

db

COLLABORATORS

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	db	1
1.1	db - Une petite et rapide base de données	1
1.2	disclaimer	2
1.3	preface	2
1.4	introduction	4
1.5	features	5
1.6	future	5
1.7	system	6
1.8	installing	6
1.9	settings	6
1.10	ref	8
1.11	window	9
1.12	view	9
1.13	gadgets	10
1.14	keys	11
1.15	menus	11
1.16	arexx	17
1.17	installarexxmenu	19
1.18	arexxcommands	19
1.19	rff	22
1.20	identifiers	24
1.21	speed	28
1.22	history	28

Chapter 1

db

1.1 db - Une petite et rapide base de données

Bienvenue dans
db 2.10
Le programme de base de données

Table des matières
=====

Généralité

Manuel de référence

Préface

La fenêtre

Introduction

Les Vues

Fonctions

Les gadgets

Le futur

Les touches

Système requis

Les menus

Installation et mise en marche

ARexx dans db

Préférences

Liste des commandes ARexx
Informations techniques

Le format de fichier RFF

Les identificateurs RFF

Une note sur SpeedRender

Historique des changements

1.2 disclaimer

Généralité

DB a été testé et trouvé stable pour une utilisation quotidienne. Cependant, l'auteur n'est pas responsable pour une perte des données, dommages aux programmes ou matériel qui peuvent résulter directement ou indirectement de l'utilisation de ce programme.

1.3 preface

Préface

Ce programme est postcardware, c'est à dire freeware, ce qui signifie que vous pouvez le copier gratuitement tant que vous ne demandez pas plus d'argent pour lui que le montant occasionné pour la copie. Si vous aimez ce programme, envoyez une carte postale à l'auteur (adresse ci-dessous). Vous avez le droit d'utiliser DB pour un usage privé, en groupes et dans des entreprises commerciales.

Pour ceux qui ont déjà envoyé une carte postale :
Merci, votre réponse m'encourage à continuer ce produit.

Si vous désirez distribuer ce programme, vous devez donner l'archive entière (compressé .lha ou décompressé). Vous êtes libre de mettre une copie de DB sur les disquettes de magazine (coverdisk). Si vous le faites, envoyez moi une copie de ce magazine. Les magazines semble être fénéantes depuis que DB soit apparu sur plusieurs revues, mais je n'en ai jamais reçu une seule. Allez !

Une permission spéciale doit être demandée pour inclure DB dans une collection de domaine public comme Fred Fish.

Comment recevoir la dernière version de db:

Le moyen le plus rapide de recevoir la version la plus récente de db

est de récupérer l'archive d'un site Aminet. Par exemple : ftp.luth.se (biz/dbase)

Si vous avez fait un script AREXX intéressant qui améliore db, ou bien dessiné une page de base facile à utiliser, partagez le avec d'autres qui peuvent en voir l'utilité.

Si vous êtes en train de faire une base de données avec une interface canon pour la communauté Amiga, mais vous ne voulez pas trop mettre de RFF, envoyez moi une lettre/email pour avoir la version 3.0 bêta de db avec un GUI-maker intégré.

Envoyez moi votre travail, comme ça je pourrais l'inclure dans l'archive. J'attends avec impatience la traduction de l'exemple 'Addresses'.

Si vous avez des suggestions ou des remarques sur ce programme, ou si vous trouvez un bug, dites le moi. J'aime recevoir des nouvelles des utilisateurs.

Ecrivez à cette adresse :

David Ekholm
Mantalsvägen 33
s-175 43 Järfälla
Sweden

Pour ceux qui préfèrent un moyen plus rapide :

phone: int +46 8-580 15668
email: david-ek@dsv.su.se

Merci à tous ceux qui m'ont donné leurs suggestions et trouvé des bugs.

Dessin de l'icône :

Icône Programme & Projet par Mikael Östergren
Autre icône par Roger McVey (r.mcvey@genie.geis.com)

Merci à Jan van den Baard pour GadToolsBox (used in v1.0 & v1.1)

Scripts AREXX additionnels par Richard Ludwig
email: md93-ril@nada.kth.se

Merci à Andrew Leppard pour le court (mais intense :-) code asm pour la composition des numéros en utilisant le haut parleur.
email: 9405571x@lux.levels.unisa.edu.au

Les remerciements vont aussi aux personnes suivantes pour les traductions :

Traduction allemande par Uwe Roehm
email: roehm@forwiss.uni-passau.de

Traduction polonaise par Michal Letowski
email: letowski@ci-1.ci.pwr.wroc.pl

Traduction Dutch par Edmund Vermeulen

email: edmundv@grafix.xs4all.nl

Traduction danoise par Christian Hoj
email: cbh@vision.auc.dk

Traduction suédoise par David Ekholm (zut, c'est moi :-))
email: david-ek@dsv.su.se

Traduction italienne par Michele Rotellini (50%)
(no email)
et Piergiorgio Ghezzeo (50%)
email: pghezzeo@oink.dsi.unive.it

Traduction finnoise par Jukka Kauppinen
email: Grendel@Freenet.hut.fi

Traduction française par Goncalves A. Georges
email: melkor@ramses.fdn.org
et Olivier ANH (tout sauf le .catalog)
Snail mail : Bugss c/o Rudy Mathey
44, rue Morion
F-33800 Bordeaux
France

Traduction norvégienne par Lars Magnus Nordeide
email: larsmn@oslonett.no

1.4 introduction

Une Introduction à db

db est un rapide et petit programme de base de donnée que j'ai écrit après avoir testé un nombre d'autre programmes de base de données du domaine public et j'ai toujours trouvé qu'il manquait quelque chose ou que quelque chose m'irritait. Il doit y avoir des douzaines de fonctions qui n'existent pas dans db, mais ils ne sont pas sensibles aux polices de caractères, n'ont ni une interface standard et un caractère OS 3.0.

Mon principal besoin était de garder les adresses et numéros de téléphone d'amis et de companies. Avant la version 2.0, db ne devait être qu'une base de données avec adresses et téléphones, mais ceci a changé. db a été dessiné avec une fenêtre modifiable par l'utilisateur en tête dès le départ. Actuellement vous devez utiliser un éditeur de texte pour spécifier les champs de ma base et la mise en page tant qu'il n'y a pas d'interface pour ça. (Voir

Le format de fichier RFF
)

De toute façon, c'est un boulot à temps complet, donc UTILISEZ le simplement et amusez vous.

Un support complet de création de l'interface sera probablement inclu dans le shareware.

S'il vous plait, Lisez ça! .

Lorsque vous utilisez db, vous remarquez que l'interface utilisateur a été fait le plus compact possible (peu de gadgets, menus et fenêtres). Toutefois la fonctionnalité de, par exemple, la recherche et le tri est grande. C'est intentionnel. Je préfère peu de boutons avec de grande possibilités que l'inverse. La requête d'ASL est à mon opinion un bon exemple d'un bon design. Ca semble simple, mais cache des fonctions comme la création automatique de répertoire et la complétion automatique de fichier. J'espère que vous dépenserez suffisamment de temps avec db pour découvrir ses fonctions cachées.

1.5 features

Listes de Fonctions

Une liste partielle des fonctions de db :

- o Gestion dynamique de la mémoire. Le nombre d'enregistrements et de champs ne sont limités que par la mémoire.
- o Basé sur la GadTool (Utilise les chaînes et les boîtes à cocher)
- o Marche à la souris et au clavier
- o Champs et pages modifiables par l'utilisateur
- o Multiple vues d'une même base de données
- o ClipBoard de Commodore pour de meilleur échange avec d'autres programmes
- o AppWindow - Glissez des icônes de base sur db pour les charger
- o Aide sur les menus en ligne - Pressez Help pendant la sélection des menus
- o Sensible aux polices de caractères
- o Requetes ASL pour des chargement/sauvegarde flexible
- o Localisé
- o Support AREXX
- o Compose un numéro en utilisant un modem ou un haut parleur
- o Analyse de la ligne de commande Shell ou WB suivant Commodore
- o Recherche rapide et flexible utilisant les motifs AmigaDos
- o Recherche Rapide par liste (ListView Browser)
- o Filtres
- o Tri rapide et flexible. Un tri multiple peut être spécifié
- o Exportation par vue et 2 fonctions standards d'exportation ASCII
- o Importation automatique ASCII (Séparation avec Tabulation)

1.6 future

Le futur

Ce qui suit sont les plans pour le futur. Les principaux changements seront probablement inclus dans un shareware. Lisez ça ! .

- o Un format de fichier simple à lire
 - o Une interface pour dessiner vos propres bases
-

1.7 system

----- Système requis & Compatibilité

db ne tourne que sous AmigaOS 2.04 ou plus.
db a besoin au moins de l'AmigaOS 2.1 pour utiliser la localisation.
db a besoin d'avoir toute la base en mémoire pour marcher. Cependant les ordinateurs moderne comme l'Amiga avec, disons, 2 mega de mémoire permet des bases de plus de 5000 enregistrements - suffisant pour la plupart des utilisations.

L'idée est la suivante : "on ne conduit pas un semi remorque en plein milieu de la ville". db est une rapide et petite voiture.

db alloue exactement la quantité de mémoire nécessaire pour chaque fichier. Vous ne spécifiez que le nombre maximum de caractère si vous avez une limite, pour une raison (code postal...). (Si vous ne spécifiez pas de limite, la limite par défaut est de 200 caractères). C'est un bon exemple pour la gestion mémoire.

db a été testé sous AmigaOS 3.0 et écrit en SAS C 6.55.

1.8 installing

----- Installation et mise en marche de db

db peut se lancer depuis le Workbench ou depuis le Shell.
Voir les

Préférences
pour la listes de paramètres.

Installer db est simple. Déposez l'icône du programme db et le tiroir Catalogs dans les répertoires voulus. (Si vous ommmétez le tiroir Catalogs, vous n'aurez que du texte en anglais.)

Pour lancer db depuis le shell, entrez juste db suivit de paramètres optionnels. Entrez db ? pour les voir.

Pour lancer db manuellement depuis le Workbench, double cliquez sur son icône.

Vous pouvez aussi lancer db en cliquant sur un projet db. Ce projet se chargera automatiquement.

Un autre moyen de lancer un projet est de simplement glisser ce projet sur la fenêtre de db et de relacher la souris.

1.9 settings

Préférences

db peut être modifié depuis soit le Shell soit ses types d'outils.
Les paramètres suivants s'appliquent aux 2 environnements :

DEVICE=<serial device name>	-Device utilisé pour composer un numéro avec un modem. Par défaut, c'est le serial.device, mais ceux qui possèdent un modem interne doivent mettre modem0.device ou dans ce genre.
UNIT=<device unit number>	-Pour de multiple unité série. 0 est par défaut.
TONEDIAL	-Compose en utilisant le haut-parleur.
TONEDIALSPEED=<number 1-10>	-Fixe la vitesse de la composition. Une valeur basse donne une grande vitesse.
CCITT5	-Utilise les fréquences CCITT5 lors de la numérotation, au lieu de DTMF.
DIALPREFIX=<préfixe>	-Chaine à ajouter avant le numéro lors de la numérotation. Par défaut c'est ATDT (Modems Hayes).
DIALPOSTFIX=<postfix string>	-Comme dessus, mais il sera ajouté après le numéro (+ code de retour). Par défaut ;H met le modem en mode attente et raccroche immédiatement.
AREACODE=<areacode string>	-Ajoute le code régional (area code) au numéro pour ne pas conondre les système téléphoniques. Disons que votre code régional est (08). Entrez ce code dans AREACODE fera que le numéro 123456 sera numéroté (08)123456.
PUBSCREEN	-Ouvre db sur un écran public particulier. Le Workbench si il n'est pas trouvé.
FONTNAME	-Vous pouvez spécifier une police de caractères aulieu de celle par défaut. Le nom doit être donné avec le .font, par exemple : FONTNAME=times.font
FONTSIZE	-Taille de la police (voir plus haut). Si la fenêtre est trop grande, db essaiera la police écran voire la topaz 8. Ces 2 types d'outils doivent être mis ensemble.
NOICONS	-Supprime la sauvegarde avec l'icône projet.

NOSPEEDRENDER	-Arrête SpeedRender
NORETURNSTEP	-Arrête la possibilité d'utiliser Return ou Enter pour passer d'un champ à l'autre. Presser Alt en pressant return ou enter remet temporairement en marche.
NOSERIAL	-Ce type d'outil n'a pour le moment pas de fonction.
HORIZBAR	-Place l'ascenseur horizontalement en bas de la fenêtre au lieu du coté droit. Permet aussi l'utilisation des touches gauche-droite pour se déplacer.
NOBORDER	-Affiche les champs sans les bordures de v37. Utilisez le si vous pensez que la fenêtre est trop chargée. Note: Cette option utilise SpeedRender
HIGHLABEL	-Eclaire les champs actifs. Cette option va bien avec l'option NOBORDER.
ESCQUIT	-Permet à la touche Escape de quitter db (en mode normal)
MAKEBACKUP	-Permet la sauvegarde d'un fichier backup chaque fois que la base est sauvée. (.bak est ajouté au nom) Je recommande ardemment de l'utiliser.
LOCALESORT	-Fait que db utilise la locale.library pour comparer lors du tri les caractères spécifique à la langue. Cependant, dû à un bug dans le suédois, les suédois ne doivent pas utiliser ce type d'outils. LOCALESORT peut aussi améliorer considérablement la vitesse de tri.

1.10 ref

Manuel de référence

=====

Cette section contient que la plupart des gens découvrent par eux-même en testant le programme. db possède une aide en ligne qui devrait suffir pour la plupart des gens.

Je recommande que vous testiez d'abord le programme puis reveniez à

cette section si il y a un problème.

1.11 window

La fenêtre

La fenêtre de db s'ouvre normalement sur l'écran public par défaut, en général le Workbench (tant que vous n'utilisez pas l'outil PUBSCREEN). Il utilise la police de caractères de l'écran (ou celle que vous avez spécifié) et tous les textes et gadgets sont retaillés. db n'a pas de gadget Zoom pour réduire la taille de la fenêtre, si db n'est pas utilisé pendant quelques moments.

db affichera un enregistrement à la fois que l'on peut appeler l'enregistrement courant. Cependant vous pouvez voir un champ particulier de plusieurs enregistrements à la fois en utilisant le "Browser" qui est une fenêtre avec un gadget ListView (db v2.7). db peut travailler sous 3 modes:

- o Mode Normal. Entrez les enregistrements et regardez la base de là.
- o Mode Recherche. Les informations entrées dans les champs servent comme motifs de recherche. Les motifs générique AmigaDOS sont acceptés (? et #?). Pressez Enter pour commencer la recherche ou ESCape pour l'annuler. (Pour que Enter marche, il faut d'abord désactiver le champ actif).
- o Mode Tri. Les nombres entrés dans les champs servent d'ordre de tri. Pressez Enter pour commencer le tri ou ESCape pour l'annuler. (Pour que Enter marche, il faut désactiver la champ actif).

La barre de la fenêtre peut ressembler à ça :

Adresses/Vue principale 16/43

Ceci indique un nombre de chose, c'est à dire :

Nom du fichier/Nom de la vue Numéro de la vue/Nombre d'enregistrements

1.12 view

Les Vues

Les informations de la base peuvent être divisées en plusieurs vues. Ceci permet à l'utilisateur de voir la base de différente façon.

Une vue peut contenir certaines informations sur l'adresse, alors qu'une autre peut avoir d'autres informations sur les résultats d'étude de ces personnes. C'est très utile lorsqu'on exporte pour imprimer des étiquettes si un style d'étiquette de vue est choisie.

Un autre raison d'utiliser les vues est de permettre à db d'avoir plus

de champs qu'il est possible d'afficher à la fois.

Si vous ne saisissez pas l'idée, essayez quelques programmes exemples et amusez vous avec le menu Vues et regardez ce qui se passe. (Seul quelques exemples ont plusieurs vues).

1.13 gadgets

Les Opérations par les Gadgets

La fenêtre de db est faite de 2 types de gadgets, les gadgets programme, et champs. Actuellement il y a 3 gadgets programme - le scroller avec ses 2 flèches. Ces gadgets ont 2 fonctions, la première est de naviguer dans la base de façon rapide et la seconde est de savoir sa taille. Ceci est illustré par la taille et la position du bouton du scroller.

Les gadgets de champs peuvent être de 3 types : chaîne, boîte à cocher et cycle. Ils peuvent être activés de 3 façons :

- o Cliquez avec la souris sur le champs.
- o Pressez la touche correspondant au caractère souligné du nom de champ. (Cela cochera la boîte à cocher et cyclera le cycle).
- o Pressez tab ou shift-tab. (champ de chaîne seulement)

Voir

Opérations par touches
pour plus d'informations.

Utilisez la touche Help ou le bouton droit pour désactiver un champ de chaîne. Les boîtes à cocher stockent leur états en 0, 1 dans la base. Les cycles stockent le choix en 0, 1... (Et sous forme ASCII comme toujours dans db).

En utilisant les champs de chaîne, il est possible d'entrer plus d'informations qu'il est visible dans le gadget. Si c'est le cas, le texte entré se déplacera. Pressez Shift gauche ou droit déplacera le curseur au premier et dernier caractère du champ, comme d'habitude. Amiga-X effacera de même le gadget actif et Amiga-Q défera (Undo) ce que vous aurez tapé. En utilisant un programme domaine public tel que NewEdit, vous pourrez aussi couper et coller les informations entre les champs., et pas seulement les enregistrements de db.

Vous n'avez pas à cliquer en dehors d'un gadget de chaîne ou presser Enter pour pouvoir naviguer dans la base. De même lorsque vous avez fini d'entrer les données, il n'est pas nécessaire de presser Enter pour quitter un champ. Toutes les informations entrées sont enregistrées dans la base.

1.14 keys

Les Touches

Les touches suivantes peuvent être utilisées pour contrôler db (en plus des raccourcis-clavier). En mode normal, on a :

Haut	- Enregistrement précédent.
Bas	- Enregistrement suivant.
Shift Haut	- Premier enregistrement.
Shift Bas	- Dernier enregistrement.
Return	- Recherche avant.
Shift Return	- Recherche arrière.

En mode tri et recherche, les touches sont :

ESC	- Quitte le mode.
Return/Enter	- Commence le tri/recherche.

Pour se déplacer entre les champs, utilisez le Tab et Shift-Tab comme il l'est en standard aujourd'hui. Vous pouvez aussi utiliser les touches Return et Enter si le type d'outil NORETURNSTEP n'est pas spécifié. Si vous n'utilisez pas NORETURNSTEP vous avez à désélectionner le champ actif pour être capable d'utiliser la touche Return pur commencer la recherche.

1.15 menus

Les Menus

Notez que la plupart des informations peuvent être trouvés grâce à l'aide en ligne de db en sélectionnant un menu et pressant la touche Help.

Projet/Nouveau:

Efface tous les enregistrements de la base, mais garde les champs. db vous demandera de confirmer avant d'effacer la base si elle n'a pas été sauvée.

Projet/Ouvrir...:

Affiche une requête ASL qui permet à l'utilisateur de sélectionner une nouvelle base à charger. db avertira si la base précédente n'a pas été sauvée. Les bases de db peuvent aussi être glissées directement sur la fenêtre pour être chargées.

db accepte les fichiers ASCII avec des champs séparés par des tabulations ainsi que des

fichiers au format RFF

.

Projet/Sauver:

Sauve la base actuelle sous le nom connu en utilisant le
format de fichier RFF
. Si aucun nom n'est donné, db se mettra
en mode "Projet/Sauver sous...".

Projet/Sauver sous...:

Fait de même que "Sauver" décrit plus haut, sauf qu'il affiche une
requête ASL afin que l'utilisateur choisisse un nom et un chemin pour
sa base. L'utilisateur sera averti si un fichier de même type existe.
Sinon un nouveau fichier et une nouvelle icône seront créés et le
Workbench sera informé de cette création d'icône.

Projet/Exporter/la vue...:

db n'a pas de générateur d'interface. L'idée est que d'autres
programmes, meilleur dans la mise en page, comme les programmes de
PAO sont capables de gérer ça. Au lieu de ça, db sauve sous un format
ASCII formaté la
Vue
actuelle qui peut être une étiquette ou
autre chose, afin d'être importé dans un autre programme. Un script
pour PageStream est inclu. Il affichera une requête ASL.
db demandera aussi à l'utilisateur, si tous les enregistrements
doivent être exportés ou uniquement ceux trouvés pendant la recherche.

Projet/Exporter/La vue avec noms...:

Ce menu marche comme Exporter/la vue" ci-dessus, mais ajoute les
noms de fichiers de la vue actuelle devant chaque champ. C'est bien
pour les impressions génériques de la base.

Projet/Exporter/ASCII avec Tabulation:

Ce menu affiche une requête ASL pour sauvegarder la base en fichier ASCII
dont les champs sont séparés par un Tab. C'est utile pour exporter des
bases de données vers des programmes comme Excel.
db demandera aussi à l'utilisateur, si tous les enregistrements
doivent être exportés ou uniquement ceux trouvés pendant la
recherche.

Projet/Exporter/ASCII avec Virgule:

Ce menu affichera une requête ASL pour une sauvegarde en fichier
ASCII dans lequel les champs sont séparés par des virgules et des
guillemets "" autour des noms. C'est utile pour exporter des bases
vers des programmes comme ProWrite.

db demandera aussi à l'utilisateur, si tous les enregistrements doivent être exportés ou uniquement ceux trouvés pendant la recherche.

Projet/A propos....:

Affiche une requête montrant des informations sur l'auteur, fonctions cachées, numéro de révision et port AREXX. Cliquez sur OK pour faire disparaître la requête.

Projet/Quitter:

Demande d'abord à l'utilisateur lorsque la base n'a pas été sauvée et si la sauvegarde est confirmée, enlève la base installée puis quitte le programme.

Edition/Couper:

db utilise le clipboard standard de Commodore pour copier des informations entre db et d'autres programmes (traitements de textes, PAO...).

db écrit dans 2 clipboards, l'unité 0 (le clip par défaut) et l'unité 1. Les lectures ne sont fait que de l'unité 1. Voici ce qui est écrit :

Unité 0: L'enregistrement courant avec les champs formattés comme la vue courante.

Unité 1: L'enregistrement entier dans un format tel que celui-ci:
Nom_de_champ <TAB> contenu <NL>
Nom_de_champ <TAB> contenu <NL>
...

Couper copie le contenu des enregistrements dans le clipboard et efface l'enregistrement courant.

Edition/Copier:

Copie juste l'enregistrement courant dans le clipboard. Voir Edition/Couper pour plus d'information.

Edition/Coller:

Ajouter un nouvel enregistrement et le colle le contenu du clipboard dans le champ correspondant à cette base dans le nouvel enregistrement. Edition/Coller collera le clipboard unité 1 qui a été décrit dans Edition/Couper.

Si il n'y a pas de donnée dans le clipboard qui passent dans les champs, rien ne se passera.

En utilisant le clipboard public ne permet pas seulement de copier et coller entre db et traitement de texte. Plusieurs instances de db peuvent aussi déplacer les champs entre eux.

Edition/Ajouter:

Ajoute un enregistrement dans la base. Le nouvel enregistrement sera ajouté après l'enregistrement courant.

Edition/Effacer:

L'enregistrement courant est effacé en mode normal.
Dans le mode recherche et tri, les champs sont vidés.
Vous ne pouvez pas effacer un enregistrement si il n'y en a qu'un.

Menu Vues:

Utilisez ce menu pour passer d'une
Vue
a une autre de
la base (si il en a plus d'une bien sûr). Voir la section sur les
Vues pour plus d'informations.

Action/Chercher...

Met db en mode recherche. La fenêtre courante indiquera le motif de recherche désiré.

- * Entrez un motif de recherche dans un ou plusieurs champs.
- * Pressez Enter/Return pour quitter les champs.
- * Après un second Enter/Return, db commencera la recherche dans la base de donnée du début vers la fin et s'arrêtera à la première occurrence trouvé.

En général, peu de lettres sont nécessaires comme motif de recherche. Par exemple:

'da' permet de trouver aussi bien 'David' que 'Daniel'.

Vous pouvez aussi utiliser les motifs AmigaDOS. Par exemple:

'#?d' ou '*d' sélectionnera les champs terminés par un d.
(david|micke)' trouvera 'David' et 'Micke'.

Presser la touche ESCape ou cliquer la bouton de fermeture de la fenêtre remet db dans le mode normal.

Note: La fonction export utilise cette fonction pour filtrer les enregistrements.

Note! Les champs de boites à cocher et cycles sont ignorés dans

la recherche et le tri.

Pour avoir le possibilité d'utiliser tous les champs, l'utilisateur doit construire une vue qui ne contient que des gadgets de chaîne.

Une fonction 'recherche rapide' a été ajoutée à la touche espace. En pressant l'espace, db entre en mode recherche et vide tous les champs automatiquement. (v2.5)

Action/Chercher suivant:

Ce menu n'est là que pour la compatibilité avec d'autres programmes. Il continuera la recherche pour un enregistrement compatible avec la demande. C'est mieux d'utiliser Enter/Return et Shift Enter/Return.

Action/Trier...

Met db en mode tri. L'enregistrement courant indiquera l'ordre de tri désiré. Entrez juste le numéro des champs. Tout autre chose que les chiffres sont ignorés. Par exemple, entrer un '1' dans le champ Code Postal et un '2' dans le champ Nom indique que vous voulez trier d'abord les codes postaux puis les noms.

db triera les caractères accentués correctement à l'inverse de nombreux autres programmes. En pressant ESCape ou en cliquant sur le gadget de fermeture de la fenêtre remet db en mode normal.

Note! Les champs de boîtes à cocher et cycles sont ignorés dans la recherche et le tri. Pour avoir le possibilité d'utiliser tous les champs, l'utilisateur doit construire une vue qui ne contient que des gadgets de chaîne.

Action/Composer

db composera un numéro de téléphone en utilisant votre modem ou un haut-parleur. Voir les Préférences pour savoir comment configurer votre modem et le port série etc.

Pour composer un numéro en utilisant le modem :

1. Sélectionnez le gadget contenant un numéro de téléphone correct (c'est à dire, au moins un chiffre)
2. Sélectionnez le menu ou pressez Amiga-D.
3. Lorsque db commence à composer, prenez le combiné et attendez. Dès qu'il y a une connection, db raccrochera et vous pourrez prendre le suite.

Pour arrêter la composition de db, pressez une nouvelle fois Amiga-D. Ceci devrait marcher avec la plus part des modems.

Composer en utilisant le haut-parleur est identique :

1. Sélectionnez le gadget contenant un numéro de téléphone correct (c'est à dire, au moins un chiffre)
2. Levez le combiné et placez le près du haut-parleur avec un bon volume.
3. Sélectionnez le menu ou pressez Amiga-D.

Vous pouvez aussi utiliser ARexx pour simplifier la composition en ne double-cliquant que sur le champ. Voir les scripts d'exemples.

Fans du clavier, notez ceci : Au lieu de double-cliquez, vous pouvez utiliser Amiga-gauche + le raccourci défini dans le champ désiré (marqué par un souligné _ dans le champ).

Action/Parcourir...

Le browser (to browse = parcourir) est une fenêtre avec un listview qui permet de parcourir rapidement la base de données, en voyant les champs de plus d'un enregistrements. Pour l'utiliser, choisissez un champ à parcourir et choisissez Parcourir...

Vous pouvez aussi double-cliquez sur un champ pour ouvrir le browser ou utiliser Amiga-gauche + un raccourci si un est défini pour ce champ (du moins tant qu'aucune commande ARexx n'est installée).

Fans du clavier, notez ceci : Au lieu de double-cliquez, vous pouvez utiliser Amiga-gauche + le raccourci défini dans le champ désiré (marqué par un souligné _ dans le champ).

La fenêtre principale est inaccessible lors de l'utilisation du browser, mais la fenêtre reste ouverte, comme ça vous pouvez voir tous les champs de l'enregistrement choisit.

Ce qui suit n'est pas évident :

- Cliquer un fois pour voir l'enregistrement dans la fenêtre principale.
- Double-cliquez pour choisir un enregistrements ET quitter le browser. (La touche Return marche aussi)
- Vous pouvez utiliser les touches de curseur pour parcourir la base. Essayez la combinaison avec Shift et Alt.
- Le browser a une fonction de 'completion' : Tapez juste la chaîne que vous voulez voir et le browser recherchera simultanément. Utilisez Del pour effacer le tampon de completion et backspace pour effacer un caractère. Le texte entré est affiché dans la barre de titre du browser. Utilisez le Tab, Shift-Tab pour sauter entre plusieurs enregistrements trouvés.
- Vous pouvez quitter le browser en pressant ESCape.

Préférences/Avertissements:

Avec ce menu sélectionné, l'utilisateur sera averti avant l'effacement d'un enregistrement (Kill) non-vidé.

Préférences/Direction du tri

L'utilisateur peut aussi bien choisir une direction de tri de A vers Z ou l'inverse. db triera aussi bien les caractères accentués.

Préférences/Définition champ...

THIS FEATURE IS CURRENTLY UNDER CONSTRUCTION AND THEREFORE DISABLED

This is the door to db's field definition section. It allows you to add, edit and remove fields using a user friendly GUI.

Note! You are just changing the internal fields here. Use "View design..." to design the visual apperance of your fields.

Préférences/Mise en vue...

THIS FEATURE IS CURRENTLY UNDER CONSTRUCTION AND THEREFORE DISABLED

This is the door to db's View design section. It allows you to add, edit and remove visual fields in the current View using a user friendly GUI. The visual fields are handled similar to how character are handled in a wordprocessor. This is WYSIWYG (What You See Is What You Get).

Note! You don't work with any real fields here, just a visual presentation of a subset of the database fields. Use "Field definition..." to define the database fields.

ARexx/Execute commande ARexx...

Ce menu affichera une requête ASL pour executer des scripts AREXX. En dessous de la barre horizontale, se trouve de la place pour les scripts AREXX prédéfinis.

Voir

Installer un menu AREXX

.

1.16 arexx

Support AREXX

Le support ARexx de db a été amélioré afin de permettre à l'utilisateur d'écrire des scripts pour vérifier l'unicité des enregistrements, implementer des liens entre bases, connecter db à

des browsers WWW et autres.

Auparavant db n'avait pas de support ARexx complet tant que j'avais l'intention de distribuer de shareware. Je le fait quand même mais cette version à d'autres avantages. L'ancien support ARexx n'est/était pas maldu tout. Il permet de faire de chose comme:

- * Montrer des images, textes et jouer des sons etc depuis db.
- * S'assurer que les données entrées dans les champs sont formatés comme vous le voulez (MAJUSCULES...)
- * Etendre les codes en leur noms entiers etc (un sorte de "complétion de nom").

Pour arriver à celà, les programmes ARexx peuvent être appelés de 3 façons :

- * Lorsque l'utilisateur choisit une entrée dans le menu AREXX ou presse la touche de fonction correspondante. (db2.10)
Voir
 plus bas
 pour une aide sur l'installation
d'un menu AREXX.
- * Lorsque l'utilisateur double-clique sur un champ ou presse Amiga-gauche + la touche soulignée d'un champ de la vue. Ceci est appelé une invocation 'à la demande' et est parfait pour montrer des images comme ci-dessus. L'invocation est asynchrone, le programme ARexx tourne simultanément avec db. Utilisez BLOCKINPUT et FREEINPUT pour prévenir les problèmes ici.
- * Lorsque l'utilisateur appuie sur Esc, Help, Enter ou Return pour quitter un champ ou manipuler une boite à cocher ou un cycle. Ceci est appelé une invocation automatique et peut être utilisé pour formater l'entrée de différente façon. Ici l'invocation est synchrone alors n'écrivez pas de scripts lents qui bloquerons l'utilisateur pendant l'exécution.

Vous pouvez spécifier les fichiers ARexx ou commandes à executer pour chaque champ de chaque vue, pour une vue entière ou pour le projet entier. Actuellement vous avez à éditer les lignes RFF pour contrôler db. Voir

 Le format de fichier RFF
 pour plus
d'informations.

Il y a des scripts ARexx prêt à l'emploi pour des utilisations courantes dans cette archive. Essayez les avec les exmples de projets qui les utilisent.

La première fois que db est appelé, il ouvre un port appelé 'db.1'. Si plusieurs programmes sont lancés ilauront un nombres plus grand ('db.2', 'db.3'..). Vous pouvez changer le nom de base (c'est à dire 'db') en autre chose en spécifiant le tag RFF : RXPORNAME dans la base.

1.17 installarexxmenu

Installer un menu AREXX

Depuis db2.10, le menu AREXX est apparu. Il vous permet d'ajouter des commandes AREXX de votre choix comme entrée. En utilisant ça, vos commandes ne restent plus coincés dans un champ particulier. Ajoutez les commandes qui vous manquent ici.

Pour faire un menu AREXX personnalisé, vous devez utiliser un éditeur de texte et écrire une ligne de code RFF dans le fichier. Un exemple vous le montrera plus clairement :

Disons que vous voulez qu'un menu ressemble à ça :

```
Merge
Vérifier double enregistrements (Check for duplicate)
```

Sur votre disque, vous avez 2 fichiers AREXX pour ces fonctions. Ils s'appellent Merge et Dupl. La ligne RFF qui installe ces commandes dans le menu peut ressembler à ça :

```
'@'rff=1.2,type=rxmenu rxfile=Merge rxfile=dupl,name="Vérifie double ←
enregistrements"
```

(Ignorez les apostrophes. Notez qu'il y a des tabulation au dessus, pas des espaces)

Dans le premier cas, on veut utiliser le nom de fichier "Merge" dans le menu, c'est pour cela qu'on a seulement spécifié le nom de fichier en utilisant la commande RFF RxFile. Dans le second cas, on ne veut pas afficher le nom bizarre "dupl" dans le menu, alors on donne à cette option un nouveau nom. On peut aussi utiliser la commande RxString pour spécifier une ligne de commande à exécuter au lieu d'un fichier disque. De cette façon vous pouvez entrer directement des commandes AREXX dans le code RFF.

Les 10 premières entrées sont automatiquement liés aux touches de fonctions.

1.18 arexxcommands

Listes des commandes AREXX

Ceci est la liste des commandes ARexx actuellement supportées dans db. Certaines commandes font certaines tâches dépendant de leur appel avec ou sans paramètre.

Par ex. 'CurrentRecord' retournera l'enregistrement courant et 'CurrentRecord <x>' positionnera l'enregistrement courante sur <x>)

Les commandes retournent les réponses dans la variable 'Result'. Dans le cas d'un avertissement, 'RC' sera mis à 5. n'oubliez pas

de spécifier 'Options Results' dans votre scripts pour avoir les réponses dans ces variables.

Commandes en relation avec les enregistrements

Add

Comme dans le menu.

Kill

Comme dans le menu

(Note: cette commande effacera les champs si on est en mode Recherche ou Tri).

FirstRecord

Saute au premier enregistrement de la base. Retourne "1"

NextRecord

Saute à l'enregistrement suivant et retourne son numéro.
RC sera positionné si il n'y a plus d'enregistrements.

CurrentRecord [<nombre>]

Retourne le nombre de l'enregistrement courant si aucun argument n'est donné. Saute à l'enregistrement donné. RC sera positionné si le <nombre> est illégal.

RecordSum

Retourne le nombre d'enregistrements de la base.

FindFirst

Recherche de puis de début de la base. (Fixez les critères de recherche dans le mode Recherche, en remplissant les champs).
Si db trouve une occurrence, et qu'il se trouve en mode Recherche, le programme retournera automatiquement en mode 'Normal'. Retourne le numéro de l'enregistrement courant. Si la recherche a failli, RC sera positionné.

FindNext

Recherche l'occurrence suivante. Sinon fait comme FindFirst.

Cut

Comme le menu.

Copy

Comme le menu.

Paste

Comme le menu.

Merge

Mélange l'enregistrement courant avec le contenu du clipboard.

Commandes en relations avec les champs

(Ces commandes marche avec les champs internes. Il n'y pas de 'Vue' en ARexx.)

FirstField

Met le 'champ courant' au premier sur le premier champ et retourne son nom.

NextField

Met le 'champ courant' au champ suivant et retourne son nom.
Met RC à 5 si il n'y a plus d'autres champs.

CurrentField [<champ>]

Met le 'champ courant' sur le champ <champ>. Si aucun argument n'est spécifié, il retourne le nom du 'champ courant'. Au départ, le 'champ courant' est le même que le dernier champ activé.

GetField [<champ>]

Le contenu du champ courant (habituellement le dernier champ activé) ou le champ spécifié (nom de champ interne) sera retourné dans la variable Result. Pour les boites à cocher et cycles, leur valeur seront retournés pas d'espace ou de nom.

PutField [<var>]

Met <var> ou rien dans le champ courant. Utilisez 'CurrentField' d'abord pour spécifier un autre champ d'écriture.

Autres commandes

Mode [<mode, soit "Normal", ou "Find" ou "Sort">]

Met db dan l'un des 3 modes. Ceci marche comme dans les menus.
Retourne le nom du mode courant si aucun paramètre n'est donné.

Save

Sauve le projet actuel sur disque.

Sort

Tri la base suivant l'ordre donné dans le mode 'Tri'.

Quit

Quitte db sans condition (ne demande pas de sauvegarde)

BlockInput

Cette commande fait 2 choses :

1. Bloque l'entrée de l'utilisateur et met le pointeur en attente pour éviter que l'utilisateur ne modifie des choses pendant que le programme ARexx tourne.
2. Annule la mise à jour automatique de l'interface tant que vous contrôlez db depuis ARexx. db devient 'silencieux'. Les mises à jours peuvent être maintenant faites à la demande avec UpdateGUI.

FreeInput

Libère l'entrée de l'utilisateur, remet le pointeur en mode normal et met à jour automatiquement l'interface.

UpdateGUI

Si vous avez mis BlockInput, l'interface ne sera pas mise à jour lorsque vous contrôlez db depuis ARexx tant que vous ne lui demandez pas explicitement avec cette commande.

WindowToFront

Met la fenêtre de db en avant.

ScreenToFront

Met l'écran de db en avant.

ActivateWindow

Active la fenêtre de db

DisplayBeep

Fait flasher l'écran pour indiquer que quelque chose va mal.

Okay1 <message>

Affiche une requête pour informer l'utilisateur de quelque chose. Rien n'est retourné. La requête à un bouton 'Ok'.

Okay2 <message>

Affiche une requête demandant à l'utilisateur de faire un choix. La requête a 2 boutons. Un bouton 'Ok' et un 'Cancel' retournant respectivement 1 et 0.

CurrentGadget <gadget ou index>

Selectionne le gadget qui à le nom <gadget> ou ressemble à <index> comme gadget courant. Le gadget courant est celui qui s'active lorsque l'utilisateur presse la touche de tabulation.

RetryInput

Reactive le dernier gadget utilisé. Pour activer un autre gadget, utilisez d'abord 'CurrentGadget'. Cette commande est souvent utilisée pour forcer une entrée correcte.

Dial <number>

Compose le numéro donné en utilisant votre modem ou le haut-parleur.

(Rappelez vous d'utiliser 'Options Results' dans vos scripts pour avoir les réponses. Aussi :
Les commandes DOS TCO, TCC, TS et TE aident beaucoup pour le debuggage des scripts.)

1.19 rff

Informations techniques

=====
Le format de fichier RFF

db est un programme de base de donnée général, c'est à dire qu'il n'a pas un nombre de champ fixe. Mais actuellement il n'y a aucune interface d'édition des champs que vous voulez.

Pour créer une base personnelle en utilisant db, vous devrez d'abord éditer le fichier de donnée manuellement. db stocke toutes les informations autres que les noms de champ dans les lignes appelées RFF (Les noms de champ sont stockés dans la première ligne d'un fichier comme dans un texte ASCII standard pour fichiers de base de donnée).

db utilise une version étendue du format de base de donnée ASCII standard appelé RFF. La différence entre les 2 est que RFF est capable de stocker des informations tel que la mise en page, les noms de champs affichés (comparés au nom de champ interne dans le format de fichier ASCII standard), longueur maximale des champs, types de champs et plus.

Cependant un fichier RFF peut être convertis en une base de donnée ASCII en effaçant simplement les lignes commençant par @RFF. Dans le standard RFF, toutes les lignes RFF doivent être au début du fichier, mais après la première ligne qui est celle des nom de champ, en accord au standard de base de donnée ASCII.

Notez que dans les versions précédentes (1.1 et 1.0) db ignorait l'information spécifique au standard RFF, mais écrivait les fichiers de façon à être compatible à RFF.

Normalement, il n'y a qu'une ligne RFF par

Vue

dans un fichier,

mais la première ligne RFF décrit les informations internes (à opposé aux informations visuelles) comme la longueur maximale des champs. Ainsi un fichier RFF de 2 lignes, a une vue, c'est à dire une fenêtre affichant une seule façon de voir la base.

A la suite se trouve le format d'un ligne RFF :

```
{<identificateur>=<donnée>[,<identificateur>=<donnée>]...[<tab>]}...<NL>
```

Exemple:

```
NAME=_Entreprise,OFFS=0,SIZE=37 NAME=_Nom,OFFS=1,SIZE=14,NEXT=space
```

Ceci doit être lu : Un ou plusieurs champs 'identificateur=donnée' séparé par des virgules. Les groupes et champs identificateurs=donnée séparé par des virgules doivent être séparé par une tabulation. Les majuscules ne sont pas significatives dans l'identification. Si vous avez besoin d'utiliser une virgule ou un espace comme donnée, enfermez les données dans des "guillemets".

L'idée est que toutes les informations qui appartiennent à une vue sont rassemblées sur une seule lignes, et que toutes les informations concernant un champ d'une vue sont mises entre 2 tabulations tout comme les champs de données.

LIMITE DES POUVOIRS : GLOBAL AREA ET LOCAL AREA

Une ligne RFF 1.1 est divisée en 2 régions, la zone globale et la zone locale. La zone globale est la zone AVANT la première tabulation, et la zone locale est le reste de la ligne. Les identificateurs mis dans la zone globale affecte la vue entière (ou tout le projet pour

une ligne RFF interne). Les identificateurs d'une zone locale n'affecte que ce champ. Le tout marche comme les variables locales et globales dans les langages d'ordinateur comme le C ou le Pascal.

Certains identificateurs peuvent apparaître dans plusieurs zones, voire plusieurs fois, alors que d'autres n'apparaissent qu'une seule fois dans une zone spéciale. Ceci est indiqué après la spécification du tag dans la liste plus bas.

Pour expliquer comment les tags sont utilisés, j'ai inclu une entrée Status: dans la liste des identificateurs. Voici une explication :

global	-peut apparaître dans le zone globale d'une ligne RFF.
local	-peut apparaître dans le zone locale d'une ligne RFF.
first	-doit être le premier tag.
internal	-peut aussi apparaître dans une ligne RFF interne.
internal-only	-doit uniquement être dans une ligne RFF interne.
required	-ne doit pas être ommis.

Les identificateurs inconnus sont ignorés mais gardés dans le programme jusqu'à la sauvegarde. C'est un peu comme à la manière des fichiers IFF qui permet des améliorations futures sans perdre une compatibilité descendante. Dans la version 2.4, l'utilisateur a la possibilité de spécifier les champs boites à cocher et cycle, et pas seulement chaîne. De tels fichiers peuvent être chargés dans une vieille version de db, être éditée puis être rechargée dans un version récente de db sans aucune erreur de codes ni perte d'informations.

J'ai aussi choisit le format RFF car il est LISIBLE. Lisible par aussi bien les hommes que les machines de types différents. Pour d'autres programmes de base de donnée, un fichier RFF devrait apparaître comme un fichier ASCII normal avec quelques enregistrements bizarre à au début. Pas mal, non ?

1.20 identifièrs

Les identificateurs RFF:

'@'RFF=<version.revision> (Ignorez les 'apostrophes')

L'identificateur de ligne RFF en personne. Il a un numéro de version et de révision. Il doit être le premier identificateur d'une ligne RFF. Un nouveau numéro de version indique à un vieux analyseur RFF que beaucoup de changement ont été fait que la ligne entière doit être ignorée.

Status: global, first, internal, required

TYPE=<type de donnée décrit>

Cet identificateur décrit ce que la ligne RFF courante décrit.

Les paramètres peuvent l'un de ce qui suit :

internal	-Informations ne concernant aucune vue.
form	-C'est une vue formatée.
list	-C'est une vue de liste (actuellement, t non implémentée)

Status: global, internal, required

LNAM=<nom de page>

C'est le nom de la vue spécifiée. Il sera affiché dans les menus et dans le titre de la fenêtre de la base lorsque cette vue est choisie.

Si vous omettez LNAM le nom de fichier est utilisé.

Status: global

TABSIZE=<nombre de caractères>

Lorsqu'on utilise la tabulation pour séparer visuellement les champs (voir le tag NEXT plus bas), ce tag fixe la distance entre 2 positions de tabulations (mesuré en caractères).

Si vous omettez TABSIZE, un défaut de 6 est utilisé. (db2.1)

Status: global

RXPORTNAME=<nom de port>

Nom de base du port Rexx de db. db ajoutera .1, .2 etc. au nom de base si nécessaire. (db2.9)

Status: global internal only.

RXFILE=<nom de fichier>

Nom du fichier ARexx à exécuter si l'utilisateur double clique sur un champ de chaîne ou presse Amiga-gauche + touche, où la touche correspond au caractère souligné dans le nom de champ. (db2.2)

Cette commande peut être utilisée dans le menu AREXX (2.10)

Status: global, local, internal

RXSTRING=<Programme de chaîne ARexx>

Nom du fichier chaîne ARexx à exécuter si l'utilisateur double clique sur un champ de chaîne ou presse Amiga-gauche + touche, où la touche correspond au caractère souligné dans le nom de champ. Cet identificateur a priorité sur RXFILE si les deux apparaissent en même temps. (db2.2)

Cette commande peut être utilisée dans le menu AREXX (2.10)

Status: global, local, internal

AUTORXFILE=<nom de fichier>

Comme RXFILE, mais n'exécute que lorsque l'utilisateur presse ESC, help, Enter ou Tab pour quitter un champ de chaîne. (db2.2)

Status: global, local, internal

AUTORXSTRING=<Programme de chaîne ARexx>

Comme RXSTRING, mais n'exécute que lorsque l'utilisateur presse ESC, help, Enter ou Tab pour quitter un champ de chaîne. (db2.2)

Status: global, local, internal

FLEN=<longueur maximum du champ>

Cet identificateur décrit la longueur maximum permise dans un champ. Il est utilisé pour calculer la taille des tampons pour

les gadgets de chaîne.

Note : db n'allouera jamais plus de mémoire ou d'espace disque que nécessaire pour remplir une chaîne, ainsi vous pouvez utiliser de grande longueur pour FLEN sans perdre de l'espace.

Si vous omettez FLEN, la longueur par défaut est de 200 caractères.
Status: local, internal-only

NAME=<nom d'objet>

Cet identificateur est utilisé pour nommer différent objet dans db. Avant v2.10, il décrivait le nom de champ qui devait être affiché (qui peut être différent au nom de fichier interne). A partir de maintenant, cette commande est sensible aux minuscules/MAJUSCULES, C'est à dire que le sens exact de cette commande dépend de la situation dans laquelle il est utilisé. Dans db2.10 cette commande définit aussi le nom de l'entrée-utilisateur dans le menu AREXX. Cependant son utilisation la plus simple est la définition du label d'un champ d'une vue et aussi décrit quel raccourci clavier utiliser pour activer ce gadget. Ceci est fait en plaçant un souligné avant le caractère qui doit agir comme raccourci clavier.

Exemple : NAME=E_mail, mettra la touche m comme raccourci clavier pour activer ce champ.

Si vous omettez NAME, le nom de champ interne sera utilisé (de la première ligne du fichier)

Status: local

PLACE=<direction>

Cette commande contrôle où le titre du champ doit être mis. Par défaut, db le met à gauche, mais vous pouvez spécifier "above" (au dessus) pour le mettre au dessus le champ. (db2.10)

OFFS=<offset du champ dans la base>

Cet identificateur est très important. Il aide db à 'connecter' un gadget de champ au bon champ de la base. En effet les champs affichés peuvent avoir un ordre différent des champs internes. Il peut y avoir même moins de champs affichés que de champs internes. Ceci est bien sûr utilisé seulement avec des vues multiples.

Si vous omettez OFFS, le dernier OFFS+1 est pris.

Status: local

SIZE=<taille affiché du champ en caractères>

Cet identificateur est utilisé pour calculer la taille horizontale des gadgets de chaîne.

Si vous omettez SIZE, la taille par défaut est de 25 caractères.

Status: local

NEXT=<visual separator between fields>

Ce tag contrôle la position des champs dans la fenêtre. Pas avec les coordonnées x-y comme d'autres programmes, mais en disant à db comment déplacer les 'crayons' invisible lorsque un champ a été dessiné.

db commence à dessiner à partir du coin en haut à gauche.

Voici les paramètres définis (Le retour à la ligne est par défaut):

- space -déplace d'un espace vers la droite le gadget suivant.
- tab -déplace d'une tabulation le gadget suivant.
- para -déplace de 2 lignes vers le bas avant le gadget suivant.

Status: local

CMNT=<"chaîne de commentaire">

Ceci donne la possibilité d'insérer des commentaires invisible dans la base. Les chaînes contenant espaces, tabulations, ou virgules doit être enfermées entre "".

Status: global, local, internal

FTYP=<type de champ>

Détermine le type de champ. Actuellement voici les types définis (la chaîne est par défaut) :

- integer -En fait un type numérique, non alphabétique.
(en interne seulement, v2.8)
- checkbox -affiche une boîte à cocher. Un 0 ou 1 est entré dans la base
- cycle -affiche un gadget de cycle avec les choix spécifiés plus bas. Le choix actif est stocké par son ordre en commençant par 0 dans la base.
- text -Affiche un gadget texte. Un gadget texte est un gadget de chaîne uniquement lisible. Il peut cependant être écrit depuis AREXX. (db2.10)
- calc -un champ de calcul, contenant une formule.
(non implémenté. Utilisez plutôt AREXX)
- external -Ce champ stocke le nom de fichier d'un fichier externe.
(non implémenté. Utilisez plutôt AREXX)

Status: local, internal

SFMT=<format de la chaîne>

Cet identificateur donne un formatage spécial comme ce qui suit (aucun formatage est pas défaut) :

- upper -Affiche tout en majuscule.
(Non implémenté. Utilisez plutôt AREXX)
- caps -Première lettres en majuscule des mots.
(Non implémenté. Utilisez plutôt AREXX)
- right -justification à droite du texte. (db2.6)
- center -justification centrée du texte. (db2.6)

Status: local

CENT=<Entrée du gadget cycle>

Cet identificateur peut apparaître plusieurs fois et dit à db quel sont les choix qui doivent apparaître dans un gadget de cycle. Si vous utilisez FTYP=cycle et ne spécifiez pas CENT, vous aurez une erreur au chargement du fichier.

Status: local, multiple

1.21 speed

Note sur SpeedRender

Vous vous demandez certainement ce que le type d'outils (NOSPEEDRENDER) signifie dans l'icone de db. Eh bien en voici la réponse :

J'ai essayé de programmer db de telle façon qu'il devra marcher avec des versions futures de l'OS. Mais pour atteindre une vitesse raisonnable dans l'affichage de la fenêtre, j'ai adopté une technique appelée SpeedRender. Ce que fait SpeedRender est de copier tous les pointeurs de bordure des gadgets dans une liste privée puis efface les pointeurs de champ GadgetRender dans les structures gadgets. Maintenant Intuition n'a plus à redessiner les bordures des gadgets chaque fois qu'un gadget est mis à jour (ça se passe chaque fois que l'utilisateur utilise le scroller).

Mettre ce tag annule la fonction.

1.22 history

Historique des changements

=====

(Note de traduction : Cette partie n'est pas traduite par principe)

95-11-22 v2.10 NEWS:

- * There is now a user defineable ARexx menu in db. Please look at the "Addresses2" example.
- * Needed RAM has dropped by 40% per record for a typical file as a result of a new memory handling scheme.
- * Read-only string fields are now supported (They can be written to from ARexx though). These fields are great to use as calculated fields. Their type in db is "text" (FTYP=text). Please look at the example "ARexxdemos/MagicFieldsIII".
- * The visible name of a field can be positioned above the field by specifying PLACE=above in the RFF code. Please Look at the example "ARexxdemos/MagicFieldsIII".
- * It is now considerably faster to quit a large database.
- * Switching between views is now faster as db no longer closes and reopens the window.
- * If you leave all fields blank when you sort a database it will be un-sorted to the state it was in upon loading. A kind of "undo sort".

- * Smaller code than in 2.8. =)

BUG FIXED:

- * Reading the contents of an empty string field in a newly created record from ARexx incorrectly resulted in a warning result.

95-09-27

v2.9 NEWS:

- * The ARexx support has been improved a lot.
db now supports more than

30 ARexx commands

.

You are now able to easily make advanced applications with relations and build complete economy systems. (There is an invoice system coming for Swedish users.) Check out the new example in Examples/Relations.

- * All icons are NewIcons from now on. NewIcons look much better than the original ones as they seldom show up in wrong colour. To see the new icons you have to install the NewIcons package. Otherwise the icons will look like before with the exception of some funny tooltypes in the icons.
(You can get NewIcons at Aminet:util/wb/NewIcons.lha)
- * I've included the original .ct files next to the .catalog files for each country to make it simpler for helpful and skilled translators to provide me with language updates as I release new versions.

95-08-26

v2.8 NEWS:

- * Sorting now uses the superior Quicksort algorithm (ACM 271). Alan Wigginton (author of QuickFile) notified me on the terribly bad sort benchmarks he got when sorting a large database using db. It could take hours on an Amiga 500. Now even large databases should sort in seconds. If you get into trouble when sorting a large database now, please set the stack to a higher value.
- * db now sorts numbers correctly if you add the FTYP=integer RFF tag to the internal field specification. Look at the "Music" example!
- * The whole database will be sorted on the active field if you hold down the shift key while activating the Browser. (A shortcut to sorting, one could say.) (Some users want to have just the Browser list sorted and some users want to have the whole database sorted simpler. This is an intermediate better-than-nothing solution until I figure out the best way to do it.)
- * Minor speed improvements when entering and exiting

the Browser has been made.

BUGS FIXED:

- * NEWS, new bugs they say...
When using the keys to navigate in the Browser under OS 2.0 or 2.1 (below v39) the system locks completely. My first Guru in db :-(This has been fixed. Thanks to Alan Wigginton for telling me about it. (My ordinary testers all seem to use v39+ today.)
- * Since v2.6 db didn't sort the Swedish ÅÄÖ characters correctly. This is fixed.
(Don't use the LOCALESORT ToolType if you want Swedish characters sorted correctly as there is a bug in the Swedish language support in the OS. But DO use LOCALESORT otherwise as this speeds up sorting and should make db sort your language specific text correctly.)

95-08-16

v2.7 NEWS:

- * Added a browser window that consists of a listview where you can see fields from many records at a time. The browser is handy when you don't even remember a keyword to search for or when you want an overview of the database. One way to start the browser is by doubleclicking a field. There is on-line help also.
- * Added two new tooltypes to further control the appearance of fields (HIGHLABEL & NOBORDER).
See
Settings
for more info.
- * The following catalogs are updated (please copy them to your directory): Norwegian, French, Polish, Swedish. (Text that has changed will show up in English if you use an old catalog.)

95-07-11

v2.6 NEWS:

- * An Italian, Finnish and Norwegian catalog is now added to the archive.
- * A handy ARexx script to simplify entering of dates is added to the ARexxdemos drawer.
- * Data in fields can now be shown centered and right-justified (nice for numbers) by setting the RFFtag SFMT. For more information, refer to the section on the RFF file format. To see right justified fields, look at the "Music" and "Game reviews" examples.

BUGS FIXED:

- * It was not possible to specify (%) to search for empty fields (note: db uses AmigaDOS search patterns. You can for example specify ~(%) to search for non-empty fields). This very old bug was found by Fredrik Jervfors.

- * The Norwegian catalog incorrectly used 'S' as a shorthand for both 'Save' and 'Search'.

95-02-18

v2.5 NEWS:

- * The sort function can now be set to consider language specific character sets using the LOCALESORT tooltype.
Note: This might not work in your country if Commodore hasn't set up your language's specific character set. If set, characters like ååå will typically be sorted next to the character a. LOCALESORT sorts more than five times faster than the normal sort when sorting the 250 records-in-size "Music" example on an Amiga 4000 with the cache turned off (!). I hope that speed is enough for you. Currently I'm using an improved shaker-sort algorithm. This one works fast if only a few records are misplaced, but I know that the quick-sort algorithm wins when a lot of records are misplaced.
- * A "fast find" function has been added to the space key. Upon pressing space, db will enter find mode and clear all fields automatically.

BUG FIXED:

- * The routines to update the database and gadgets have been rewritten to eliminate possible destruction of the contents of the current record when jumping between sort and find modes. This is an old bug.

94-11-20

v2.4 NEWS:

- * db now handles fields of checkbox and cycle type, as well as string fields. Take a look at the "Game Reviews" file in the Examples drawer.
- * db now also dials numbers using the loudspeaker. Thanks goes to Andrew Leppard for the dial code. See

Settings

for more info.

- (There are three new ToolTypes to accommodate this)
- * Added an 'Output view with names' menuitem allowing the database to be outputted with the fieldnames of the current view preceding each field.
- * Added a MAKEBACKUP ToolType that makes db save a backup of the database every time a database is saved.
- * Added an OKAY1 and OKAY2 ARexx command enabling the script programmer to put up information and selection requesters.
- * On-line help somewhat improved.
- * An Italian and Finnish catalog is now added to the archive, making db speak 8 languages!
(Where are you French and Spanish guys?)

BUG FIXED:

- * Reading the contents of an empty string field from

ARexx returned garbage.

94-09-28

v2.3 NEWS:

* Several language catalogs added. See Preface

- * Clicking the closewindow gadget when in sort- or find mode returns db to normal mode instead of quitting as suggested by Edmund Vermeulen. In normal mode this quits db as always.
- * Added a HORIZBAR ToolType that places the record-dragbar horizontally at the bottom of the window. Suggested by Edmund Vermeulen.
- * Added a PUBSCREEN ToolType that opens db on the specified public screen or Workbench if not found.
- * Added an ESCQUIT ToolType that enables the Escape key to quit db (in normal mode).
- * Improved the GETFIELD ARexx command to accept an optional from-field as argument.
- * Added a RETRYINPUT ARexx command that can be used in AUTORX scripts to enforce correct input. RETRYINPUT will reactivate the last field entered.
- * Added CUT COPY and PASTE Arexx commands.
- * Closing large databases is now much faster. Records are de-allocated in the reverse order from how they got loaded thus helping exec in it's job.
- * Cut can now also be used in sort and find mode.

BUGS FIXED:

- * Pasting in sort mode now works.
- * A theoretical problem with trashing of global program-variables has been fixed (has never happened though)
- * Since v2.2: ARexx scripts could be started even in sort and find mode, that "feature" has been removed.

94-09-05

v2.2b NEW CATALOG:

A German catalog added. Thanks goes to Uwe Roehm.

BUG FIXED:

- * In v2.2 Fixed problem where an ARexx program (asynchronous commands only) could suddenly halt if it sent a command to db at exactly the same time as some IDCMP event occurred (mouse click etc).

94-09-03

v2.2 NEW FEATUERS:

- * ARexx support added. ARexx programs can be invoked when the user leaves a field, doubleclicks a field or presses LAmiga+key. Example ARexx scripts are included that shows text and pictures, plays sound, dials numbers and adjust fields in different ways.
- * Added an edithook routine to make GadTool string-fields smarter (like in ASL requesters). User may now perform the following operations IN string-fields:
 - ESC deactivates (without the ugly square)
 - Up/down keys (w shift) moves between records

- Menus can be accessed with RAmiga+Key.
- Doubleclicking performs a special action
- Pressing Enter/Shift-Enter cycles around just like pressing Tab/Shift-Tab
- * Tab key now remembers the last active string-field and re-activates that one if de-activated instead of activating the first gadget.
- * Improved the parsers tolerance to "misplaced" spaces
- * db will now block input and show a waitpointer when needed (when loading, sorting..)

BUGS FIXED:

- * In v2.1: If the user made a change to a project by using cut, kill or paste and then quit. db wouldn't put up a project-not-saved warning.
- * Since v2.0: If the user tried to dial a number before any field had been selected, db would cause Enforcer read-hits.
- * Before v2.2: The Default Tool field of icons created by db wouldn't always get a correct path (to db) if db was invoked by doubleclicking on a project icon.

No Gurus so far anyway :-)

- | | | |
|----------|------|---|
| 94-08-20 | v2.1 | <ul style="list-style-type: none"> * The return and enter keys can now be used to step between fields. (Use the NORETURNSTEP ToolType to only allow the tab key for this like before v2.1) * Added a TABSIZE tag to the RFF format to simplify layout work. * The RFF parser will parse all 1.x files not just RFF 1.1 files. * Saves and loades are faster. |
| 94-08-13 | v2.0 | <ul style="list-style-type: none"> * Major changes. db is now a general database, not just a telephone and address database. db now uses the field- and layout specifications found in the datafiles (the RFF lines) * Multiple views implemented. * Added Amiga-W shortcut for "Save as..." menuitem * Commaseparated ASCII export implemented (user request) * When exporting records: filtering of records can be specified. * Custom fontname and fontsize can be specified * db now moves one titlebar down from the top left corner of the screen when zoomed. * NOICONS can be specified to suppress icon saving (user request) * before 2.0 db didn't intentionally remember the last exported filename. This has been changed. * Before v2.0: If the user made a change to a project and performed an ASCII export and then quit. db wouldn't put up a project-not-saved warning. This has been corrected. * Fixed bug where db sometimes sorted records incorrectly in projects where there were records |
-

which had blank fields which were also part of the sort order.

- 94-06-04 v1.1 * Added modem-dialing feature.
 * db now runs from Shell as well as from Workbench
 * Settings can now be made from Shell using Commodore's
 template parsing or from db's icon using ToolTypes.
 * db now moves to the top left corner of the screen
 when zoomed.
 * Gadget positions and sizes somewhat adjusted.
 * Selecting Cut won't incorrectly put up the
 delete warning requester.
 * minor code changes made.
- 94-03-04 v1.0 Initial release.
-