster	(Next)
	Cursor Down	
	Erster	(First Record)
	Ctrl Cursor Up	
	Letzter	(Last Record)
	Ctrl Cursor Down	
	Gehe zu...	(Goto...)
	A G	

Felder	(Fields)		
	Typ	(Field Type)	»
		String	Ctrl S
	Integer	Ctrl I	
	Float	Ctrl F	
	Boolean	Ctrl B	
	Cycle	Ctrl C	
	Slider	Ctrl S	
	Date	Ctrl A	
	Time	Ctrl M	
	Extern	Ctrl E	
	Datatypes	Ctrl D	
	Text	Ctrl T	
	Hinzufügen...	(Add Field...)	
	Enter		
	Ändern...	(Edit Field...)	
	Enter		
	Relationen Ändern.	(Edit Relatio)	
	A &		
	Entfernen	(Remove Field)	
	Del		
Liste	(List)		
	Spalte verber.	(Hide column)	
	A [
	Spalte sichtb.	(Show column...)	
	A]		
	Alle Spalten s.	(Show all colums)	
	Liste neu ber.	(Recalc List)	
	A %		
Vergleichen	(Compare)		
	Suchen...	(Find...)	
	A F		
	Weitersuchen	(Find next)	
	A >		
	rückwärts suc.	(Find previous)	
	A <		
	Ersetzen...	(Replace...)	
	A R		
	Zählen...	(Count...)	
	A #		

Sortieren... (Sort...)
A =

Filter Ändern.(Edit Filter...)
A ~

/ Filter An? (Use Filter?)
A `

Kontrolle (Control)

/ Record-Modus (Edit Records)
A D

/ Masken-Modus (Edit Mask)
A M

/ ServiceFenster (ServiceWindow)
A W

/ ListenFenster (ListWindow)
A L

/ ARexx-Debug (ARexx-Debug)
A B

Einstellungen (Settings)

/ Icons Erzeugen.(Create Icons?)

/ Backups Erzeugen.(Create Backups?)

/ Relativ. schreiben.(Write Rels?)

/ Relativ. aktualisieren?(Update Rels?)

/ GZZ-Fenster?(GZZ- Windows?)

/ Sicherheitsanforderungen.(Security-Reqs?)

/ Auto-Öffnen von SerWin.(AutoOpen SerWin)

/ SerWin dynamisch.(Dynamic SerWin?)

Schrift auswählen.(Choose Font...)

Einstellungen speichern.(Save Settings)

Einstellungen speichern als (Save Settings as)

Einstellungen laden (Load Settings...)
Benutzer (User)

Ändern... (Edit...)
A U

1.54 Projekt/Neu (Project/New)

Shortcut: A N

Erzeugt ein neues Projekt mit dazugehörigem Masken-Fenster. Es enthält dann weder Records noch Felder. Nun können sie entweder eine neue Datenbank erstellen, oder mit

Open...
eine bereits existierende

laden.

Querverweise:

Open...

1.55 Projekt/Leeren (Project/Erase)

Shortcut: A Z

Löscht jegliche Daten aus dem aktuellen Projekt, das Projekt wird hinterher in einem Zustand wie nach dem Programmstart oder

Projekt/Neu
sein.

Falls die Daten des Projektes seit dem letzten Abspeichern verändert wurden, wird ein vorher ein Sicherheitsrequester angezeigt.

1.56 Projekt/Öffnen (Project/Open...)

Shortcut: A O

Öffnet einen ASL-Requester, mit dem man ein Fiasco-Projekt zum Laden auswählen kann. Die Daten werden im aktiven Fenster angezeigt. Ein eventuell bereits in diesem Fenster geladenes Projekt wird aus dem Speicher entfernt. Um versehentlichen Datenverlust zu vermeiden, werden Sie gefragt, ob sie wirklich laden wollen, falls das Projekt seit dem letzten Speichern verändert wurde.

1.57 Projekt/Optionen... (Project/Options...)

Shortcut: A \$

Über diesen Menüpunkt können Sie den
Options-Requester
öffnen, in dem sie

projektspezifische Optionen einstellen können. Dies sind:

-
- Maskendehnung
- Name des Autors sowie Anmerkungen
- Dateiname des Projektes

Der letzte Punkt ermöglicht es, ohne

Save As...

aufrufen

zu müssen, den Namen des Projektes zu ändern.

1.58 Projekt/Statistik... (Project/Statistic...)

kein Shortcut

Gibt ein paar Informationen zum aktuellen Projekt aus. Beispiel:

Ein Record benötigt ca. 100 Byte RAM.

200 Records dieses Records belegen ca. 19 KByte RAM.

Der Speicher reicht für ca. 2300 weitere Records.

Der Speicherplatzbedarf für ein Projekt mit Feldern an sich wird nicht mitgerechnet.

1.59 Projekt/Relationen neuladen (Project/Reload Rels)

Shortcut: A !

Hier kann man Fiasco instruieren, alle Relationen im aktuellen Projekt neu zu laden, wie es bereits beim Laden des Projektes geschehen ist. Dies ist insbesondere für diejenigen nützlich, die den Menüpunkt

Settings/Update Relations?

deaktiviert haben,

einige Schlüsselworte neu eingegeben bzw. verändert haben und nun die Ergebnisse sehen wollen.

1.60 Projekt/Speichern (Project/Save)

Shortcut: A S

Mit Save werden die Daten des aktuellen Projektes unter dem selben Namen gespeichert, mit dem es auch geladen wurde. Falls Sie das Projekt unter einem anderen Namen speichern möchten, wählen sie

Speichern Als

, oder

Projekt/Optionen

um den Namen

zu ändern.

1.61 Projekt/Speichern als... (Project/Save As...)

Shortcut: A A

Hier können Sie das aktuelle Projekt unter einem anderen Namen als dem Namen, unter dem Sie es geladen haben, speichern. Der Name wird mit einem Requester der ASL-Library erfragt und nach dem Speichern beibehalten.

1.62 Projekt/Über Fiasco... (Project/About...)

Shortcut: A ?

Hiermit wird ein kleiner Requester angezeigt, der Auskunft über Version, Copyright und ein paar Systeminternia gibt.

1.63 Projekt/Beenden (Project/Quit)

Shortcut: A Q

Dieser Menüpunkt schließt das aktive Projekt. Falls es in der Zwischenzeit verändert und noch nicht gespeichert wurde, werden Sie über einen Requester gefragt, ob sie wirklich dieses Projekt schließen möchten. Falls dies das letzte Projekt war, das Fiasco offen hatte, wird Fiasco beendet.

1.64 Records/Hinzufügen (Records/Add Record)

Shortcut: A +

Fügt der Record-Liste des aktuellen Projektes einen neuen Record hinzu. Die einzelnen Felder enthalten nun entweder nichts, oder falls dies im

Masken-Modus

so eingestellt wurde, einen Startwert.

Falls das Listen-Fenster offen ist, erscheint dort auch ein entsprechender Eintrag.

Falls ein

Filter

aktiv ist, wird der Record automatisch für gültig erklärt, egal was in den einzelnen Feldern steht. Damit neuerstellte Records korrekt in den Filter miteinbezogen werden, sollte man

Vergleichen/Filter Ändern...

anwählen und einfach "Ok" anklicken.

Dieser Menüpunkt ist nur im
Record-Modus
anwählbar.

Querverweise:

Records/Löschen

1.65 Records/Duplizieren (Records/Duplicate Record)

Shortcut: A 2

Kopiert den aktuellen Record auf Haar genau in einen neuen Record. Jegliche Vorgaben wie "Init Cont", "Use current Time" o.ä. werden ignoriert. Selbst ein Feld, das das Attribut "Eindeutiger Schlüssel" gesetzt hat, enthält denselben Wert wie der alte Record, wodurch die Schlüssel nicht mehr eindeutig sind.

1.66 Records/Entfernen (Records/Remove Record)

Shortcut: A -

Löscht den aktiven Record und die darin befindlichen Daten. Falls sich noch

Relationen
in anderen Dateien auf diesen Record beziehen, sind diese ab ↔
sofort

nicht definiert.

Dieser Menüpunkt ist nur im
Record-Modus
anwählbar.

Falls der Menüpunkt
Settings/Security-Requester?
aktiv ist, erscheint

zuvor noch eine Sicherheitsabfrage, ob dieser Record wirklich gelöscht werden soll.

Querverweise:

Records/Hinzufügen

1.67 Records/Voriger (Records/Previous)

Shortcut: Cursor up

Aktiviert den dem aktuellen Record vorgehenden Record. Falls der aktuelle Record der erste in der Liste ist, wird die Anzeige "gebeebt".

Bitte beachten sie, daß beim Einsatz von
Filtern

der nächste vorhergehende
Record, der nicht ausgefiltert wird, aktiviert wird.

Der Tastatur-Shortcut wurde in Hinblick auf die Darstellung in der Liste
gewählt, in der der vorgehende Record über dem aktuellen liegt.

Dieser Menüpunkt ist nur im
Record-Modus
anwählbar.

Querverweise:

Next

First

Last

Goto

Find previous

1.68 Records/Nächster (Records/Next)

Shortcut: Cursor down

Aktiviert den Record nach dem aktuellen Record. Falls der aktuelle
Record der letzte in der Liste ist, wird die Anzeige "gebebt".

Bitte beachten sie, daß beim Einsatz von
Filtern
der nächste
Record, der nicht ausgefiltert wird, aktiviert wird.

Der Tastatur-Shortcut wurde in Hinblick auf die Darstellung in der Liste
gewählt, in der der nächste Record unter dem aktuellen liegt.

Dieser Menüpunkt ist nur im
Record-Modus
anwählbar.

Querverweise:

Previous

First

Last

Goto

Find next

1.69 Records/Erster (Records/First Record)

Shortcut: Ctrl Cursor up

Aktiviert den ersten Record des aktuellen Projektes.

Bitte beachten Sie, daß durch einen

Filter

dieses Verhalten verändert wird.

In so einem Fall wird der erste Eintrag aktiviert, der die Kriterien des Filters erfüllt.

Dieser Menüpunkt ist nur im
Record-Modus
anwählbar.

Querverweise:

Next

Previous

Last

Goto

1.70 Records/Letzter (Records/Last Record)

Shortcut: Control Cursor down

Aktiviert den letzten Record des aktuellen Projektes.

Bitte beachten Sie, daß durch einen

Filter

dieses Verhalten verändert wird.

In so einem Fall wird der letztmögliche Eintrag aktiviert, der die Kriterien des Filters erfüllt.

Dieser Menüpunkt ist nur im
Record-Modus
anwählbar.

Querverweise:

Next

Previous

First

Goto

1.71 Records/Gehe zu... (Records/Goto...)

Shortcut: A G

Öffnet den

Goto-Requester
mit dem man über die Nummer eines Records

diesen erreichen kann.

Bitte beachten Sie, daß die Nummern sich durch Einfügen oder Löschen von Records oder durch

Filter
die Nummern verändern können.

Dieser Menüpunkt ist nur im
Record-Modus
anwählbar.

Querverweise:

Next

Previous

First

Last

1.72 Feldtyp

Wählen Sie in diesem Untermenü den aktuellen Feldtypen aus. ↔

Dieser wird

beim Erstellen von Feldern benutzt. Das Gadget am unteren Rand des

Service Fensters
hat die selbe Funktion.

Diese Feldtypen sind verfügbar (mit Shortcut):

String
Ctrl S

Integer
Ctrl I

Float
Ctrl F

Boolean
Ctrl B

Cycle
Ctrl C

Slider

Ctrl S

Date
Ctrl A

Time
Ctrl M

Extern
Ctrl E

Datatypes
Ctrl D

Text
Ctrl T

1.73 Felder/Feld Hinzufügen... (Fields/Add Field...)

Shortcut: Return

Öffnet den

Feld-Requester

für den aktiven Feld-Typ und fügt das so
erstellte Feld an der Cursorposition ein.

Dieser Menüpunkt ist nur im
Masken-Modus
anwählbar.

Falls sich an der Cursorposition schon ein Feld befindet, läßt sich kein Feld
erzeugen.

Bitte beachten sie, daß Return auch Shortcut für
Feld Ändern
ist.

Enter hat die Funktion "Feld Hinzufügen", wenn kein Feld aktiv ist, ansonsten
wird "Feld Ändern" aufgerufen.

Querverweise:

Edit Field...

Edit Relations...

Remove Gadget

1.74 Felder/Feld Entfernen (Fields/Remove Field)

Shortcut: Del

Entfernt das aktive Feld aus der
Maske
und somit auch alle damit

verbundenen Daten. Falls sich noch irgendwelche Relationen auf dieses Feld verlassen, werden sich diese erst beim nächsten Zugriff beschweren.

Dieser Menüpunkt ist nur im
Masken-Modus
anwählbar.

Querverweise:

Edit Field...
Edit Relations...
Add Field

1.75 Felder/Feld Ändern... (Fields/Edit Field...)

Shortcut: Return

Öffnet den

Feld-Requester
für das aktive Feld, in dem man

diverse Daten für das Feld verändern kann. Würden bestimmte Veränderungen Datenverlust zur Folge haben, werden Sie zuvor darauf hingewiesen und haben die Möglichkeit den Vorgang abzubrechen.

Bitte beachten Sie, daß Felddtypen auf diesem Weg nicht verändert werden können.

Mit einem Doppelklick auf ein Feld läßt sich derselbe Zweck erzielen.

Bitte beachten sie, daß Return auch Shortcut für
Feld Hinzufügen
ist.

Enter hat die Funktion "Feld Hinzufügen", wenn kein Feld aktiv ist, ansonsten wird "Feld Ändern" aufgerufen.

Dieser Menüpunkt ist nur im
Masken-Modus
anwählbar.

Querverweise:

Add Field
Edit Relations

1.76 Felder/Relationen ändern... (Fields/Edit Relations...)

Shortcut: A &

Über diesen Punkt können sie die
Relationen
für das aktive Feld

im

Relations-Requester
einstellen. "Edit Relations" ist nur
anwählbar, wenn das Projekt im
Masken-Modus
ist.

1.77 Liste/Spalte verbergen (List/Hide column)

Shortcut: A [

Über diesen Menüpunkt kann man eine Spalte aus dem Listenfenster entfernen.
Es wird die aktuelle Spalte, die über den Listenkopf ausgewählt werden kann,
entfernt. Die Spalten rechts davon rücken danach auf.
Die Spalte kann über
Spalte sichtbar machen
wieder in
die Liste eingefügt werden.

Dieser Menüpunkt kann sowohl im Record- als auch im Masken-Modus aufgerufen
werden.

1.78 Liste/Spalte sichtbar machen... (List/Show column...)

Shortcut: A]

Dieser Menüpunkt ermöglicht es, die Spalten, die mit
Spalte verbergen
aus der Liste entfernt wurden, wieder in die Liste einzufügen. Es ↔
wird
versucht, die Spalte so nahe wie möglich an der alten Position wieder
einzufügen.

Dieser Menüpunkt kann sowohl im Record- als auch im Masken-Modus aufgerufen
werden.

1.79 Liste/Alle Spalten sichtbar (List/Show all columns)

kein Shortcut

Fügt alle Spalten, die über
Spalte verbergen
aus der Liste
entfernt wurden, in einem Schritt wieder ein.

Dieser Menüpunkt kann sowohl im Record- als auch im Masken-Modus aufgerufen
werden.

1.80 Liste/Liste neu berechnen (List/Recalc List)

Shortcut: A %

Dieser Menüpunkt berechnet alle Positionen und Dimensionen der Spalten in der Liste neu. Verborgene Spalten werden jedoch nicht wieder angezeigt.

Dieser Menüpunkt kann mit dem Menüpunkt "Inhalt aufräumen" ("Clean up") der Workbench verglichen werden.

1.81 Vergleichen/Suchen... (Compare/Find...)

Shortcut: A F

Öffnet den

Such-Requester
, in dem man Suchkriterien bestimmen kann.

Dieser Menüpunkt ist nur anwählbar, wenn sich das aktuelle Projekt im

Record-Modus
befindet und mindestens einen Record

enthält.

Querverweise:

Such-Requester

Find next

Find previous

1.82 Vergleichen/Weitersuchen (Compare/Find next)

Shortcut: A >

Aktiviert den nächsten Eintrag, der mit den Suchkriterien übereinstimmt, die mit dem

Such-Requester
spezifiziert worden sind. Falls kein Eintrag
mehr gefunden wird, erscheint ein Requester mit dem Text "Pattern not found!".

Dieser Menüpunkt ist nur anwählbar, wenn sich das aktuelle Projekt im

Record-Modus
befindet und mindestens einen Record

enthält.

Querverweise:

Such-Requester

Find...

Find previous

1.83 Vergleichen/rückwärts Suchen (Compare/Find previous)

Shortcut: A <

Aktiviert den vorherigen Eintrag, der mit den Suchkriterien übereinstimmt, die mit dem

Search-Requester

spezifiziert worden sind. Falls kein Eintrag

mehr gefunden wird, erscheint ein Requester mit dem Text "Pattern not found!".

Dieser Menüpunkt ist nur anwählbar, wenn sich das aktuelle Projekt im

Record-Modus

befindet und mindestens einen Record

enthält.

Querverweise:

Search-Requester

Find...

Find next

1.84 Vergleichen/Ersetzen... (Compare/Replace...)

Shortcut: A R

Öffnet den

Ersetzen-Requester

, mit dem Parameter für das

Ersetzen von Daten angegeben werden können.

Dieser Menüpunkt ist nur anwählbar, wenn sich das aktuelle Projekt im

Record-Modus

befindet und mindestens einen Record

enthält.

1.85 Vergleichen/Zählen... (Compare/Count...)

Shortcut: A #

Öffnet den

Count-Requester

,

mit dem Sie die Anzahl des Vorliegenden Musters in dem Projekt zählen können.

Dieser Menüpunkt ist nur anwählbar, wenn sich das aktuelle Projekt im

Record-Modus
befindet und mindestens einen Record

enthält.

Querverweise:

Find...

1.86 Vergleichen/Sortieren... (Compare/Sort...)

Shortcut: A =

Öffnet den

Sortier-Requester
, in dem Sie diverse Sortier-
kriterien angeben und den Sortiervorgang starten können.

Dieser Menüpunkt ist nur anwählbar, wenn sich das aktuelle Projekt im

Record-Modus
befindet und mindestens einen Record

enthält.

1.87 Vergleichen/Filter Ändern... (Compare/Edit Filter...)

Shortcut: A ~

Öffnet den

Filter-Requester
, mit dem man einen
Filter
erstellen

kann.

Dieser Menüpunkt ist nur anwählbar, wenn sich das aktuelle Projekt im

Record-Modus
befindet und mindestens einen Record

enthält.

1.88 Vergleichen/Filter An? (Compare/Use Filter?)

Shortcut: A ` (Graue Taste oben links)

Über diesen Menüpunkt kann man den aktuellen

Filter

ein und ausschalten.

Falls noch kein Filter erstellt worden ist, wird der Filter-Requester geöffnet.

Dieser Menüpunkt ist nur anwählbar, wenn sich das aktuelle Projekt im

Record-Modus befindet und mindestens einen Record

enthält.

1.89 Kontrolle/Record-Modus (Control/Record Mode)

Shortcut: A D

Hiermit können Sie das aktuelle Projekt in den Record-Modus versetzen,

in dem sie Records und deren Inhalt verändern können. Wenn dieser Modus aktiv ist, wird ein Haken vor den Menüpunkt gesetzt.

Das Gegenstück zu diesem Punkt ist Masken-Modus

.

Querverweise:

Record-Modus

Masken-Modus

1.90 Kontrolle/Masken-Modus (Control/Mask Mode)

Shortcut: A M

Hiermit können Sie das aktuelle Projekt in den Masken-Modus versetzen,

in dem sie die

Maske

verändern können. Wenn dieser Modus aktiv ist,

wird ein Haken vor den Menüpunkt gesetzt.

Das Gegenstück zu diesem Punkt ist Record-Modus

.

Querverweise:

Masken-Modus

1.91 ServiceFenster (ServiceWindow)

Shortcut: A W

Mit diesem Menüpunkt können sie das Service-Fenster öffnen und schließen. Das Service-Fenster erleichtert die wichtigsten Record- und Maskenoperationen und zeigt Statusmeldungen an. Das Fenster kann auch über das Close-Gadget geschlossen werden.

Das Service-Fenster ist global für alle Projekte zuständig.

1.92 ListenFenster (ListWindow)

Shortcut: A L

Mit diesem Menüpunkt können sie das Listen-Fenster öffnen und schließen. Das Fenster kann auch über das Close-Gadget geschlossen werden.

Jedes Projekt kann seine eigene Liste haben.

1.93 ARexx-Debug

Shortcut: A B

Hier kann man den speziellen Debug-Modus für die ARexx-Schnittstelle von Fiasco aktivieren. Wenn Fiasco-Kommandos Fehler erzeugen, werden zusätzlich Requester angezeigt, die das Problem näher bezeichnen.

1.94 Einstellungen/Icons Erzeugen? (Settings/Create Icons?)

Hier wird festgelegt, ob beim Speichern von Fiasco-Projekten Icons erzeugt werden, oder nicht

1.95 Einstellungen/Backups Erzeugen? (Settings/Create Backups?)

Hier wird festgelegt, ob beim Speichern von Fiasco-Projekten von den alten Dateien Backups unter <Dateiname>.bak abgelegt werden sollen.

1.96 Einstellungen/Relationen schreiben? (Settings/Write Relations?)

Hier kann man einstellen, ob beim Speichern von Projekten auch ↔
Relationen in
die entsprechenden Dateien, aus denen sie auch gelesen wurden, abgespeichert
werden. Ansonsten sind Änderungen, die in solchen Feldern gemacht wurden,
verloren. Dieser Menüpunkt sollte nur aktiv sein, wenn auch

Relationen aktualisieren?
aktiv ist, oder die Relationen vor
dem Speichern mit
Projekt/Relationen neuladen
aktualisiert
wurden, ansonsten könnten im Projekt, in dem die Relationen im Klartext
stehen, manche Einträge durch Leerstrings oder Falsche Einträge ersetzt werden.

1.97 Einstellungen/Relationen aktualisieren? (Settings/Update Rels?)

Hier kann man steuern, ob bei Eingabe eines neuen Schlüsselwortes ↔
in ein
Relations-Feld die Daten sofort erneuert werden sollen. Dazu sind
Diskettenzugriffe nötig, die für Menschen, die keine Festplatte haben, leicht
nervig werden können. Falls man diesen Menüpunkt deaktiviert, sollte man auch

Relationen schreiben?
deaktivieren, damit keine falschen Daten
in ein anderes Projekt geschrieben werden. Um die Änderungen so zu
betrachten, kann man die Relationen in einem Abwasch über

Projekt/Relationen neuladen
aktualisieren.

1.98 GZZ-Fenster? (GZZ Windows?)

Hier kann man kontrollieren, ob Maskenfenster im GZZ-Modus geöffnet werden
sollen. Dies verhindert, daß Intuition die Grafiken seiner Gadgets in die
Fensterränder abdriften läßt. Auf kleineren Maschinen kann jedoch die
Performance bald in die Knie gezwungen werden.

1.99 Einstellungen/Sicherheits-Requester? (Settings/Security-Reqs?)

Hier kann man einstellen, ob vor dem Löschen von Feldern oder Records noch
eine Bestätigung benötigt wird, um Datenverlust zu vermeiden.

1.100 Einstellungen/Auto-Öffnen ServiceWin? (Settings/Auto-Open ServiceWin?)

Hier kann man kontrollieren, ob beim Programmstart das Service-Fenster automatisch geöffnet werden soll.

1.101 Einstellungen/ServiceWin dynamisch? (Settings/Dynamic ServiceWin?)

Hier kann man kontrollieren, ob beim Öffnen des Service-Fensters eine freie Stelle gesucht werden soll, oder ob feste Koordinaten benutzt

werden sollen.

1.102 Einstellungen/Schrift auswählen... (Settings/Choose Font...)

Öffnet einen Font-Requester, mit dem man die Schrift auswählen kann, die für Masken- und Listendarstellung verwendet wird.

1.103 Einstellungen/Einstellungen speichern (Settings/Save Settings)

Speichert die Einstellungen unter "env:fiasco.prefs" und "envarc:fiasco.prefs". Sie sind somit auch nach einem Reboot aktiv.

1.104 Einstellungen/Einstellungen speichern als... (Settings/Save Settings as...)

Öffnet einen ASL-Requester, in dem man eine Datei auswählen kann, in der die Einstellungen gespeichert werden sollen.

1.105 Einstellungen/Einstellungen laden... (Settings/Load Settings...)

Öffnet einen ASL-Requester, in dem man eine Datei auswählen kann, aus der die

Einstellungen geladen werden sollen. Sie werden danach benutzt. Um sie dauerhaft benutzen zu können, sollte man sie mit Einstellungen speichern in envarc: und env: sichern.

1.106 Benutzer/Ändern... (User/Edit...)

Shortcut: A U

Öffnet den UserMenu-Requester, in dem man das Usermenu verändern kann.

1.107 Alle Requester

Edit Field
Felder

Find
Suchen

Replace
Ersetzen

Count
Zählen

Sort
Sortieren

Filter
Filtern

Usermenu
Definieren des User-Menus

Options
Projektspezifische Optionen

Goto
Gehe zu Record Nr. x

Edit Relation
Relationen

Show column
Eine Spalte ins Tageslicht holen

1.108 Der Feldrequester

Der Feld-Requester erlaubt das Manipulieren der Attribute ↔
jedes Feldes.

Da jeder Feldtyp andere Attribute hat, unterscheidet sich auch der Feld-Requester von Feldtyp zu Feldtyp. Die Gadgets, die im Feld-Requester

dargestellt werden entsprechen genau den Feld-Attributen, die bei den

Felddokumentationen
definiert sind.

Man wird mit dem Feldrequester konfrontiert, wenn man
Fields/Add Field
,

Fields/Edit Field
oder auf einem Feld doppelklickt.

Wenn man den Requester mit Ok bestätigt, werden alle Werte auf
Gültigkeit überprüft. Falls ein Wert nicht benutzt werden kann, wird man mit
einem Requester darauf aufmerksam gemacht.

Hier eine kleine Zusammenfassung aller Bedingungen:

- Falls man eine Id angeben kann, muß man sie angeben (und zwar eindeutig)
- Falls es MaxChars gibt, muß der Wert ≥ 1 sein.
- Width muß größer als zwei sein.

Wenn Dimensionswerte aufgrund weiterer Felder, die im Weg sind, nicht
benutzt werden können, erscheint ein anderer Requester, der einem die
Möglichkeiten "Shift", "Squeeze" und "Cancel" anbietet. Cancel tut gar
nichts, außer zum Feld-Requester zurückzukehren. "Squeeze" verkleinert das
Feld soweit, daß es paßt. "Shift" verschiebt das Feld so weit, daß es
genug Platz hat. "Shift" kann nicht immer angewandt werden.

Falls man ein bereits bestehendes Feld verändert, daß die Daten in
String-Form speichert und MaxChars unterstützt (also
String

,
Datatypes
und
Extern

) und MaxChars so weit verkleinert hat, daß bestehende
Einträge nicht mehr hinein passen würden, werden Sie gefragt, ob sie diese
wirklich auf die neue Länge verkleinern möchten.

1.109 Der Suchrequester

Feld (Field)

Hier können Sie das Feld wählen, in dem gesucht werden soll. In dem
Listview werden die Ids der Felder angezeigt, Textfelder werden nicht
angezeigt. Es kann immer nur ein Feld ausgewählt werden.

Muster (Pattern)

In dieses String-Gadget können sie ein Suchmuster eingeben. Entweder
ein einfacher Wert, oder ein Wert mit
Mustern

.

Bitte beachten Sie, daß dieser Wert später auch in den Count und
Replace-Requestern benutzt wird.

unscharf (blurred)

Falls sie eine
 unscharfe Suche
 durchführen wollen,
muß dieses Gadget aktiviert sein.

faktor (factor)
Hier können Sie die Toleranz der unscharfen Suchfunktion einstellen.
0 entspricht, daß die Funktion nur exakt übereinstimmende Einträge
akzeptiert, 100 entspricht fast allen Einträgen.

Nächstes (Next) (Return)
Bestätigt den Requester und sucht nach dem nächsten Eintrag, der mit
den oben angegebenen Bestimmungen übereinstimmt.

Erstes (First)
Bestätigt den Requester und sucht nach dem ersten Eintrag.

Voriges (Previous)
Bestätigt den Requester und sucht rückwärts nach dem nächsten Eintrag.

Abbrechen (Cancel) (Esc)
Schließt den Requester ohne weitere Aktion.

1.110 Der Ersetzen-Requester

Falls Sie sich bereits mit dem
Find-Requester
 auskennen,
sollten Sie sich hier sehr schnell zurechtfinden.

Feld (Field)
Hier können Sie das Feld wählen, in dem gesucht werden soll. In dem
Listview werden die Ids der Felder angezeigt, Textfelder werden nicht
angezeigt. Es kann immer nur ein Feld ausgewählt werden.

Muster (Pattern)
In dieses String-Gadget können sie ein Suchmuster eingeben. Entweder
ein einfacher Wert, oder ein Wert mit
 Mustern
 .
Bitte beachten Sie, da dieser Wert später auch im Suchrequester
benutzt wird, und umgekehrt.

Ersatz (Replacement)
Hier können Sie den Wert eingeben, der in die Einträge eingesetzt
werden soll, die vorher mit Pattern übereinstimmten.
Hier sind keine Muster möglich.

bestätigen (Confirm@){ub}
Wenn Sie wünschen, daß Sie jedesmal gefragt werden, ob ein Eintrag
wirklich ersetzt werden soll, sollten sie dieses Gadget aktivieren.

uunscharf (blurred)
Falls sie eine

unscharfe Suche
durchführen wollen,
muß dieses Gadget aktiviert sein.

Faktor (factor)

Hier können Sie die Toleranz der unscharfen Suchfunktion einstellen.
0 entspricht, daß die Funktion keinerlei Einträge akzeptiert, 100
entspricht fast allen Einträgen.

1.111 Der Zähl-Requester

Dieser Requester erlaubt es Ihnen, die Records zu zählen, die \leftrightarrow
einem
bestimmten Suchkriterium entsprechen. Mehr zum Zählen im
Suchabschnitt

Der Requester ist über den Menüpunkt
Vergleichen/Zählen
erreichbar.

Falls Sie sich bereits mit dem
Find-Requester
auskennen,
sollten Sie sich hier sehr schnell zurechtfinden.

Feld (Field)

Hier können Sie das Feld wählen, in dem gesucht werden soll. In dem
Listview werden die Ids der Felder angezeigt, Textfelder werden nicht
angezeigt. Es kann immer nur ein Feld ausgewählt werden.

Muster (Pattern)

In dieses String-Gadget können sie ein Suchmuster eingeben. Entweder
ein einfacher Wert, oder ein Wert mit
Mustern

Bitte beachten Sie, daß dieser Wert auch später im Suchrequester
benutzt wird und umgekehrt.

unscharf (blurred)

Falls sie eine
unscharfe Suche
durchführen wollen,
muß dieses Gadget aktiviert sein.

Faktor (factor)

Hier können Sie die Toleranz der unscharfen Suchfunktion einstellen.
0 entspricht, daß die Funktion keinerlei Einträge akzeptiert, 100
entspricht fast allen Einträgen.

Ok (Return)

bestätigt den Requester und zählt, wie oft das Suchmuster gefunden
wird. Dies wird zum Schluß angezeigt.

Abbrechen (Cancel) (Escape)
schließt den Requester ohne jede weitere Aktion.

1.112 Der Sortier-Requester

Im Sortier-Requester kann man ein Fiasco-Projekt alphabetisch ↔
bzw.
numerisch sortieren. Er ist über den Menüpunkt
Compare/Sort...
erreichbar.

Sortieren nach (Sort by)
Hier wird die Liste aller Felder angezeigt. Aus diesen sollte man ein
Feld auswählen, nach dem die Suche orientiert wird.

Ok (Return)
bestätigt den Requester und beginnt den Sortiervorgang. Der vorher
aktuelle Record wird beibehalten, die Nummer des Records wird sich
jedoch höchstwahrscheinlich ändern.

Abbrechen (Cancel) (Escape)
schließt den Requester ohne irgendetwas anderes noch zu tun.

1.113 Der Filter-Requester

Filter bieten die Möglichkeit, eine Übersicht über eine ↔
bestimmte Gruppe
von Records zu schaffen. Ein Filter erzeugt den Eindruck, als ob ein Projekt
nur aus Records bestände, die mit einem bestimmten Suchkriterium überein-
stimmen. Filter werden nicht während des normalen Programmablaufs ausge-
wertet, sondern immer nur auf einmal erstellt. Dies bedeutet, daß Records,
die während ein Filter aktiv ist hinzugefügt werden, auch angezeigt werden,
wenn das Suchkriterium nicht übereinstimmt. Das selbe gilt für veränderte
Records.

Der Filter-Requester ist über den Menüpunkt
Vergleichen/Filter Ändern
erreichbar.

Falls Sie sich bereits mit dem
Find-Requester
auskennen,
sollten Sie sich hier sehr schnell zurechtfinden.

Feld (Field)
Hier können Sie das Feld wählen, in dem gesucht werden soll. In dem
Listview werden die Ids der Felder angezeigt, Textfelder werden nicht
angezeigt. Es kann immer nur ein Feld ausgewählt werden.

Muster (Pattern)

In dieses String-Gadget können sie ein Suchmuster eingeben. Entweder ein einfacher Wert, oder ein Wert mit Mustern

unscharf (blurred)
Falls der Filter unscharf sein soll, muß dieses Gadget aktiviert sein.

faktor (factor)
Hier können Sie die Toleranz der unscharfen Suchfunktion einstellen. 0 entspricht, daß die Funktion keinerlei Einträge akzeptiert, 100 entspricht fast allen Einträgen.

Ok (Return)
bestätigt den Requester und erzeugt den Filter. Später scheint das Projekt nur noch aus Records zu bestehen, die den Bedingungen entsprechen.

Abbrechen (Cancel) (Escape)
schließt den Requester ohne jede weitere Aktion.

1.114 Der Usermenu-Requester

Fiasco bietet die Möglichkeit, eigene Menüpunkte zu definieren ← und mit speziellen Funktionen zu belegen. Dabei ist die Wahl zwischen Name und Kommando, das das Item ausführen soll, frei wählbar. Die in diesem Requester definierten Menüpunkte sind neben dem "normalen" Weg mit der Maus über die F-Tasten erreichbar. F1 bis F10 entsprechen den ersten zehn Menüpunkten. Shift+F1 bis F10 entsprechen dann den Menüpunkten 11 bis 20. Falls jemand mehr als 20 Menüpunkte definieren sollte, muß sich dann damit abfinden, die verbleibenden Menüpunkte mit der Maus aufrufen zu müssen.

Weiterhin lassen sich nicht mehr als 63 Menüpunkte definieren. Tut man es doch, wird ein Requester erscheinen, der darauf hinweist, daß die Anzahl der Menüpunkte auf 63 zurückgeschraubt wurde.

Die Menüpunkte lassen sich über Settings/Save Settings dauerhaft sichern.

Items
Hier wird eine Liste aller bereits bestehenden Menu-Items dargestellt. Mit dem "+"-Button kann man weitere hinzufügen, mit dem "-"-Button das aktuelle entfernen und mit den "<"/>"-Buttons kann man die Position des aktuellen Items in der Liste verändern.

Typ (Type)
Hier kann man einstellen, ob das aktuelle Item ein Programm oder ein ARexx-Script ausführen soll.

Kommando (Command)

Hier kann man das Programm bzw. ARexx-Script eingeben, das ausgeführt werden soll.

1.115 Der Options-Requester

Der Options-Requester dient dazu, einige Einstellungen \leftrightarrow
speziell
für das aktuelle Projekt zu machen. Er ist über den Menüpunkt

Project/Options...
oder dem ARexx-Befehl

F_OptionsReq
erreichbar.

Name

Über dieses Gadget kann man den Datei-Namen des Projektes verändern. Unter diesem Namen wird das Projekt in Zukunft gespeichert werden, alle Verzeichnis-Relativen Operationen werden ab dann dieses Verzeichnis als aktuelles Verzeichnis haben.

Ator (Athor)

Hier kann sich der Ersteller des Projektes verewigen. Der Name wird sich dann am Anfang der Projekt-Datei finden. Ansonsten wird der Inhalt ignoriert.

Bemerkungen (Annotations)

Noch ein Gadget zur freien Benutzung. Hier kann man irgendwelche Notizen, z.B. einen Versions-String (mit "\$VER:" am Anfang) eintragen. Dieser String wird noch vor dem Autor abgelegt. Und wiederum wird der Inhalt sonst ignoriert.

Dehnung (stretch) X Y

Hier kann man einen Wert angeben, der zur Breite bzw. der Höhe des Cursors hinzuaddiert wird. Dies bewirkt eine Dehnung der Make in X-Richtung bzw. Y-Richtung. Mehr zu diesen Dehnungs-Werten
hier

1.116 Goto-Requester

Der Goto-Requester ist einer der übersichtlichsten Requester \leftrightarrow
in
Fiasco überhaupt. Er ist über den Menüpunkt
Records/Goto...
erreichbar und bietet die Möglichkeit, über die Record-Nummer \leftrightarrow
diesen Record
zu aktivieren.

Bitte beachten Sie, daß
Filter
die Record Nummern
beeinflussen.

Gehe zu (go to)
Nimmt die Nummer des Records auf.

Ok (Return)
Bestätigt den Requester und springt zum Record mit der Nummer.

Abbrechen (Cancel) (Esc)
Ist mit gerade entfallen... %-)

1.117 Der Relations-Requester

Dies ist die Schaltzentrale für die
Relationen
in Fiasco.

Dieser Requester ist über
Fields/Edit Relations...
erreichbar.

Schlüssel hier
In diesem Listview muß man den Schlüssel in dem aktuellen Projekt
auswählen.

eigentl. Feld hier
Hier wird die Id des Feldes, dessen Relationen gerade bearbeitet werden,
angezeigt.

Relations-Datei
Hier muß man die Projekt-Datei relativ zum Verzeichnis des aktuellen
Projektes angeben, in der die passenden Informationen stehen.

Schlüssel dort
Hier muß man das Feld in dem Projekt, das unter "Relations-Datei"
angegeben wurde, auswählen, das den eindeutigen Schlüssel beinhaltet.
Da die Typen der beiden Schlüssel übereinstimmen müssen, werden hier nur
die Felder angezeigt, die zu dem Schlüssel unter "Schlüssel hier" passen.

eigentl. Feld dort
Hier muß man das Feld in dem Projekt, das unter "Relations-Datei"
angegeben wurde, auswählen, das den Partner zu "eigentl. Feld hier"
darstellt. Aus diesem Feld werden dann die Daten, die in "eigentl. Feld
hier" eingetragen werden, bezogen. Auch hier werden nur die Felder
angezeigt, die zu "eigentl. Feld hier" passen.

Ok (Return)
Bestätigt den Requester und versucht die Relationen zu laden. Falls dabei
ein Fehler auftritt, wird der Requester mit einer entsprechenden Mitteilung
wieder aktiviert, ansonsten wird das Haupt-Fenster aktiviert.

Cancel (Esc)

Schließt den Requester ohne weitere Aktion.

1.118 Spalte anzeigen-Requester

Mit diesem Requester, der über
List/Show column...
erreichbar
ist, kann man Listen-Spalten, die vorher mit
List/Hide coulumn
verborgen
wurden, wieder sichtbar machen. Die Spalten werden dann ihrem alten Platz ←
möglichst
nahe in die Liste eingefügt.

Feld (Field)

Hier werden alle verborgenen Spalten angezeigt, aus denen man eine
auswählen sollte.

Ok (Return)

Bestätigt den Requester und berechnet die Liste neu, so daß die Spalte
wieder zu sehen ist.

Abbrechen (Cancel) (Esc)

Schließ den Requester ohne noch was zu tun.

1.119 arexx

ARexx ist eine Macro-Sprache mit der Fähigkeit, Programme ↔
untereinander
kommunizieren zu lassen. ARexx wurde von William S. Hawes entwickelt und
befindet sich im Lieferumfang von Amiga OS 2.0 oder höher.

Der ARexx-Port von Fiasco kann entweder aus einem Script unabhängig vom
Programm angesprochen werden, oder Fiasco ruft selber Scripts auf. Dies
geschieht Beispielsweise, wenn bei einem Feld ein

Script

angegeben wurde und

der Inhalt dieses Feldes geändert wurde. Dieses Script hat dann die Möglich-
keit auf die Änderung zu reagieren und bestimmte Werte anzupassen.

Damit ein ARexx-Script mit Fiasco in Verbindung treten kann, muß ARexx
am Anfang instruiert werden, nach Fiascos ARexx-Port zu suchen. Dies
geschieht mit "Address Fiasco"

Fast alle Operationen, die über die GUI von Fiasco ausgeführt werden
können, können auch mit ARexx-Scripts erledigt werden. Außerdem kann Fiasco
mit ARexx fast beliebig an Funktionen erweitert werden. Viele ARexx Kommandos
tun genau das selbe wie die Menüpunkte. Das heißt, daß sie unter Umständen
einen Requester öffnen. Es ist aber meistens möglich, dies zu umgehen. Dies

wird später einmal behoben werden. Weiterhin gibt es Befehle, die immer einen Requester öffnen. Dies mag zwar für Scripts auch manchmal nützlich sein, ist jedoch primär implementiert wurden, um Fiasco eine Art zweites Menu wie Iconleisten zu geben. Ich habe bereits mit ToolManager-Docks experimentiert, doch für schnelles Umherblättern in einer Datenbank reagiert das ganze noch zu träge.

Allgemeines zu ARexx mit Fiasco

Index aller ARexx Kommandos

Index aller GUI-Äquivalenten Kommandos

Index aller GUI-Unabhängigen Kommandos

1.120 Allgemeines zu ARexx mit Fiasco

Ein Fiasco-Kommando gibt normalerweise in der Variable RC 0 ← zurück, wenn alles

glatt gelaufen ist. Falls ein Kommando nicht die Gegebenheiten, die es benötigt, vorfindet, wird RC auf 5 gesetzt. Fehler wie falsche Parameter werden mit RC = 10 belohnt, fatalere Fehler kriegen sogar RC = 20. Parameterfehler werden übrigens nur bei zu wenigen Parametern erzeugt, überzählige werden ignoriert. Parameter werden normalerweise durch Leerzeichen begrenzt, es sei denn, sie sind in Anführungszeichen (" , nicht ') eingeschlossen. Bei bestimmten Kommandos wird die gesamte Eingabezeile als ein Parameter behandelt. Dies wird in den einzelnen Kommando-Dokumentationen festgelegt. Wenn ein Kommando einen Wert zurückliefert steht dieser normalerweise in der Variablen Result. Um RESULT zu benutzen, muß man die Zeile "OPTIONS RESULTS" am Anfang eines ARexx scripts einfügen.

Die Fehlersuche in ARexx-Scripts ist etwas problematisch. Scripts, die durch das User-Menu oder durch Felder aktiviert wurde, können keine Ausgaben machen, so werden jegliche Fehlermeldungen verschluckt. Zum Testen von ARexx-Scripts empfiehlt es sich daher, sie vorher von der Shell aus zu testen (mit rx <Scriptname>; Fiasco muß natürlich dabei mit den passenden Daten bereitstehen). Um mehr über die Gründe zu erfahren, warum ein an Fiasco gerichtetes Kommando fehlerhaft arbeitete, kann man im Kontrolle-Menu den Menüpunkt "ARexx-Debug" aktivieren. Dies bewirkt, daß jedesmal, wenn Fiasco einen Fehlercode zurückgeben würde, ein Requester erscheint, in dem das Problem beschrieben wird. Hier hat man auch die Möglichkeit, dem Script vorzugaukeln, das Kommando wäre erfolgreich (Ignore Error-Gadget). Will man, daß der Fehlercode normal weitgegeben wird, sollte man "Continue" anwählen.

Die Dokumentation zu den Kommandos ist im Stil an die Amiga OS-Autodocs angelehnt. Synopsis legt eine Aufrufschablone fest, die anderen Punkte sind eigentlich selbsterklärend.

Index aller ARexx Kommandos

Index aller GUI-Äquivalenten Kommandos

Was für ein externes Menu verwendet werden kann.

Index aller GUI-Unabhängigen Kommandos
Was keine Requester erzeugt

1.121 Index der ARexx Kommandos

F_AboutReq
F_AddFieldReq
F_AddRecord
F_ClearProject
F_CloseServiceWin
F_CloseList
F_CountRecs
F_CountReq
F_DupRec
F_FilterReq
F_FindFirst
F_FindNext
F_FindPrev
F_FindReq
F_GetFieldCont
F_GetProjName
F_GetProjFullName
F_GetRecNum
F_GotoFirstRec
F_GotoLastRec
F_GotoNextRec
F_GotoPrevRec
F_GotoRec
F_GotoRecReq

F_IsVirgin
F_Locate
F_MakeVirgin
F_NewProject
F_OpenServiceWin
F_OpenList
F_OpenProject
F_OpenProjectReq
F_OptionsReq
F_Quit
F_RemRecord
F_SaveProject
F_SaveProjectReq
F_SaveSettings
F_SetDefFontReq
F_SetFieldCont
F_SetMode
F_SetSearchPat
F_SetSearchField
F_SortReq
F_SelectProj
F_UserCommand

1.122 Index der GUI-Abhängigen ARexx-Kommandos

F_AboutReq
F_AddFieldReq
F_AddRecord
F_ClearProject

F_CloseServiceWin
F_CloseList
F_CountReq
F_DupRec
F_FilterReq
F_FindNext
 *
F_FindPrev
 *
F_FindReq
F_GotoFirstRec
F_GotoLastRec
F_GotoNextRec
F_GotoPrevRec
F_GotoRec
F_GotoRecReq
F_NewProject
F_OpenServiceWin
F_OpenList
F_OpenProjectReq
F_OptionsReq
F_Quit
F_RemRecord
F_SaveProject
F_SaveProjectReq
F_SaveSettings
F_SetDefFontReq
F_SetMode
 *
F_SortReq

```
F_UserCommand
*
```

* -- Diese Kommandos entsprechen nicht genau den GUI-Funktionen.
Mit ein paar Programmzeilen mehr können diese jedoch nachgebildet
werden.

1.123 Index der GUI-Unabhängigen ARexx-Kommandos

```
F_AddRecord
F_ClearProject
*
F_CloseServiceWin
F_CloseList
F_CountRecs
F_DupRec
F_FindFirst
F_FindNext
F_FindPrev
F_GetFieldCont
F_GetProjName
F_GetProjFullName
F_GetRecNum
F_GotoFirstRec
F_GotoLastRec
F_GotoNextRec
F_GotoPrevRec
F_GotoRec
F_IsVirgin
F_Locate
F_MakeVirgin
F_NewProject
```

```
F_OpenServiceWin
F_OpenList
F_OpenProject
F_Quit
*
F_RemRecord
*
F_SaveProject
F_SaveSettings
F_SetFieldCont
F_SetMode
F_SetSearchPat
F_SetSearchField
F_SelectProj
F_UserCommand
*
```

* -- Diese Kommandos können unter Umständen einen Requester öffnen.
Bitte schauen Sie in die jeweilige Dokumentation, wie man dies
verhindern kann.

1.124 f_openservicewin

```
NAME
F_OpenServiceWin -- Öffne das Service-Fenster

SYNOPSIS
F_OpenServiceWin

FUNKTION
Öffnet das
Service-Fenster
, falls es noch nicht offen ist.

ARGUMENTE
keine

RESULTATE
keine

SIEHE AUCH
```

F_CloseServiceWin

1.125 f_closeservicewin

NAME

F_CloseServiceWin -- schlieÙe das Service-Fenster

SYNOPSIS

F_CloseServiceWin

FUNKTION

Schließt das Service-Window. Falls es zum Zeitpunkt des Aufrufes nicht offen ist, tut F_CloseServiceWin gar nichts.

ARGUMENTE

keine

RESULTATE

keine

SIEHE AUCH

F_OpenServiceWin

1.126 f_newproject

NAME

F_NewProject -- Öffne ein neues Projekt-Fenster

SYNOPSIS

F_NewProject

FUNKTION

Öffnet ein neues Projekt, d.h. ein neues Fenster und aktiviert dieses. Anschließend können neue Felder erstellt werden oder eine Datenbank geladen werden.

Macht exakt dasselbe wie

Project/New...

ARGUMENTE

keine

RESULTATE

keine

BUGS

Sollte Fehlerfall signalisieren.

SIEHE AUCH

1.127 f_openprojectreq

NAME
F_OpenProjectReq -- Öffne den OpenProject-Requester

SYNOPSIS
F_OpenProjectReq

FUNKTION
Öffnet einen File-Requester, mit dem der Benutzer eine Fiasco-Datenbank zum Laden auswählen kann. Falls das alte Projekt zwischenzeitlich geändert wurde, wird ein Sicherheitsrequester erzeugt, der fragt, ob man das alte Projekt wirklich aus dem Speicher werfen will. Entspricht dem Menüpunkt
Projekt/Öffnen...

ARGUMENTE
keine

RESULTATE
keine

SIEHE AUCH
F_OpenProject
Projekt/Öffnen

1.128 f_openproject

NAME
F_OpenProject -- Lade ein Projekt

SYNOPSIS
F_OpenProject <Name>
rc = Erfolg

FUNKTION
Versucht ein Fiasco-Projekt von Diskette ins aktuelle Fenster zu laden. Die aktuellen Daten werden ohne Sicherheitsabfrage gelöscht.

ARGUMENTE
Name - Dateiname des Projektes

RESULTATE
rc = 0, wenn alles glatt ging
= 10, wenn Argument fehlt oder Datei nicht geladen werden kann

SIEHE AUCH
F_OpenProjectReq

1.129 f_saveproject

NAME
F_SaveProject -- Speichert das aktuelle Projekt

SYNOPSIS
F_SaveProject

FUNKTION
Speichert das aktuelle Projekt unter dem aktuellen Namen auf Disk.
Entspricht dem Menüpunkt
Projekt/Speichern...

ARGUMENTE
keine

RESULTATE
keine

BUGS
Informiert das Programm nicht über Fehlerzustände

SIEHE AUCH
F_SaveProjectReq

1.130 f_saveprojectreq

NAME
F_SaveProjectReq -- öffne Requester und speichere unter diesem Namen

SYNOPSIS
F_SaveProjectReq

FUNKTION
Öffnet einen FileRequester, mit dem der Benutzer einen Namen angeben kann, unter dem das aktuelle Projekt anschließend gespeichert wird.
Entspricht dem Menüpunkt
Projekt/Speichern Als...

ARGUMENTE
keine

RESULTATE
keine

BEMERKUNGEN
Da der Benutzer im Requester auch "Abbrechen" anwählen kann, ist es möglich, daß das Projekt nicht gespeichert wird.

SIEHE AUCH
F_SaveProject

1.131 f_aboutreq

NAME
F_AboutReq -- öffnet den About-Requester

SYNOPSIS
F_AboutReq

FUNKTION
Öffnet einen Intuition-EasyRequester, der einige Informationen über das Programm anzeigt.
Entspricht dem Menüpunkt
Projekt/Über...

ARGUMENTE
keine

RESULTATE
keine

SIEHE AUCH

1.132 f_optionsreq

NAME
F_OptionsReq -- Öffne den Options-Requester für das Projekt

SYNOPSIS
F_OptionsReq

FUNKTION
Öffnet den
Options-Requester
, mit dem der Benutzer bestimmte
Projektspezifische Einstellungen bestimmen kann.
Entspricht dem Menüpunkt
Projekt/Optionen...

ARGUMENTE
keine

RESULTATE
keine

SIEHE AUCH

1.133 f_setdeffontreq

NAME
 F_SetDefFontReq -- Öffne den Fontrequester

SYNOPSIS
 F_SetDefFontReq

FUNKTION
 Öffnet den Fontrequester, mit dem der Benutzer die Standard-Schrift für die Maske auswählen kann.
 Macht genau das selbe wie "Einstellungen/Font auswählen..."

ARGUMENTE

RESULTATE

SIEHE AUCH

1.134 f_findfirst

NAME
 F_FindFirst -- Suche nach dem eingestellten Muster

SYNOPSIS
 F_FindFirst
 Result = Number_of_Record

FUNKTION
 Sucht vorwärts nach der ersten Übereinstimmung mit dem mit
 F_SetSearchPat
 eingestellten Suchmuster. Wenn rc = 0 ist, enthält result ↔
 die Nummer
 des gefundenen Records. Diesen kann man dann mit
 F_GotoRec
 dann
 aktivieren. Falls keine Übereinstimmung gefunden wurde, oder noch kein
 Suchmuster angegeben wurde, ist rc = 5.

ARGUMENTE
 keine

RESULTATE
 Wenn rc = 0, result = Nummer der Übereinstimmung.
 Wenn rc = 5, nichts mehr gefunden.

BEISPIEL
 /* Find-Example.rexx */
 options results
 address Fiasco

count = 0

F_SetSearchField "TestFeld" /* irgendein Feldname
 * (Sollte hier String sein) */
 F_SetSearchPat "?#?" /* Suche nach allen Records,

```

                                * die mindestens ein Zeichen
                                * beinhalten */
F_FindFirst                      /* Suche nach dem ersten Record */

do while rc = 0                  /* Suche solange, bis nichts mehr
                                * gefunden wird */
    F_GotoRec Result            /* Aktiviere den gefundenen Record */

    count = count + 1

    F_FindNext                  /* Weitersuchen */

end

/* Alle Records durchlaufen */

SIEHE AUCH

```

1.135 f_findnext

```

                                NAME
F_FindNext -- Suche nach dem eingestellten Muster

SYNOPSIS
F_FindNext
Result = Number_of_next_Record

FUNKTION
Sucht vorwärts nach der nächsten Übereinstimmung mit dem mit

    F_SetSearchPat
    eingestellten Suchmuster. Wenn rc = 0 ist,
enthält result die Nummer des gefundenen Records. Diesen kann man dann mit

    F_GotoRec
    dann aktivieren. Falls keine Übereinstimmung
gefunden wurde, oder noch kein Suchmuster angegeben wurde, ist rc = 5.

BEMERKUNG
Der aktuelle Record wird bei der Suche mit F_FindNext und F_FindPrev nicht
mit einbezogen. Will man also ein Programm schreiben, in dem das ganze ←
Projekt
durchsucht wird, muß man zuerst
    F_FindFirst
    und dann
F_FindNext benutzen.

ARGUMENTE
keine

RESULTATE
Wenn rc = 0, result = Recordnummer der Übereinstimmung.
Wenn rc = 5, nichts mehr gefunden.

```

BEISPIEL
siehe

F_FindFirst
SIEHE AUCH

1.136 f_findprev

NAME

F_FindPrev -- Suche nach dem eingestellten Muster rückwärts

SYNOPSIS

F_FindPrev
Result = Number_of_prev_Record

FUNKTION

Sucht rückwärts nach der nächsten Übereinstimmung mit dem mit

F_SetSearchPat
eingestellten Suchmuster. Wenn rc = 0 ist,
enthält result die Nummer des gefundenen Records. Diesen kann man dann
mit

F_GotoRec
dann aktivieren. Falls keine Übereinstimmung gefunden
wurde, oder noch kein Suchmuster angegeben wurde, ist rc = 5.

ARGUMENTE

keine

RESULTATE

Wenn rc = 0, result = Recordnummer der Übereinstimmung.
Wenn rc = 5, nichts mehr gefunden.

BEMERKUNG

Innerhalb von ARexx-Script ist F_FindPrev relativ unpraktisch.
Bessere Ergebnisse lassen sich mit Kombinationen von
F_FindFirst
und
F_FindNext
erzielen.

SIEHE AUCH

1.137 f_findreq

NAME

F_FindReq -- Öffne den Such-Requester

SYNOPSIS

F_FindReq

FUNKTION
Öffnet den
Such-Requester
, entspricht exakt dem Menüpunkt

Vergleichen/Suchen...
Dieses Kommando kann nur im
Record-Modus
aufgerufen werden.

ARGUMENTE
keine

RESULTATE
keine

SIEHE AUCH

1.138 f_setsearchpat

NAME
F_SetSearchPat -- Setze das Suchmuster

SYNOPSIS
F_SetSearchPat <Suchmuster>
rc = Success

FUNKTION
Setzt das Suchmuster für das aktive Projekt. Anschließend kann,
falls mit
F_SetSearchField
auch ein Suchfeld eingestellt wurde,
mit
F_FindFirst
und
F_FindNext
nach dem Muster gesucht werden.

ARGUMENTE
Suchmuster - Ein String, nach dem gesucht werden soll.

RESULTATE
0, wenn alles glattgegangen ist.
10, falls das Argument fehlt.
20 bei einem internen Systemfehler (wie Speichermangel, etc.)

SIEHE AUCH

1.139 f_setsearchfield

NAME
F_SetSearchField -- Setze das Suchfeld

SYNOPSIS
F_SetSearchField <FeldId>
rc = Success

FUNKTION
Setzt das aktuelle Bezugsfeld für die Suche auf das Feld, das als Argument angegeben wurde.

ARGUMENTE
FeldId - Die Id des Feldes, in dem gesucht werden soll.

RESULTATE
Gibt 10 zurück wenn das Argument fehlt, oder die Id unbekannt ist, ansonsten 0.

SIEHE AUCH
F_SetSearchPat

1.140 f_addfieldreq

NAME
F_AddFieldReq -- Öffne den Add-Field-Requester

SYNOPSIS
F_AddFieldReq

FUNKTION
Öffnet den
Feld-Requester
für den eingestellten Feldtypen.
Dieses Kommando macht exakt das selbe wie
Felder/Hinzufügen...
Es kann nur im
Masken-Modus
aufgerufen werden.

ARGUMENTE

RESULTATE

SIEHE AUCH

1.141 f_gotoprevrec

NAME

F_GotoPrevRec -- aktiviere den vorherigen Record.

SYNOPSIS

F_GotoPrevRec

FUNKTION

Aktiviert den dem aktuellen Record vorhergehenden Record. Falls der aktuelle der erste Record ist oder keine Records existieren, passiert nichts.

Äquivalent mit dem Menüpunkt
Records/Voriger

Kann nur im Record-Modus aufgerufen werden.

ARGUMENTE

keine

RESULTATE

keine

SIEHE AUCH

1.142 f_gotonextrec

NAME

F_GotoNextRec -- aktivieren den nächsten Record

SYNOPSIS

F_GotoNextRec

FUNKTION

Aktiviert den Record, der dem aktuellen folgt. Falls der aktuelle der Letzte ist oder keine Records existieren, passiert nichts.

Äquivalent mit dem Menüpunkt
Records/Nächster

Kann nur im Record-Modus aufgerufen werden.

ARGUMENTE

keine

RESULTATE

keine

SIEHE AUCH

1.143 f_gotofirstrec

NAME
F_GotoFirstRec -- aktiviere den ersten Record

SYNOPSIS
F_GotoFirstRec

FUNKTION
Aktiviert den ersten Record. Falls keine Records existieren, passiert nichts.

Äquivalent mit dem Menüpunkt
Records/Erster
.

Kann nur im Record-Modus aufgerufen werden.

ARGUMENTE
keine

RESULTATE
keine

SIEHE AUCH

1.144 f_gotolastrec

NAME
F_GotoLastRec -- aktiviere den letzten Record

SYNOPSIS
F_GotoLastRec

FUNKTION
Aktiviert den letzten Record im aktuellen Projekt

Äquivalent mit dem Menüpunkt
Records/Letzter
.

Kann nur im Record-Modus aufgerufen werden.

ARGUMENTE

RESULTATE

SIEHE AUCH

1.145 f_quit

NAME
F_Quit -- schließe das aktuelle Projekt

SYNOPSIS

F_Quit

FUNKTION

Schließt das aktuelle Projekt. Dieser Aufruf entspricht dem Menüpunkt

Project/Beenden

, wenn das Projekt also nach dem letzten

Speichern nochmal verändert wurde, wird ein Requester geöffnet, der fragt, ob das Projekt wirklich geschlossen werden soll.

Die Funktion muß also nicht unbedingt ihren Zweck erfüllen.

Um dies zu verhindern, kann man mit dem Kommando

F_MakeVirgin

dem Programm vorgaukeln, daß das Projekt nicht verändert ←
wurde.

ARGUMENTE

keine

RESULTATE

keine

BEMERKUNG

Wenn das aktuelle Projekt geschlossen wird, wird ein anderes aktiviert oder, falls sonst keins mehr offen war, das Programm beendet. Ein ARexx-Script sollte sich nicht auf die Reihenfolge verlassen, in der die Projekte aktiviert werden.

SIEHE AUCH

1.146 f_gotorec

NAME

F_GotoRec -- aktiviere einen bestimmten Record.

SYNOPSIS

F_GotoRec <RecordNummer>

FUNKTION

Aktiviere den Record, dessen Nummer als Argument übergeben wurde. Falls die Nummer falsch war, tue gar nichts.

ARGUMENTE

RecordNummer - Die Nummer des Records. Bitte beachten Sie, daß sich durch Sortiervorgänge und Hinzufügen oder Entfernen von Records und Filtern die Nummern der Records verändern können.

RESULTATE

SIEHE AUCH

1.147 f_gotorecreq

NAME
 F_GotoRecReq -- Öffne den "Goto-Requester"

SYNOPSIS
 F_GotoRecReq

FUNKTION
 Dieses Kommando fordert den Benutzer auf, die Nummer eines Records anzugeben, der daraufhin aktiviert wird.

Äquivalent mit dem Menüpunkt
 Records/Gehe zu...

.

Kann nur im Record-Modus aufgerufen werden.

ARGUMENTE
 keine

RESULTATE
 keine

SIEHE AUCH

F_GotoRec
 Menu
 Records/Gehe zu...

1.148 f_setfieldcont

NAME
 F_SetFieldCont -- Ändere den Inhalt eines bestimmten Feldes

SYNOPSIS
 F_SetFieldCont <FeldId> <Inhalt>
 rc = Success

FUNKTION
 Setzt den Inhalt des Feldes mit der angegebenen Id im aktuellen Record auf das Argument Inhalt.

Kann nur im Record-Modus aufgerufen werden.

ARGUMENTE
 FeldId - Identifikationsname des Feldes
 Inhalt - Neuer Inhalt für das Feld. Dieses Argument nimmt die gesamte restliche Kommandozeile ein, inclusive Leerzeichen. Anführungszeichen sind also nicht nötig. Die Interpretation dieses Arguments hängt von Feldtypen ab:
 String - wird direkt übernommen
 Integer - Zahlen werden direkt übernommen, anderes ergibt 0
 Float - entspricht Integer

Boolean - 1 oder TRUE = gesetzt, 0 oder FALSE = nicht gesetzt
Slider - Zahl wird übernommen, zu große bzw. zu kleine werden
angepaßt.
Cycle - Zahl oder Name des Eintrags wird übernommen

RESULTATE

rc = 0 - kein Fehler
rc = 5 - Kein Record aktiv, d.h. Datenbank ist leer
rc = 10 - Fehlendes Argument oder FeldId ist falsch

BUGS**SIEHE AUCH**

1.149 f_getfieldcont

NAME

F_GetFieldCont -- Lese den Inhalt eines bestimmten Feldes

SYNOPSIS

F_GetFieldCont <FeldId>
rc = Success
result = Inhalt

FUNKTION

Liest den Inhalt des Feldes mit der angegebenen Id im aktuellen Record aus und liefert ihn in result zurück.

Kann nur im Record-Modus aufgerufen werden.

ARGUMENTE

FeldId - Id des Feldes

RESULTATE

rc = 0 - alles Ok, result enthält den Inhalt
rc = 5 - kein Record aktiv, d.h. Projekt ist leer
rc = 10 - Argument fehlt oder Id ist nicht gültig

result - enthält den aktuellen Inhalt des Feldes, falls rc = 0 ist.

Das Format des Inhaltes:

String - der Text selbst.

Integer - die Zahl selbst.

Float - z.Zt. nicht unterstützt

Boolean - 1 = gesetzt, 0 = nicht gesetzt

Slider - den Wert des Sliders

Cycle - die Nummer des Eintrages.

SIEHE AUCH

1.150 f_addrecord

NAME
F_AddRecord -- Füge einen neuen Record hinzu.

SYNOPSIS
F_AddRecord

FUNKTION
Fügt einen neuen Record nach dem aktuellen ein. Dieser Record wird dann auch automatisch der aktuelle.
Diese Funktion kann nur im Record-Modus aufgerufen werden.

ARGUMENTE
keine

RESULTATE
keine

SIEHE AUCH
F_RemRecord
Records/Hinzufügen

1.151 f_remrecord

NAME
F_RemRecord -- Entferne den aktuellen Record

SYNOPSIS
F_RemRecord

FUNKTION
Entfernt den aktuellen Record und aktiviert den nächsten.
Diese Funktion kann nur im Record-Modus aufgerufen werden.

ARGUMENTE
keine

RESULTATE
keine

SIEHE AUCH
F_AddRecord
Records/Löschen

1.152 f_setmode

NAME

F_SetMode -- Wähle den Bearbeitungs-Modus

SYNOPSIS

```
F_SetMode <Modus>
rc = Success
```

FUNKTION

Aktiviert den angegebenen Modus für das aktuelle Projekt.

ARGUMENTE

Modus - "Mask" aktiviert den Masken-Modus
"Records" aktiviert den Record-Modus

RESULTATE

rc = 0 - kein Fehler
= 5 - das Projekt war bereits im gewünschten Modus
= 10 - Fehlendes oder falsches Argument

SIEHE AUCH

1.153 f_locate

NAME

F_Locate -- Locate the Cursor

SYNOPSIS

```
F_Locate <X> <Y>
```

FUNKTION

Setzt den Cursor an die angegebene Position. An dieser wird die nächste Masken-Operation stattfinden.
Diese Funktion kann nur im Masken-Modus aufgerufen werden.

ARGUMENTE

X - X-Coordinate
Y - Y-Coordinate

RESULTATE

BUGS

Ist z.Zt. nicht besonders nützlich, da noch keine direkten Kommandos zum Verändern der Maske zur Verfügung stehen.

SIEHE AUCH

1.154 f_countrecs

NAME

F_CountRecs -- zähle die Records

SYNOPSIS

```
F_CountRecs
```

```
result = <Anzahl_der_Records>
```

FUNKTION

Zählt die Records, aus denen das aktuelle Projekt besteht.
Kann nur im Record-Modus aufgerufen werden.

ARGUMENTE

keine

RESULTATE

Anzahl_der_Records - Die momentane Anzahl der Records im aktuellen Projekt, kann 0 sein. Bitte beachten Sie, daß Filter diesen Wert beeinflussen.

SIEHE AUCH

1.155 f_sortreq

NAME

```
F_SortReq -- Öffne den Sort-Requester
```

SYNOPSIS

```
F_SortReq
```

FUNKTION

Öffnet den Sort-Requester, in dem der Benutzer die Möglichkeit hat, das Projekt nach bestimmten Kriterien zu sortieren. Dieses Kommando macht genau das selbe wie der Menüpunkt "Compare/Sort...".
Es kann nur im Record-Modus aufgerufen werden.

ARGUMENTE**RESULTATE****SIEHE AUCH**

"Compare/Sort..."

1.156 f_countreq

NAME

```
F_CountReq -- Öffne den Count-Requester
```

SYNOPSIS

```
F_CountReq
```

FUNKTION

Öffnet den Count-Requester, in dem der Benutzer die Anzahl der Übereinstimmungen des aktuellen Projektes mit einem Suchmuster bestimmen kann. Dieses Kommando macht genau dasselbe wie "Compare/Count...".
Dieses Kommando kann nur im Record-Modus aufgerufen werden.

ARGUMENTE

RESULTATE

SIEHE AUCH

"Compare/Count..."

1.157 f_openlist

NAME

F_OpenList -- Öffne das Listen-Fenster

SYNOPSIS

F_OpenList

FUNKTION

Öffnet das Listen-Fenster, in dem der Benutzer einen Überblick über die Records des Projektes bekommt. Falls das Listen-Fenster bereits offen sein sollte, passiert gar nichts.

Dieses Kommando entspricht dem Aktivieren des Menüpunktes "Control/List"

ARGUMENTE

RESULTATE

SIEHE AUCH

"Control/List"

1.158 f_closetlist

NAME

F_CloseList -- schließe das List-Fenster

SYNOPSIS

F_CloseList

FUNKTION

Schließt das Listen-Fenster. Falls das Fenster nicht offen sein sollte, passiert gar nichts.

Diese Kommando entspricht dem Deaktivieren des Menüpunktes "Control/List"

ARGUMENTE

RESULTATE

SIEHE AUCH

"Control/List"

1.159 f_filterreq

NAME

F_FilterReq -- Öffne den Filter-Requester

SYNOPSIS

F_FilterReq

FUNKTION

Öffnet den Filter-Requester, in dem der Benutzer Parameter für einen Filter angeben kann.

Dieses Kommando entspricht dem Menüpunkt "Compare/Filter..."

Es kann nur im Record-Modus aufgerufen werden.

ARGUMENTE

RESULTATE

SIEHE AUCH

"Compare/Filter..."

1.160 f_duprec

NAME

F_DupRec -- Klone den aktuellen Record

SYNOPSIS

F_DupRec

FUNKTION

Dieses Kommando dupliziert exakt den aktuellen Record. Sämtliche Parameter wie unique Key, Init Cont, current Time usw. werden ignoriert. Diese Funktion tut exakt das selbe wie der Menüpunkt "Records/Dup Record". Falls kein Record aktiv sein sollte, passiert nichts. F_DupRec kann nur im Record-Modus aufgerufen werden.

ARGUMENTE

RESULTATE

SIEHE AUCH

"Records/DupRecord"

1.161 f_getrecrenum

NAME

F_GetRecNum -- Lese die Nummer des aktuellen Records aus.

SYNOPSIS

F_GetRecNum

Result = nummer_des_Records

FUNKTION

Gibt die Nummer des aktuellen Records in Result zurück.
Kann z.B. benutzt werden, um einen Anfangszustand zwischenspeichern
und am Schluß mit
 F_GotoRec
wiederherzustellen.

ARGUMENTE

keine

RESULTATE

Result = Nummer des Records. Bitte beachten Sie, daß Filter u.ä.
Record Nummern beeinflussen können.

SIEHE AUCH

1.162 f_savesettings

NAME

F_SaveSettings -- Sichere die aktuellen Programm-Einstellungen

SYNOPSIS

F_SaveSettings

FUNKTION

Sichert die aktuellen Programm-Einstellungen (in den Menus Settings
und User). Wenn Fiasco später neu gestartet wird, werden diese
Einstellungen sofort gültig.
Dieses Kommando entspricht dem Menüpunkt "Settings/Save Settings"

ARGUMENTE

keine

RESULTATE

keine

SIEHE AUCH

"Settings/Save Settings"

1.163 f_clearproject

NAME

F_ClearProject -- leere das aktive Project

SYNOPSIS

F_ClearProject

FUNKTION

Löscht alle Daten im aktuellen Projekt, so daß es sich hinterher
in einem Zustand wie gerade über "Project/New" neu geöffnete

Projekte befindet.

Dieses Kommando entspricht dem Menüpunkt "Project/Erase", so ist es also auch möglich, daß ein Requester erzeugt wird, in dem gefragt wird, ob das Projekt vorher noch gespeichert werden soll. Um dies zu umgehen, sollte man vorher mit

F_MakeVirgin

Fiasco sagen, daß das Changed-Flag zurückgesetzt wird, ←
nachdem

man eventuell mit

F_IsVirgin

die jungfräulichkeit des Projekts
überprüft hat.

ARGUMENTE

nichts

RESULTATE

nichts

SIEHE AUCH

"Project/Erase"

F_IsVirgin

F_MakeVirgin

1.164 f_getprojname

NAME

F_GetProjName -- lese den Dateinamen des aktuellen Projektes

SYNOPSIS

F_GetProjName
Result = <Dateiname>

FUNKTION

Gibt den Dateinamen ohne Pfad des aktuellen Projektes zurück.
Dieser Wert kann für
F_SelectProj
benutzt werden.

ARGUMENTE

keine

RESULTATE

Result - Name der Datei des aktuellen Projektes ohne Pfad
Es muß nicht unbedingt eine Datei mit diesem Namen existieren, z.B. wenn er über Options nachträglich geändert wurde.

SIEHE AUCH

F_GetProjFullName

1.165 f_getprojfullname

NAME
F_GetProjFullName -- Lese den Dateinamen des aktuellen Projekts

SYNOPSIS
F_GetProjFullName
Result = <Name>

FUNKTION
Gibt den Dateinamen des aktuellen Projektes incl. Pfad zurück.

BEMERKUNG
Der Pfad ist relativ zum aktuellen Verzeichnis von Fiasco.

ARGUMENTE

RESULTATE
Name - Name des Projektes incl. Pfad.

SIEHE AUCH
F_GetProjName

1.166 f_selectproj

NAME
F_SelectProj -- aktiviere ein bereits geladenes Projekt

SYNOPSIS
F_SelectProj <DateiName>
rc = <Success>

FUNKTION
Aktiviert ein Projekt, das sich bereits im Speicher befindet.
Zur Identifikation dient der Dateiname, ohne Pfad!!
Dies kann der Name, der durch
F_GetProjName
gewonnen wurde sein. Alle nachfolgenden Kommandos beziehen ↔
sich dann
auf das aktuelle Projekt

ARGUMENTE
DateiName - Name des Projektes, ohne Pfad!.

RESULTATE
Success = 5, falls das Projekt bereits aktiv ist
= 10, falls das Projekt nicht gefunden werden konnte, oder das
Argument fehlt.

SIEHE AUCH
F_GetProjName

1.167 f_usercommand

NAME

F_UserCommand -- Ruft ein Benutzer-Definiertes Kommando auf.

SYNOPSIS

F_UserCommand <Kommando-Nummer>

FUNKTION

Ruft ein Kommando auf, das durch den Benutzer im "User"-Menu definiert wurde.

ARGUMENTE

Kommando-Nummer - Nummer des Kommandos: 0 = erstes Menu-Item, 1 = zweites, usw. Falls eine Kommando-Nummer nicht existiert, wird sie ignoriert.

RESULTATE

keine

BEMERKUNG

Dieses Kommando sollte nur für das Implementieren einer Icon Leiste oder Ähnlichem benutzt werden. Benutzen sie es nicht in normalen Scripts, da Sie nicht wissen können, was hinter dem Kommando steckt.

SIEHE AUCH

1.168 f_isvirgin

NAME

F_IsVirgin -- Ist das Projekt unverändert?

SYNOPSIS

F_IsVirgin
rc = <IsVirgin>

FUNKTION

Testet, ob das aktuelle Projekt unverändert ist, d.h. ein Sicherheits-Requester bei Quit, Erase, Load o.ä. erscheinen würde.

ARGUMENTE

keine

RESULTATE

rc = 0 - Unverändert
rc = 5 - Verändert

SIEHE AUCH

F_MakeVirgin

1.169 f_makevirgin

NAME

F_MakeVirgin -- Gebe vor, daß ein Projekt unverändert ist.

SYNOPSIS

F_MakeVirgin

FUNKTION

"Macht" das aktuelle Projekt unverändert. Dies bewirkt, daß bei bestimmten Operationen (Erase, Load, Quit,...) kein Sicherheits-Requester geöffnet wird. Intern wird die Variable Changed auf FALSE gesetzt.

ARGUMENTE

keine

RESULTATE

Ein Projekt, das glaubt, daß es seit dem letzten Speichern nicht mehr verändert wurde

SIEHE AUCH

F_IsVirgin

1.170 f_

NAME

SYNOPSIS

FUNKTION

ARGUMENTE

RESULTATE

SIEHE AUCH

1.171 Beispiel-Projekte

Im Verzeichnis Databases liegen mehrere Fiasco-Projekte, die z ↵
.T. auch so

für eigene Zwecke weiterverwendet werden können. Hier folgen die Beschreibungen dieser Projekte:

Addresses

Adress-Verzeichnis

DatatypesDemo

Demonstration des Datatypes-Feldtyps (ab OS 3.0)

FamilyTree
Stammbaum

Videos
Videokassetten-Verwaltung

1.172 Adress-Verzeichnis

Das Adress-Projekt kann als Adressbuch benutzt werden, in dem Name, Adresse, Anrede usw. eingestellt werden kann. Das Feld "Country" wird über eine Relation mit der Datei "countrie.fdb" und dem Länderkürzel "SCountry" als Schlüssel kontrolliert. So braucht man nur noch das Länderkürzel einzugeben und schon wird das richtige Land herausgesucht.

Die Felder für Telefon, Fax und Postleitzahl sind String-Felder, da hier auch Sonderzeichen wie "/" oder eine "0" am Anfang beachtet werden müssen.

Eine weitere Idee wäre, den Namen der Stadt nach der Postleitzahl über eine Relation zu laden.

1.173 Datatypes Demo

Dieses Projekt ist eine einfache Demonstration des Datatypes-Feldtyps der die datatypes.library benötigt und so erst ab OS 3.0 verfügbar ← ist.

In der Maske gibt es drei Felder, in denen alle Daten angezeigt werden können, für die die entsprechenden Datatypes-Treiber vorhanden sind (AmigaGuide und Anim scheinen mit Fiasco manchmal Probleme zu haben). Zwei Felder haben Scrollbalken unten und rechts, damit man die angezeigte Datei frei bewegen kann. Im Stringgadget unter der Anzeige befindet sich der Name der Datei. Da die Daten in einer externen Datei liegen und bei jedem Record-Wechsel nachgeladen werden müssen, kann es dabei zu leichten Verzögerungen kommen.

1.174 Stammbaum

Der Stammbaum besteht aus den Projekten "persons.fdb" und "families.fdb". In "persons.fdb" müssen alle Personen, die im Stammbaum vorkommen sollen, aufgeführt werden. Hier können auch Geschlecht, Beruf, Geburtsdatum, usw. eingetragen werden.

Diese Daten werden dann von "families.fdb" über Relationen benutzt, um Namen von Ehepartnern, Kindern zu holen. Weiterhin sind hier noch Datums- und String-Felder für Heirat und Scheidung. Durch die intensive Nutzung von Relationen gibt es in diesem Projekt nur noch 10 "echte" Felder, die auf Disk abgelegt werden. Die restlichen 12 werden aus "persons" geladen.

1.175 Videos

In der Video-Datenbank können Homevideos komfortabel verwaltet werden. Die Datenbank besteht aus zwei Projekten: "filme.fdb" und "tapes.fdb". In "Filme" werden Daten zu einzelnen Filmen gespeichert, u.a. Genre, Regie, Darsteller und Jahr. Über das Feld "Kassette" wird jeder Film einer Kassette zugeordnet, die in "tapes" zu finden ist. Hier sind Marke und Länge der Kassette definiert. Über ein AREXX-Script wird dann die restliche freie Zeit jeder Kassette berechnet.

Dies funktioniert z.Zt. übrigens nur, wenn beide Projekte gleichzeitig geladen sind!

1.176 Anpassung an Sprachen

Ab Amiga OS 2.1 ist es dank der "locale.library" möglich, \leftrightarrow
Programme an
beliebige Sprachen anzupassen. Aufgrund meiner eher bescheidenen
Sprachkenntnisse gibt es für Fiasco zur Zeit nur Sprachen-Kataloge für
Deutsch und Englisch (eingebaut).

Falls jemand Fiasco an eine andere Sprache anpassen möchte, liegen bei Fiasco die Dateien fiasco.cd und fiasco.ct bei (in Development), die für das Erzeugen für Locale-Kataloge nötig sind. Ich gehe mal davon aus, daß die Übersetzer wissen, wie solche Dateien aufgebaut werden. Mehr Informationen zu Fiasco und den Catalogs stehen am Anfang von fiasco.cd.

Obwohl es kein Muß ist, würde ich mich freuen, wenn die Autoren von Locale-Katalogen
mir
diese und die Quelldateien zukommen
lassen würden. Ihre Kataloge werden dann der nächsten Distribution von Fiasco beiliegen.

Ebenso kann jeder der will die Anleitungen und die Beispiele an seine Sprache anpassen.

1.177 Wie man mich erreichen kann

Geschenke, Ideen, Bug-Reports usw. bitte an:

Nils Bandener
Dekanatsgasse 4
D-34369 Hofgeismar
Germany

email habe ich (noch) nicht.

1.178 bugs

Falls Sie irgendwelche Bugs oder Fehlfunktionen in Fiasco ←
entdecken,
wenden Sie sich bitte an
mich
. Bitte beschreiben Sie
den Fehler detailliert, d.h. geben Sie an, wie sich der Fehler aus-
drückt, und geben Sie eventuell im Hintergrund laufende Programme und
die Computer-Konfiguration an.

Von Fiasco sind z.Zt. folgende Fehler bekannt:

- Unter Kick 37.x flackert der Fensterrahmen des List-Fensters in
höchst merkwürdiger weise, wenn Menuoperationen ausgeführt wurden,
und das Fenster aktiv war.
- ARexx scheint Probleme mit Dateinamen zu haben, die Leerzeichen
enthalten. Der Name wird dann immer nur bis zum Leerzeichen inter-
pretiert. Da Fiasco vorher den Dateinamen zum vollen Pfad expandiert,
wirkt sich dies auch auf den vollen Pfad aus.
- Scheint manchmal (nicht immer!) etwas Speicher nicht freizugeben

1.179 todo

Natürlich ist an Fiasco noch lange nicht alles perfekt. Hier ←
ist eine
Liste aller Dinge, die ich noch irgendwann ändern oder hinzufügen will.
Wenn Sie auch Ideen haben, wie man Fiasco noch verbessern können, schreiben
sie
mir
!

- Schwächere Speicher-Fragmentation. Zur Zeit wird noch für jedes Feld
in jedem Record ein eigenes Stückchen Speicher angefordert. Die
Benutzung von Pools oder ähnlichem wird auch eine Beschleunigung des
Ladens und des Löschens von Projektes aus dem Speicher mit sich
bringen.
- Besseres Scrolling im Masken-Fenster. Durch die Benutzung von Gadgets
der GadTools-Library müssen alle Gadgets neu erstellt werden, wenn sie
bewegt werden müssen. Ich plane, die benutzen GadTools-Gadgets in Zu-
kunft zu emulieren, wobei der Benutzer möglichst wenig merken sollte.
(Tools wie CycleToMenu würden dabei jedoch nicht mehr funktionieren)
Außerdem könnte ich mir dieses sch**ß GZZ-Fenster sparen.
- Bessere Implementation von Relationen.
- Neue Suchfunktion, in der die Maske selbst zum Eingeben von Suchmustern
benutzt werden kann. Dies würde die Benutzung mehrerer Suchmuster
ermöglichen. Bin mir noch nicht sicher, ob ich dann die jetzige Such-
funktion beibehalten oder entfernen soll.

- Sortieren sollte sich auf mehrere Felder beziehen und sollte schneller werden.
- Neue ARexx-Kommandos: ReadRecord und WriteRecord. Sollten alle Feldinhalte eines Records durch ARexx-Variablen mit der Feld-Id als Inhalt lesen bzw. schreiben können.
- Neue Suchmuster. Weiß irgendjemand, ob die Dos-Patternmatching-Funktionen offiziell mehr als 30 Zeichen erlauben?
- "Virtuelle" Felder, deren Inhalt nicht gespeichert wird, sondern während der Laufzeit berechnet wird.
- Verbergen von Feldern in der Maske, wie es schon in der Liste möglich ist.
- Eine Abwandlung von Cycle-Feldern in Zusammenhang mit Relationen, wobei die einzelnen Labels eines Cycle-Gadget für die möglichen Werte stehen. Die Schlüssel würden dann unsichtbar werden.
- Reports
- Mehrzeilige Stringfelder
- Eine Option, die es erlaubt, nicht alle Records ins Ram laden zu müssen.
- Eine Möglichkeit, die es erlaubt, einen Feldtypen in einen anderen umzuwandeln
- Umgekehrte Logik beim Suchen/Zählen/Sortieren
- AppWindows für Datatypes und Extern-Felder
- "Packen" von Projekten, indem nach unbenutzten Feldern gesucht wird, und ← benutzte auf die benutzte Länge verkleinert werden.
- Überprüfen, ob ein ähnlicher Record bereits vorhanden ist (automatisch)
- Nach Extrema suchen (search -> höchster Eintrag, count -> meisten Einträge)

1.180 index

```

!=
Suchen

#?
Suchen

,
Suchen

<
Service-Fenster

```

<
Suchen

<=
Suchen

>
Service-Fenster

>
Suchen

>=
Suchen

>|
Service-Fenster

?
Suchen

|<
Service-Fenster

Add
Service-Fenster

Address Fiasco
ARexx und Fiasco

Ähnlichkeit
Unschärfe Suche

Aktives Projekt
Service-Fenster

ARexx

Aufräumen der Liste

Bold
Feld-Attribut (Text)

Boolean
Feldtyp

Breite von Spalten
Liste

Catalogs
Fiasco an Sprachen anpassen

Command
Feld-Attribut (Extern)

Confirm
Replace

Count
Requester

Count
Suchen

Cycle
Feldtyp

Datatypes
Feldtyp

Date
Feldtyp

Debugging
von ARexx-Scripts

Delete
Service-Fenster

Edit User Menu
Requester

Erstellen eines Feldes

Esc-Taste
GUI

Extern
Feldtyp

F-Tasten
User-Menu

FALSE
Suchen

Feld
Maske

Feldtyp
Service-Fenster

fiasco.cd
Fiasco an Sprachen anpassen

fiasco.ct
Fiasco an Sprachen anpassen

Filter
Requester

Filter
Suchen

Find
Requester

Float
Feldtyp

Format
Feld-Attribut (Slider)

gimme unique Key
Feld-Attribut (Integer)

GUI
Graphic User Interface

Height
Feld-Attribut

Highlight
Feld-Attribut (Text)

Hilfe
GUI

Id
Feld-Attribut

Init Cont
Feld-Attribut

Integer
Feldtyp

Italics
Feld-Attribut (Text)

Items
Feld-Attribut (Cycle)

Kommandos
die Fiasco über ARexx unterstützt

locale.library
Fiasco an Sprachen anpassen

Macros
ARexx und Fiasco

Maske

Maskendehnung
Projekt-Option

max chars
Feld-Attribut (String)

max chars

Feld-Attribut (Integer)

max chars
Feld-Attribut (Extern)

max chars
Feld-Attribut (Datatypes)

max format len
Feld-Attribut (Slider)

max value
Feld-Attribut (Slider)

min value
Feld-Attribut (Slider)

Name des Projektes
Ändern

Normal
Feld-Attribut (Text)

Online-Hilfe
GUI

Options
Requester

Parameter
ARexx und Fiasco

Pen
Feld-Attribut (Text)

Postion von Spalten
Liste

RawDoFmt ()
Level-Format

RC
ARexx und Fiasco

Recalc List
Liste

Relation

Replace
Suchen

Replace
Requester

RESULT
ARexx und Fiasco

Return-Taste
Requester

Script
Feld-Attribut

Scripts
ARexx und Fiasco

Scrollbars
Feld-Attribut (Datatypes)

ServiceWindow

Shift
Feld-Requester

Slider
Feldtyp

Sort
Requester

Spalten
im List-Fenster

Spaltenkopf
im List-Fenster

Sprachenanpassung
von Fiasco

Squeeze
Feld-Requester

Stack
Feld-Attribut (Extern)

Status
Service-Fenster

Stretching
Projekt-Option

String
Feldtyp

Suchen

Suchen mit ARexx

Suchmuster
Suchen

Text
Feldtyp

Text
Feld-Attribut (Text)

Time
Feldtyp

Toleranz
Unscharfe Suche

TRUE
Suchen

Underlined
Feld-Attribut (Text)

unscharf
Suchen

Unterlegung
GUI

use old value
Feld-Attribut

User
Menu

Verbergen von Spalten
Liste

Width
Feld-Attribut

Zählen
Suchen

Zeilen
im List-Fenster

Ziehen von Feldern
Datenbankerstellung
