

# **AddressAssist**

---

Program User Guide  
for AddressAssist Release 3.0 (Version 41.1), third Edition  
January 29th 1997

by **Alex H. Schneider / Quarz Development**

---

If you fail to expect the unexpected, the system will crash!

Hope, I didn't!

Copyright © 1994-97 Quarz Development / Alex H. Schneider

This is the third edition of this documentation,  
and it is consistent with ADDRESSASSIST Release 3.0  
(Version 41.1, January 29th 1997).

Published by Quarz Development  
Alex H. Schneider  
Kruggasse 179  
CH-8439 Siglistorf, Switzerland, Europe

# 1 Introduction

ADDRESSASSIST (AA) is a database application for address management. It sports an easy to use and very nice MUI-driven graphical user interface and combines lots of extra functions not seen in other applications of this kind on the Amiga or even on other platforms.

The program is distributed as *Shareware*. See [Copyright and Registration], page 60, for details about your rights and the restriction of the freely distributable version.

## 1.1 Program features

Read the following list to get a short overview over the program's highlights:

- 16 field address database including 4 user definable text fields, a 'group'-field to distinguish up to 10 record groups and two additional multiline remark fields with unlimited field length. (Section 4.5 [Database format], page 9)
- Intuitive user interface (MUI), including a nice image-button toolbar.
- Selections of records. Selected records may be viewed, printed and exported separately. Powerful filters (*field string match* and *group match* filters) are provided. (Section 4.12 [Record selection], page 14)
- Powerful printing: output Address Lists and (multi columned) Labels Sheets with flexible text style and format. Ability to Load and Save print-setup. (Section 4.9 [Database print], page 13)
- ARexx interface. Most of the functions available through the GUI are also supported in ARexx. Freely configurable ARexx menu for easy access to AREXX scripts. (Chapter 7 [ARexx], page 32)
- Sports record links: link records to other records. (Section 4.13 [Record links], page 14)
- Easy database navigation through 'QuickAccess'-buttons on the toolbar. (Section 4.7 [Database navigation], page 12)
- Extensive use of MUI's Bubble-Help feature! (Section 4.3 [Help!], page 8)
- Record templates for simplified data input. (Section 4.14 [Record template], page 15)
- Powerful database sort feature including multiple sort indices and directions. (Section 4.8 [Database sort], page 13)
- Flexible Import/Export filters for ASCII files. Freely definable file format. Exchange data with all famous Wordprocessors, Spreadsheets and Database applications on the Amiga and all other platforms. Ability to Load and Save of ASCII-Transfer setup. (Section 4.15.2 [Import/Export ASCII], page 15)

- Data exchange through AMIGA clipboard.
- Special function for removing identical records.
- Optional: Direct exchange of address data with Casio SF series of palm-top computers using a serial link cable (available separately at the author).
- Localized under WB2.1 and above. Built-in English language. German, French, Italian and Norsk catalog translations and empty catalog description file provided. Documentation in english, german und french (release 2 only)!
- Works with WB2.04 and above, needs MUI v3.6 or above

## 1.2 Magic User Interface (MUI)

To provide maximal comfort, the user interface of this piece of software was designed to use MUI extensively. MUI is a separate product which is copyright by Stefan Stuntz. Based on famous BOOPSI (Basic Object Oriented Programming System for Intuition), it provides a great enhancement and possibility of standardisation of the Graphical User Interface (GUI) mainly for the user. More than that, from the point of view of the developers, it makes life much easier to provide a consistent, and so called style guide conform GUI for a software project. Use MUI in your own software projects to make you programs even better!!!

See 'Readme.MUI' for more information about MUI, its distribution and where to get it from.

## 1.3 Textfield gadget

AA sports two textfields with more than one lines of contents, which are implemented in separate windows. In order to use these, the **textfield.gadget** must be installed on your system. **textfield.gadget** is a separate product and is Copyright by Mark Thomas. Version 3.1 of **textfield.gadget** is supplied in the distribution archive and will be installed by the installation procedure.

See 'Textfieldgadget.readme' in the **gadgets** drawer for more information.

## 2 What is new

This chapter gives a short introduction into the new features of the latest program versions. It doesn't give any information about bug fixes or other internal stuff. See Section 9.2 [History & Future], page 59, for more information about this.

### 2.1 Release 3.0 (January 29th 1997)

#### Image-button toolbar

Enhanced GUI: the text buttons in the window main window are replaced by an all new tool bar with image-buttons, the so called Section 4.4 [Image-button toolbar], page 9.

#### Enhanced ASCII transfer

Added ability to create Merge files to export address data to easy use address data in Wordprocessors. Imported records may be assigned to groups or added to the record selection during import process. See Section 4.15.2 [Import/Export ASCII], page 15.

#### Enhanced printing

Printing of multi-columns lists with optional field titles on every page and multi-column label sheets is now supported. Ability to print headers and footers containing additional information (such as database name, date and time, number of records, page number) with freely definable text formats and styles added. Section 4.9 [Database print], page 13.

#### Online help

Push the HELP button and there is no longer a big silence when nothing happens. Section 4.3 [Help!], page 8. Furthermore, AA makes extensive use of MUI's *Bubble help* feature. Just move the mouse pointer over a button, text field or group and wait: you'll be informed, what is the use of this control.

#### Added ARexx port

The long outstanding full featured ARexx Port is finally available and provides full control over the database application to ARexx. In addition, there is an all new ARexx menu, where ARexx scripts may be executed directly from the GUI. See Chapter 7 [ARexx], page 32, for full information about this exciting enhancement.

#### Record links

Record links are shortcuts to other records, which are linked to a specific record. **Record links** introduce a new and very useful

way to structure your database. (Section 4.13 [Record links], page 14).

**Enhanced database view**

The format of the `Database listview` is now fully configurable: you may choose the record fields displayed yourself. (Section 6.7 [Preferences dialogbox], page 31)

**Enhanced sorting**

Added much faster sorting as well as the ability to sort with multiple indices and directions. (Section 4.8 [Database sort], page 13)

## 3 Installation

Installation of AA is a pretty easy task: just double-click on one of the `Install_xxx` icon on the workbench. AA will be installed using Commodore's **Installer** program. Just answer the questions mainly concerning the install path.

Some of you might want to know, what the `Install` program does. Here is a description of the files, their function and where they are copied:

Most of the files will be copied into the application's drawer.

There is a directory called `'Prefs/'` in the application drawer, where the preferences files are located by default. (`'AddressAssist.prefs'` is the main preferences file, `'*.ppref'` are specific to print preferences, `'*.xpref'` are ASCII Transfer files and `'*.scripts'` are ARexx Menu script entries).

Furthermore, there is an own 'AddressAssist' directory in the `'envarc:'` assignment, where mainly the key-file `'AddressAssist.key'` and the default data icon (`'def_AddressAssist.info'`) are located. If you move the AA's executable out of its drawer, you also have to copy the following support files to the global locations on the system:

- the contents of the `'Prefs/'` drawer to `'envarc:AddressAssist/'`
- the contents of the `'Rexx/'` drawer to `'Rexx:'`
- the `'printAssist.library'` from the `'libs/'` drawer to `'LIBS:'`
- and the `'catalog/<language>/AddressAssist.catalog'` files to `'LOCALE:'`.

To change the program's data-icon, just copy an icon to `'envarc:AddressAssist/def_AddressAssist.info'`. Note: A project icon is stored only if the appropriate switch in the Preferences dialogbox is set.

## 4 Overview

### 4.1 Program startup

To start AA, just double-click on the application's icon (or the icon of a project file) on the Workbench. From the shell, you may type

```
AddressAssist [<ProjectFile>]
```

The path of the project file is the only parameter supported when starting AA. A **Project File** contains all record data of the database as well as specific data for the database and application setup (Chapter 8 [Preferences], page 56).

If the application doesn't start, some of the needed files may not be present on your system. Please refer to Chapter 3 [Installation], page 5 in this manual in this case.

**Note:** The program stack must be set to 40000 in order to use AA without problems.

### 4.2 Quick Tour

After startup, the **main window** is opened. There you may recognize two different control groups:

- The **Image-button toolbar** at the top of the main window provides quick access to the most important functions. See Section 4.4 [Image-button toolbar], page 9, for a description of the button's use.
- The lower part of the window is covered by the **Database View Group**. There are two types of database views available: The *Database List view* and the *Record Form view*. See Section 4.6 [Database view], page 11, for more information about this important topic.

Furthermore, there is a menu strip attached to the main window, which provides access to *all* available functions in the application (Chapter 5 [Menu Reference], page 18):

- **Project Menu:** database related functions.
- **View Menu:** view related functions.
- **Record Menu:** record (say address) related functions.
- **Selection Menu:** database selection related functions.
- **Process Menu:** special functions like database import/export, sorting and the preferences editor.



- **ARexx Menu:** Configure ARexx interface and execution of ARexx scripts.

There is a short ARexx based tutorial to AA: just click on the 'ARexx demo' menu item in the **ARexx menu**.

To load a database, select **Load...** from the **Project Menu** and choose the database filename. If the program starts the first time after installation, a standard database will be loaded automatically. You may make a current database to be loaded automatically after startup by the **Save path** menu item from the **Project Menu**.

After startup, all records in the database are shown in the big list view. This is the **Database List view**. Use the **Listview format** page in the Section 6.7 [Preferences dialogbox], page 31 to change the format of the list view.

Switch to the **Record Form view** by double-click on a record in the listview or use the menu item in the **View Menu** (Section 4.6 [Database view], page 11).

Now you can watch - and edit - the full record data. To edit record data, just click in a string field, change data and press <RETURN> or <TAB> to move to the next field. Save changes to the record, using the **Save record** button or the equivalent menu item in the **Record menu**. You will be asked to save changes, if you forgot!

You may change the names of the *FreeX* fields in the Section 6.7 [Preferences dialogbox], page 31.

Check one of the 10 group checkmarks to assign the current record to this group. Alternatively you may use the **Groups dialogbox** available through the **Set/Remove groups...** menu items from the **Record Menu** to perform this task on several records. The group checkmarks are used to distinguish up to ten Section 4.11 [Record groups], page 14. The names of the groups may be changed in the Section 6.7 [Preferences dialogbox], page 31.

There are two more text fields available. Implemented in separate windows, these fields feature multi-line text. Use the menu items in the **View Menu** to open or close the field windows (Section 4.5.3 [The remark fields], page 10).

You may link other records to this current record: Think about yourself for good uses of this feature! Just select a link alias from the **Links:** popup to view the linked record. Add or change record links with the buttons in the **Links:** popup and the **Record links dialogbox**. See Section 4.13 [Record links], page 14, for more infos about this extra feature.

Switch back to the **Database List view** by using the second button from the right on the Image-button bar (beside the Help! button) or the menu item in the **View menu**.

To move to other records of the database, use the cursor keys - <SHIFTED> to move to the first or last record - or the equivalent buttons in the **Image-**

**button toolbar.** You may like to search for a record by a key. Use the **Find...** and **Find next** menu items to enter a search string and find the record.

A very useful feature of AA is the ability to define a Section 4.12 [Record selection], page 14. Selected records may be printed, exported, edited or viewed separate from the other, say unselected, records. A record may be selected just by checking the **checkbox** to the right of the *Salutation* field. In the **Database List view**, selected records are marked with a star ('\*'). You may add or remove multiple records to/from a selection by using the four buttons in the **Database List view: All, None, Toggle** current or by a *Filter*. The **Filter...** item opens the Section 6.4 [Filter dialogbox], page 28, where you may filter records to or from a selection by **text-pattern** or **group match**.

Use the **Sort...** menu item in the **Process** menu to change sort directions and sort indices of the database in memory (Section 4.8 [Database sort], page 13).

From AA you may print you record data in many different ways: there are templates available to print columnar record lists or label sheets. Most of the parameters that define the output may be varied: the most important are the selective record field print and ordering, the output text styles and formats as well as headers and footers. See Section 4.9 [Database print], page 13, for further information about this interesting and powerful feature.

AA stores databases to disk by default in so called *project files*, which are only readable by AA.

If you like to transfer data to other database programs, please take a look at the powerful Import/Export functions. AA provides a variety of possibilities to exchange data with other applications. These possibilities are included in the <undefined> [Import and Export], page <undefined> functions. You may interchange data in plain ASCII files, readable by most other database, wordprocessors and spreadsheet applications on the AMIGA and on all other platforms. The format of the transfer files may be changed to your needs using the Section 6.2 [Transfer ASCII dialogbox], page 27. You may save and reload a current exchange setup. Some predefined setups are provided.

If you ever need a hint how to solve a specific task in AA, just make use of MUT's bubble help feature or the **Help** key (Section 4.3 [Help!], page 8)

## 4.3 Help!

Help on Help! If you press the <Help> key or the equivalent button on the Section 4.4 [Image-button toolbar], page 9, AA will open this file as **Online-Help**. If the application doesn't find the helpfile ('AddressAssist .guide'), it opens a filerequester window to select the path of the helpfile.

To **change the help file path**, just press <shift> <Help>: this will open the filerequester window. The path of the helpfile is saved automatically.

Since MUI 3, AA supports **Bubble help!** Just move the mouse pointer to a control (button, text-field, list-field...) or control group, wait and you'll get infos about the purpose of this control.

## 4.4 Image-button toolbar

The buttons on the toolbar are structured in 7 groups/individuals. From the left to the right, the following functions are accessible through the buttons:

- **New**
- **Open**
- **Save**      Create a new, load and save a database (Section 5.1 [Project Menu], page 18).
- **Print**      Print database, a selection of records or only the current record (Section 4.9 [Database print], page 13).
- **New record**
- **Save record**
- **Duplicate record**
- **Delete record**
  - Create a new record, save, duplicate and remove the current record (Section 4.10 [Record edit], page 13).
- '|<'
- '<'
- '>'
- '>|'
- Move to the first, previous, next or last record (Section 4.7 [Database navigation], page 12).
- **Find...**
- **Find next**
  - Search records by key string (Section 4.7.2 [Find a record], page 12).
- **Toggle view**
  - Switch between the Section 4.6 [Database view], page 11 types: **Database list view** or **Record form view**.
- **Help!**      Get online Section 4.3 [Help!], page 8.

## 4.5 Database format

- **The record fields:** Record data fields.
- **User definable fields:** The FreeX fields.
- **The remark fields:** The Remark field windows.

### 4.5.1 The record fields

AA's database consists of 16 record fields. In the context of AA and this manual, an 'address' is referenced as *record*.

'Salutation'	<i>length: 255 chars; useful for printing address labels</i>
'Name'	
'Name2'	<i>length: 255 chars each; first and last name or company name</i>
'Phone1'	
'Phone2'	<i>length: 255 chars each; useful for phone and fax numbers</i>
'Street1'	
'Street2'	<i>length: 255 chars each; street and number or additional information</i>
'Place'	<i>length: 255 chars each; post code, city and state information</i>
'Free1'	
'Free2'	
'Free3'	
'Free4'	<i>length: 255 chars each; user definable fields, see below.</i>
'Code'	<i>length: 255 chars; useful f.e. to distinguish between several classes of addresses by using an appropriate character code for each of the different classes. Classes of addresses may be, for example, private or business addresses and hotels, restaurants or companies.</i>
'Group'	10 Checkbox buttons to distinguish of up to 10 groups of record data (Section 4.11 [Record groups], page 14).
'Remark1'	
'Remark2'	2 multi-line text fields of unlimited length for use as annotations (Section 4.5.3 [The remark fields], page 10).

### 4.5.2 User definable fields

Use the four 'FreeX' fields to your own desire by name them to your choice and then use them consequently. The field names may be changed in the Section 6.7 [Preferences dialogbox], page 31.

### 4.5.3 The remark fields

There are two 2 remark fields with unlimited text length available. They are implemented in separate windows - the Section 4.6.2 [Remark windows], page 11 - and a text entry may consist of multiple lines. You may open or close the windows using the **Remark Window x** items in the Section 5.2 [View Menu], page 19. Just type text into the text field of the window, it

will be stored to the current record data in the same way the other fields are.

Note: The multi-line remark fields are available only if the Section 1.3 [Textfield gadget], page 2 is installed on your system.

## 4.6 Database view

- **Database view types:** List and Form view
- **Remark windows:** 2 Annotation fields in separate windows
- **Selective record view:** Change the set of displayed records

### 4.6.1 Database view types

AA provides two types of database view: The **Database List view** and the **Record Form view**. While the **List view** is used for quick overview and access to all the records in the database, you may watch and edit detailed record information in the **Record Form view**.

There are several ways to switch between the two views:

- From **Database List view**, the highlighted record may be accessed directly by <Double-Click> in the list view gadget. You may also use the **Edit** menu item from the **Record** menu to perform the same action.
- Use the **View** button on the Section 4.4 [Image-button toolbar], page 9 to toggle the view type easily.
- From the **View** menu, you may also use the **List view** or **Form view** menu items or their shortcuts to switch the view type.
- Additionally, the use of the Section 4.10 [Record edit], page 13 functions will change the view to **Record Form view** automatically.

Change the format of the **Database List view** in the **Listview format** page of the Section 6.7 [Preferences dialogbox], page 31.

### 4.6.2 Remark windows

Each of the two remark windows contain a text field gadget for the remarks fields (Section 4.5.3 [The remark fields], page 10). Use **Remark window 1** and **Remark window 2** in the **View** menu to toggle availability of - open or close - the remark windows, as you need them. The availability of these windows will be stored as preferences data in the database project files.

### 4.6.3 Selective record view

In a database with a huge number of records included, one may easily fail to overview a set of records needed at the time. Use the Section 5.2 [View Menu], page 19 to change the set of records that should be displayed at a time: **View all records**, **View only selected** and **View only unselected** and **View by groups....** The later opens the Section 6.3 [Group dialogbox], page 28 and lets you select the records to be displayed by their group assignment. Use the **View groups:** checkmark bar in the **List view** for a quick access to this function. See Section 4.11 [Record groups], page 14, for more information about the use of record groups.

## 4.7 Database navigation

- **Navigation controls:** How to move through the database.
- **Find a record:** Search for records by a key string.

### 4.7.1 Navigation controls

The most important controls for navigating the database are accessible through the Section 4.4 [Image-button toolbar], page 9 and through keyboard shortcuts:

- '<' (Shortcuts: **left** or **down** arrow); moves to the previous record.
- '>' (Shortcuts: **right** or **up** arrow); moves to the next record.
- '|<' (Shortcuts: **shift left** or **shift down**); moves to the first record of the database.
- '>|' (Shortcuts: **shift right** or **shift up**); moves to the last record of the database.
- 'Find...' and 'Find next'  
(Shortcuts: **ramiga =**, **ramiga f** or **space**); moves to a record by entering a key string to find it, see Section 4.7.2 [Find a record], page 12.

From the **Database List view**, activate a record by **<double-click>** on a record in the listview to change to **Record Form view** and to edit the record's details.

### 4.7.2 Find a record

You may find a record in the database by a key string. The program compares *all* record fields of the records with the string. Input the key string through the **Find...** menu item of the **Record menu**. After accepting, the

database will be examined and the first matching record is returned and made current. The search starts at the current record. Use **Find next** to find the next record using the actual key string.

There are equivalent buttons for these functions in the Section 4.4 [Image-button toolbar], page 9.

## 4.8 Database sort

A database may be sorted in 3 levels by different *indices* and in both, **ascending** and **descending**, *directions*. The default primary sort mode is **ascending** indexed by **Name**. 2nd is **ascending** indexed by **Name2** and 3rd is **ascending** indexed by **Place**. Change the sort modes in the Section 6.5 [Sort dialogbox], page 30.

The currently active primary sort mode is displayed in the title bar of the main window.

## 4.9 Database print

AA sports a powerful print interface which enables you to print your record data in different ways.

These are

- Several Record lists, with or without field names.
- Label sheets, one or multiple columns
- Columnar Lists, with or without field names as column title

Furthermore, all page formatting (including number of record lines per page, header and footer lines), record field formatting (including field order, field names) and text style may be changed.

All parameters for text printing may be changed in the **Print dialogbox**. See Section 6.1 [Print dialogbox], page 23, for detailed information about setting up the print parameters. Print setups may be loaded & saved. This way a good setup may be reused later again.

Please, try out the provided example setups to learn about the power of the printer interface.

## 4.10 Record edit

The Record edit functions include all available commands for working on and with individual records. All of them are available through the **Record menu**. Some have an equivalent button on the toolbar. See Section 5.3 [Record Menu], page 19, for detailed infos.

## 4.11 Record groups

Records may be assigned to one or several groups. In AA, up to 10 record groups are distinguished. This adds structural information to your database. Record groups may be selected and shown separate from the other records. There are two methods to change the group assignment of the records. Use the checkboxes in the **Record form view**, if you like to change only the current record. To modify several records, say assign a set of group to these records, use the Section 6.3 [Group dialogbox], page 28 to simplify this process.

## 4.12 Record selection

A special feature of AA is the capability to create a selection of records. A selection of records may be treated (print, export, assign groups) separate to all other records of the database. Whether a record is selected can be seen from the star \* preceding each record in the listview gadget of the **Database List view** or by the checked checkmark gadget on the top right of the **Record Form view**.

All selection related functions are collected in the Section 5.4 [Selection Menu], page 20. Use the 4-button bar below the listview gadget in the **Database List view** to edit record selections.

## 4.13 Record links

Record links are an all new feature brought to database applications on the AMIGA: A record link is a shortcut (or an alias) to another record, which is linked to a specific record. As many such record aliases may be added to one record, **Record links** introduce a new and very useful way to structure your database. Think about yourself for applications for this interesting new feature; there are plenty of them.

Select a **Record link** entry from the listview on the **Links:** popup to display linked record data. Switch back to **<Original>** to view the original record.

Note: Record data displayed as a link may not be edited for now: it is read only! This may change in the future.

Record links are controled from the **Links:** popup on the formview page:

Use the **Add** button and then select and name a record in the **Links dialogbox** to create a new record link.

Use the **Edit** button to change the highlighted record link.



Use the **Remove** button to remove the highlighted record link.

## 4.14 Record template

The record template function gives you the possibility to easily add new records, which have some data in some record fields common. F.e. if you have a couple of addresses in the same town, country and of the same type, you can create a record Template with the common data, and then use the template when you create new records. The template includes all data of the record text fields, the group field and the selection field. Data of the remark fields are not saved in the template.

Use the **Use Template** and the **Save current as Template** menu items from the **Record** menu together with the **New record** function, if you like to apply templates. See [\[Record menu\]](#), page [\[Record menu\]](#), for detailed information.

## 4.15 Import and export

- **Import/Export selection:** Transfer record selections .
- **Import/Export ASCII:** Transfer data with other applications.
- **Import/Export Casio:** Transfer data with Casio SF xx00 palmtops.

### 4.15.1 Import/Export selection

To use only a part of the current database in a second stand-alone database, export a defined selection by using the **Export...** function from the **SELECTION MENU**. In the same way, other AA database files may be imported. (**Import...** function from **SELECTION MENU**). When importing from a database file, only the record data will be read from the file; preferences data of the current database in AA will left untouched.

Compared to ASCII Transfer, the main difference when importing a previously exported selection is that the structural data, such as record groups and record links will be preserved.

### 4.15.2 Import/Export ASCII

To transfer data of an AA database to or from other databases or spreadsheet application on the AMIGA (f.e. Superbase, MaxonTwist, FinalData, FinalCalc) or on other platforms (f.e. MS Access, DBase, Filemaker or MS Excel), you may do this by using ASCII files.

Wordprocessors, such as Wordworth and FinalWriter/Copy on the Amiga and MS Word or Wordperfect support so called **Merge files** to create f.e. serial letters, which may be created by AA.

In AA, ASCII files may be read or written through the **Transfer ASCII dialogbox**. In this dialog, there are all controls for defining the format of the ASCII data files. A format definition may be saved to disk and restored for later use. See Section 6.2 [Transfer ASCII dialogbox], page 27 and Section 4.15.2.1 [ASCII file format], page 16, for detailed information about transfer options.

#### 4.15.2.1 ASCII file format

The use of ASCII files is not standardized. However they provide a very flexible way of exchanging data with many other applications. In AA you have all possibilities to define the format of ASCII files in a way that reading or writing of most types of exchange files is easily possible.

In general, the format of ASCII exchange files is as follows. The end of each record is marked by a special char, the so called **Record separation char**. The fields within a record are separated by another specific char, which is called the **Datafield separation char**. In AA, the default chars are 'LF' (ASCII 10) for record separation and ',' (ASCII 44) for field separation:

```
FieldA_Data1<FS>FieldB_Data1<FS>FieldC_Data1....<FS>FieldM_Data1<RS>
FieldA_Data2<FS>FieldB_Data2<FS>FieldC_Data2....<FS>FieldM_Data1<RS>
.
.
.
FieldA_DataN<FS>FieldN_Data1<FS>FieldB_DataN....<RS>
```

where <FS> and <RS> are the **Field separation** and **Record separation** chars. M is the number of record fields and N is the number of records.

For **Merge files**, the format is only slightly different. The first line is reserved for record field *names* instead of record field *data*:

```
FieldA_Name<FS>FieldB_Name<FS>FieldC_Name.....<FS>FieldM_Name<RS>
FieldA_Data2<FS>FieldB_Data2<FS>FieldC_Data2....<FS>FieldM_Data1<RS>
.
.
.
FieldA_DataN<FS>FieldN_Data1<FS>FieldB_DataN....<RS>
```

Most important, in AA it is possible to define the record field ordering easily yourself from within the GUI. This adds enormous flexibility on one hand, and comfort in use on the other hand, to the ASCII transfer feature.

If the record entries in imported ASCII file contain less record fields than AA provides, the missing fields will left blank. On the other hand, if the imported ASCII datafile contains more record fields per record, the surplus fields will be ignored.

### 4.15.3 Import/Export Casio

AA supports direct data exchange to Casio FS type palmtop computers (B.O.S.S. protocol). As not everybody needs this feature, it's distributed separate to the database application in an extra product: The **Casio Exchange** package consists of the software, which is an external module to this database application, and (optional) the hardware: a serial link cable.

The comfortable and easy to use software allows full control of data exchange parameters through a MUI based GUI. It supports direct upload and download to/from the Casio with similar data control features as available for ASCII transfer.

If you are interested, contact the author for more information and pricing. (Chapter 13 [Contacting the author], page 63)

## 5 Menu Reference

- **Project Menu:** Database related functions.
- **View Menu:** View related functions.
- **Record Menu:** All record related functions.
- **Selection Menu:** Record selection related functions.
- **Process Menu:** Database import/export, sort and preferences editor.
- **ARexx Menu:** ARexx related commands.

### 5.1 Project Menu

**About** shows the program's **Copyright** dialogbox with information about the current program version and revision as well as registration status ([Copyright and Registration], page 60).

**New** clears the current database in memory and you may begin to work on a new database. You will be prompted to save changes to the old database if necessary.

**Load...** opens the filerequester window which prompts for the filename of the address database to be loaded. Then the database will be read from disk if possible. If there is already a database in memory, it will be entirely cleared. You will be prompted to save changes to it if necessary. (**ramiga o**)

**Save...** writes the current database to disk with its associated filename. If there is no associated filename then you will be prompted to input a filename by the filerequester window. (Shortcut: **ramiga s**)

**Save as...** opens the file requester window which prompts for the filename of the address database to be saved. (Shortcut: **ramiga a**)

**Print...** opens the **Print** dialogbox, where you set all print related parameters and print the database or only parts of it. (Section 6.1 [Print dialogbox], page 23). (Shortcut: **ramiga p**)

**Save path** simply saves the path of the current database. After startup, AA will load this database automatically.

**Quit** just does what it says: Quit the program, but first ask to save changes to the database if appropriate. (Shortcut: **ramiga q**)

## 5.2 View Menu

The **List view** and **Form view** items may be used to switch database view (Section 4.6 [Database view], page 11).

Use **Remark window 1** and **Remark window 2** to open or close the separate field windows (Section 4.6.2 [Remark windows], page 11).

**View all records** displays all records currently available in the database.

**View only selected** / **View only unselected** display only records, which are currently selected / not selected (Section 4.6.3 [Selective record view], page 12).

**View by groups...** opens the Section 6.3 [Group dialogbox], page 28 and lets you select the records, which should be displayed, by their group assignment. Use the **View groups** button-bar in the **List view** for a quick access to this function (Section 4.11 [Record groups], page 14).

## 5.3 Record Menu

**New** cleans the display of the current record. If appropriate, you will be prompted to save changes to the cleaned record, Then the first string gadget is activated for input of new record data. Check the **Record template** item to fill the fields with the data of the Record template (Section 4.14 [Record template], page 15) (Shortcuts: **n** or **ramiga n**)

**Save** saves changes to the current record or adds a new record to the database. You will be prompted automatically for the save function, either when leaving the last datafield string gadget ('Code') by a **return** or by selecting a function which would cause the loss of recently changed data. (Shortcut: **s**)

**Edit** activates the first datafield of the current record ('Salutation'), which is the same action as when clicking with the mouse on this gadget or another of **current record edit group's** string gadgets. This is useful, if the program is used only by keyboard. (Shortcut: **e** or **ramiga e**)

**Duplicate** adds a new record to the database using the data of the current record. You will be prompted by a requester to confirm this action. (Shortcut: **d** or **ramiga d**)

**Delete** removes the current record from memory. You will be prompted by a requester to confirm this unrecoverable action. (Shortcut: **l** or **ramiga l**)

**Copy** copies the text data of the current record to the AMIGA's clipboard for exchange with other applications supporting the clipboard. (Shortcut: **ramiga c**)

Check the **Use Template** menu item to fill the record fields of a new record with the record template data, instead of just clearing the form view Section 4.14 [Record template], page 15.

Use the **Save current as Template** to save the current record as a record template. The data of the template will be stored in the database file and is still available after a restart or reload of the database.

**Set/remove group...** opens the Section 6.3 [Group dialogbox], page 28 window, with a cycle gadget and a set of group checkboxes. The **Apply to...** cycle gadget lets you change the set of records (**all** or **only selected records**, **current record**), which you would like to assign or remove the assignment from specific groups. Use the group checkbox gadgets to indicate which groups have to be assigned to. There are two methods to change the group assignment of the records. Use the 10 group checkboxes in the **Record form view**, if you like to edit only the current record. If you like to change similar groups assignments of several records, AA provides this function to simplify this process:

**Find...** prompts the user to input a key string. After accepting by clicking OK, the database will be examined for the given key and the first matching record is made the current one. The search begins with the first record. (Shortcuts: **=** or **ramiga =**)

Starting from the current record, **Find next** finds the next record in the database that matches the actual key. If there is no actual key, this command acts like **Find...**

## 5.4 Selection Menu

**Add all/All** selects all records of the database. (Shortcuts: **a** or **ramiga a**)

**Remove all/None** deselects all records of the database. (Shortcuts: `n` or `ramiga n`)

**Toggle current/Toggle** changes the selection state of the current record. (Shortcuts: `t` or `ramiga t`)

**Filter.../Filter** opens the `Filter` dialogbox. (Shortcuts: `?` or `ramiga ?`)

**Import/Export...** provides access to the selection import/export functions. See also Section 4.15.1 [Import/Export selection], page 15.

**Print...** opens the Section 6.1 [Print dialogbox], page 23, with the **Selection** cycle gadget (`Page setup`) set to **previously selected**.

## 5.5 Process Menu

**Import ASCII...** opens the `filerequester` window which prompts for the filename of the ASCII data file to be imported. All records will added to the current database. Identical records may be deleted later by a separate function (`Remove redundant records...` from `MENU PROCESS`), if desirable.

**Export ASCII...** first opens a dialogbox where you can specify whether to export all records or only previously selected records (Section 4.12 [Record selection], page 14) from the current database. Then you will be prompted to input the filename for the ASCII export data file.

Use **Transmit to Casio...** and **Receive from Casio...** to access the optional available *Casio Exchange* package, which provides direct reads and writes to Casio SF palmtops through a serial link cable. Contact the author, for further information about this interesting topic.

**Sort database...** opens the `Sort dialogbox`, where you can *choose* the `sort direction` and *select* the appropriate `index`. The current sort mode is displayed in the title bar of the main window (Section 6.5 [Sort dialogbox], page 30). (Shortcut: `ramiga r`)

**Remove redundant records...** examines the database for identical records and prompts whether or not to delete found identical records. Specify **All** to remove all record in one step without being prompted anymore. There will be a message at the end of the operation confirming, that there are no more redundant records in the database memory.

**Change Preferences...** opens the **Preferences dialogbox**, where you can set program and database related preferences (Section 6.7 [Preferences dialogbox], page 31).

Use **Change MUI Preferences...** to start the MUI preferences editor and change the GUI parameters.

## 5.6 ARexx Menu

**Execute ARexx script...** opens a dialogbox, where the user is prompted for the path of the ARexx script to be executed.

**Set ARexx console...** opens a dialogbox, where the console definition string for the ARexx console may be changed. Use standard AMIGA-DOS keywords for the definition string as used for all AMIGA-DOS shells. See **newshell** command documentation.

**Set ARexx scripts...** opens the Section 6.6 [ARexx dialogbox], page 30, where the ARexx script menu items of the **ARexx menu** are setup.

There are up to 20 **ARexx script entries** available in this menu. The entries are dynamically added and removed following the setup in the Section 6.6 [ARexx dialogbox], page 30. Use the appropriate menu item to execute an ARexx script.



## 6 Dialogbox Reference

- **Print dialogbox:** Printer and print interface setup.
- **Transfer ASCII dialogbox:** Import/export of ASCII text files.
- **Group dialogbox:** Record group manipulation.
- **Filter dialogbox:** Record selection filters.
- **Sort dialogbox:** Database sort options.
- **ARexx dialogbox:** Manage ARexx scripts.
- **Preferences dialogbox:** Program and database preferences editor.

### 6.1 Print dialogbox

The Print dialogbox is accessible through

- the **Print...** menu item in the **Project** menu (Shortcut: `ramiga p`),
- the **Print...** menu item in the **Record** menu,
- the **Print...** menu item in the **Selection** menu,
- and the **Print** image-button on the **Toolbar**.

AA provides a powerful print interface which enables you to print your records with many different ways. Using so called print **templates** formats, printing of column-titled lists or multi-column label sheets is an easy task.

These are, how many records or record lines in columnar mode should be printed on one sheet of paper (very useful for label printing) or whether or not to print the record field names. For each datafield, you furthermore may specify separately whether or not it should be printed at all, which style should be used for printing and how many linefeeds should be printed after each field.

Most of the print defining parameters may be changed by the user in this dialogbox. A specific print setup may be saved and reused later.

This dialogbox consists of four pages, which are described separately:

- **Page setup page:** Print format and templates.
- **Field options page:** Set record field order, style and format.
- **Header & Footer page:** Set Header and Footer format and styles.
- **Printer setup page:** Output device: printer or file; paper length and margins.

To restore standard values for preferences data in this dialogbox, AA offers both, *global dialogbox* (the **Default all** button) and *dialog page oriented* (**Default** reset of preferences data..

Use 'OK' to print with the specified setup. The 'Save' stores recently changes to the print parameters without printing. Leave the dialogbox forgetting about changes with 'Cancel'.

### 6.1.1 Page setup

**Selection** sets the set of records to be printed: **all records**, **selected records only** or **only current record**. (Section 4.12 [Record selection], page 14).

**LF before address** and **LF after address** indicate the number of linefeeds printed in front of / after a record.

**Records per page** controls the number of record, printed on one page ( <= 0: limited by page size). This makes it very easy to take care of not to split a record on two pages. Use this parameter together with **Page Length** from the **Print setup** page. Set to a 0 or <0 to prevent checking the number of addresses printed on one sheet of paper.

**Template** lets you select a predefined print template. Currently there are two templates types available: **Address list template** and **Label sheet template**. If your desired print format is not supported by these templates, use <No Template>. If you have a good idea about a new template, please drop a mail to the Author (Chapter 13 [Contacting the author], page 63).

**Number of columns** sets the number of print columns. This feature is used f.e. by the **Label sheet template** to define the number of labels printed across the page. It may also be used together with the <No Template>.

Use **Columns space** to set the space between the columns.

**Columns width** is calculated automatically from the **Number of columns** and **Columns space** variable as well as the **Page Width**, **Left and Right Margin** parameters (Section 6.1.4 [Printer setup page], page 26).

Check **Print header** or **Print footer** to print header or footer lines on each page. Use the **Number of lines** control to set the number of lines reserved for the header and the footer. In fact, this is a sort of spacing between the header / footer texts and the body of the page as the header and footer are printed on *one* line only.

### 6.1.2 Field options page

On the top of this page there is a **listview** with all record fields listed in their current print order. In addition the information about the print format and style for each field is displayed.

There are 6 controls available to change the print format and style for each field separately:

The '**Up**' and '**Down**' button change the print order of the record fields.

The **Checkbox** indicates whether or not this field is printed. **Double-click** with the mouse on the field in the listview to perform the same action.

The **Style Cycle** changes the print style of the record field. Choose between **Normal**, **Bold**, **Italic** or **Bold-Italic**.

The **Linefeed Cycle** changes the number of linefeeds printed after each record field. Select from **no LF** to up to **3 LF**.

If **no LF** is active, then the **Field length** text field is used to determine the width of each column. This is used for columns list printing. A value of smaller or equal zero (  $\leq 1$  ) adds only a single **<space>** at the end of the data field text and then continues with next field on the same line.

The controls in the **Freeformat fields group** may be used to set the parameters concerning Section 4.5.3 [The remark fields], page 10: Use the **Checkmark** to indicate whether the field should be printed at all. The **Number of lines** defines how many lines are used to print the field text. If the field data is longer than the indicated number of lines, printed text will be truncated to the given length.

The 4 controls of the **Fieldnames** group are equal to those of the record fields in the listview above.

### 6.1.3 Header & Footer page

On this page the printing of the header and a footer lines are controled. This page for headers and footers contains two control groups: one for headers one for footer. These control groups themselves are separated into three parts to input the left-aligned, centered and the right aligned parts of the header/footer line separately. The three input controls are as follows:

The **Text input field** contains the text with control codes to add filename and database information, date or time and page numbers to the header/footer line. See Section 6.1.3.1 [Header & Footer control chars], page 26 about the use of control chars.

Use the **Style** cycle field change text style: Plain, **Bold** or *Italic*.

The **Text preview** field shows the resulting text without control chars in current text style.

### 6.1.3.1 Header & Footer control chars

%F 0 (default) insert filename  
1 insert full path and filename

%T 0 (default) insert print time (hh:mm)  
1 insert print time (hh:mm:ss)

%P - insert page number

%D 0 (default) insert date (ddd mmm dd)  
1 insert date (dd.mm.yy)  
2 insert date (mm/dd/yy)  
3 insert date (dddd, dd.mmmm yyyy)

%N 0 (default) insert total number of records  
1 insert selected number of records

### 6.1.4 Printer setup page

Change printer specific setup on this page.

The **Printer** text field shows the currently selected AMIGA preferences printer driver. Change the printer driver in the **Printer preferences** program of your workbench distribution.

**Page Length** indicates the height of the sheet of paper (in number of lines) including Top and Bottom margin.

**Page Width** indicates the width of the paper sheet (in number of chars) including Left and Right margins.

Toggle **Line Spacing** between **six** or **eight lines per inch**.

**Top Margin** is counted in number of empty lines on the top of the paper.

**Bottom Margin** is counted in number of empty lines on the bottom of the paper.

**Left Margin** indicates the left margin on the paper in number of characters.

**Right Margin** indicates the right margin on the paper in characters, which is **counted from the right side**.

**Formfeed print** provides the possibility to choose, whether or not to send a Formfeed (FF) to the printer after each printed page. Use this switch if you have problems like always printing an empty page after f.e. label or list print.

**Output to** toggles current print output device: **Printer (PRT:)** or **text File**.

## 6.2 Transfer ASCII dialogbox

The Transfer ASCII dialogbox is accessible through

- the **Export ASCII...** menu item in the **Process** menu.
- and the **Import ASCII...** menu item in the **Process** menu.

The body of the dialogbox is available in both export and import mode and is used to define the file format of the ASCII transfer file. See also Section 4.15.2.1 [ASCII file format], page 16.

The **Field order** group consist of a listview with the record fields listed, and two buttons to manipulate the order of the record fields: Use *Up* or *Down* to move the highlighted record field.

The **Separation chars** may indicated separately for records (**Record separation**) and for record fields (**Field separation**). See Section 4.15.2 [Import/Export ASCII], page 15, for detailed information about the format of the ASCII transfer files.

Check **Export field names** to add the names of the record fields at the top of the transfer file: The so called **Merge files** are often used to exchange data with Wordprocessors.

The **Preferences** group is used to **Save** current file format to disk, **Load** a previously stored format or return to **Default** setup.

The additional contols depend on the mode of the dialog:

**Export mode:**

The **Apply to...** cycle at the top of the window defines the set of records to export: *all records*, *only selected records* or *only current record*.

**Import mode:**

Check the **Selected** checkmark to add imported records to the selection.

Check the **Groups** checkmark to assign imported records to a set of groups. Indicate the set of groups in the **Groups dialogbox** appearing after pushing the **Groups** button.

### 6.3 Group dialogbox

The Group dialogbox is accessible through

- the **Set groups...** menu item in the Record menu,
- the **Remove groups...** menu item in the Record menu,
- and the **View by groups...** item the View menu,
- the **Import...** item the Selection menu,
- and the **Import ASCII dialogbox**.

The Group dialogbox is used in two ways.

If invoked from the Record menu, you may **add** or **remove** the group assignments of one or several records. On the other hand the dialogbox is used to change to set of groups for the Section 4.6.3 [Selective record view], page 12, the selection and the ASCII import.

Use the **Apply to...** cycle at the top of the window to define to which selection of records the group assignments should be changed: *all*, *only selected* or *only current record(s)*.

The tem **Group checkmarks** are used to define the set of groups the action should be applied to or that should be viewed.

The name of the groups may be changed in the Section 6.7 [Preferences dialogbox], page 31.

### 6.4 Filter dialogbox

The Filter dialogbox is accessible through

- the **Filter...** menu item in the Selection menu (Shortcut: ramiga ?),
- and the **Filter** button below the record list in the Database view (Shortcut: F).

The **Filter dialogbox** is used to include or exclude record to or from a selection. It consists of two main controls:

The **Action** cycle at the top of the window lets you choose, whether to include, exclude or toggle matching records to or from the selection.

Below is the **Match** page group, which provides access to the controls of the two types of record match available:

- **Field string match:** String match of the record fields.
- **Group match:** Record match to groups.

### 6.4.1 Field string match

The **Field string match** group consist of a number checkboxes - one for each record field - and the actual Match string gadget at the bottom of the group for input of the string to match one or several record string fields. Use the checkboxes to indicate whether or not the match operation should be applied to a record field or not.

A match would mean that the entire string **TEXT** is used up in matching the **PATTERN** and conversely the matched **TEXT** uses up the entire **PATTERN**.

The match function supports the following formats in the match string:

- '\*' matches any sequence of characters (zero or more)
- '?' matches any character
- '\' suppresses syntactic significance of a special character
- [SET] matches any character in the specified set,
- [!SET] or [^SET] matches any character not in the specified set.

A set is composed of characters or ranges; a range looks like 'character hyphen character' (as in 0-9 or A-Z). [0-9a-zA-Z\_] is the minimal set of characters allowed in the [...] pattern construct.

To suppress the special syntactic significance of any of '[\*?!^-\]', and match the character exactly, precede it with a '\'

### 6.4.2 Group match

Use the **Group checkboxes** of the **Group match** controls to indicate whether a record should match a group or not. A record matches, if it is assigned to a group which is checked in this control group.

## 6.5 Sort dialogbox

The Sort dialogbox is accessible through

- the **Sort...** menu item in the **Process** menu (Shortcut: **ramiga r**),

In this dialogbox, the three sort modes for the current database may be changed. The dialogbox consists of three controls:

Use the **Sort mode** cycle to select which sort mode should be changed. There are three sort modes available.

Use the **Sort direction** cycle to select, whether the current database should be sorted in **ascending** and **descending** direction.

From the **Index** list, you may select the appropriate **sort index**.

By pushing the '**OK**' button, the database will be resorted in the desired way. Use **Default** to return to the predefined values: *ascending* direction by index *Name* for the first sort mode, *Name 2* and *Place* as second and third modes, each of which in ascending direction. The first sort mode is displayed in the title bar of the main window.

## 6.6 ARexx dialogbox

The ARexx dialogbox is accessible through

- the **Set ARexx scripts...** menu item in the **ARexx** menu.

The ARexx dialogbox is used to manage the list of ARexx scripts, executable directly from the **ARexx menu**.

In the **Available scripts** listview, the ARexx script entries that are added to the **ARexx menu** are listed. An ARexx script entry consists of an *Alias name* and the *Path name* of the ARexx script. It is the *Alias*, which is displayed as a menu item in the **ARexx menu**: click on appropriate menu item to execute corresponding an ARexx script.

Use **Drag & drop** to sort the entries in the list. The maximum number of entries is 20.

The **Add** and **Edit** buttons below the listview are used to create / change an *ARexx script entry* in a separate dialogbox. Use **Remove** to delete the highlighted or all *ARexx script entries* from the list.

**Save as** and **Load** are used to store and restore a specific list of *ARexx script entries*. Use **Append** to append a stored list to the current one.

To close dialogbox, use



**Save** to write current setup to application preferences (Chapter 8 [Preferences], page 56).

**Use** to accept current setup, but do not store it for later use.

**Cancel** to forget changes.

## 6.7 Preferences dialogbox

The Preferences dialogbox is accessible through

- the **Change preferences...** menu item in the **Process** menu.

The preferences items are splitted into four page groups of the Preferences dialogbox:

- in the **Free field names** page, the names if the four user definable (**FreeX**) record fields may be changed.
- Go to the **Group definition** page to change the names of the 10 groups using the textfields on this page.
- The **Listview format** page consists mainly of two listview field. **Drag & Drop** the fields you like to view from the **Available fields** list to the **Visible fields** list. Sort the visible fields by **drag & Drop**. In addition, information concerning size of the field and column formatting is provided in this list.

Use the **Size** slider to change the column size in %. Check the **Use Tab** checkmark to create a new column for the highlighted field. Otherwise this field's data is displayed together with the previous field.

Check the **Draw vertical border** checkmark to separate the columns in the listview by lines.

- Use the **Application** page to change program specific preferences: decide to save project files with an icon or have a 'Save reminding' requester popped up when a record is changed.

To restore standard values for preferences data in this dialogbox, AA offers both, *global dialogbox* (the **Default all** button) and *dialog page oriented* (**Default** reset of preferences data).

Use **Save** to write setup to application preferences: the set of preferences data is used after startup and in new database (Chapter 8 [Preferences], page 56).

## 7 ARexx

- **ARexx Introduction:** Using ARexx with AA.
- **ARexx commands:** ARexx command reference.

### 7.1 ARexx Introduction

ARexx is a powerful script language widely spread on the Amiga computer. Many applications sport a so called *ARexx interface*. ARexx provides a standard to externally control applications and exchange data with these applications.

In AA, nearly every command accessible in the GUI is also available in ARexx scripts. And there are even some more.

All ARexx interface specific commands in AA are collected in the ARexx menu. Section 5.6 [ARexx Menu], page 22 in the Menu reference for more details. This way AA sports the commodity to start ARexx scripts from a menu: You may put favorite scripts into the **ARexx menu**, where they may easily be executed.

### 7.2 ARexx commands

#### 7.2.1 Command argument standards

Each command template is given in the standard DOS ReadArg form. That is each argument name is given separated by commas with flags specified (separated by a /). Within an 'ARexx' script command arguments should be separated by space (not commas). The flag definitions are as follows:

Flag	Definition
------	------------

/K	Keyword required.
----	-------------------

/A	Argument required.
----	--------------------

/F	Final argument. The remainder of the line will be set to this argument.
----	---

/M	Multiple arguments (separated by space).
----	--

/S	Switch argument.
----	------------------

Note that some string arguments shown in the examples are surrounded by two sets of quotes (consisting of a pair of single quotes and a pair of double quotes). The general rule of thumb is if the argument is the final argument (indicated by a /F in the command template) then only a single pair of quotes is necessary. Otherwise, if the string contains spaces then the two sets of quotes are necessary. The reason for this is that 'ARexx'

tries to interpret every token it encounters. If the token is surrounded by quotes then the token is interpreted by removing the quotes and leaving the enclosed string intact. Likewise, the DOS ReadArgs function (used by 'MUI' to parse incoming 'ARexx' command lines) parses arguments separated by spaces, therefore, strings with spaces must be enclosed by quotes (hence, the need for the two sets of quotes). An exception is if the argument is designated as the final argument in which case the ReadArgs function will set the remainder of the command line as the final argument.

## 7.2.2 MUI standard commands

- **quit** FORCE/S

This command will end AA, closing windows and freeing all associated memory. Note that if a script fails the application may still actually be running. Use this command to end the process by using an inline 'ARexx' command (e.g. issue `rx "address [portname] quit"` from a shell).

- **hide** Hides (iconifies) the application.

- **show** Shows (pops up) an iconified application.

- **info** ITEM/A

According to the given parameter the result string is filled with the following contents (or something reasonably close):

`'title':` AddressAssist

`'author':` Alex H. Schneider

`'copyright':`

Copyright 1994-1996, Quarz Development

`'description':`

Database application.

`'version':` Current version string.

`'base':` Name of the ARexx port

`'screen':` Name of the public screen

- **help** FILE/A

A list of all 'ARexx' commands available for the application is written into the given file. In addition to the default commands an MUI application can (and of course should) support many application specific commands. The help list will contain these commands as well.

## 7.2.3 Application specific commands

### 7.2.3.1 newdatabase

**SYNTAX** NewDatabase "Name"

**FUNCTION**

This function creates a new database named **Name** and removes the current database from memory.

**PARAMETERS**

*Name:* Name of database

**RETURN** always RC = 0

with RESULT = <name of database>

**SEE ALSO**

Section 7.2.3.2 [loaddatabase], page 34, Section 7.2.3.3 [save-database], page 34

### 7.2.3.2 loaddatabase

**SYNTAX** LoadDatabase "Filename"

**FUNCTION**

Reads a database from disk after removing the current database from memory.

**PARAMETERS**

*Filename:* Name of Database file to load. Getfilename dialog will open, if not specified.

**RETURN** RC = 0, if successful; RC > 0, if failed to load database. with RESULT = <name of database>

**SEE ALSO**

Section 7.2.3.2 [loaddatabase], page 34, Section 7.2.3.3 [save-database], page 34

### 7.2.3.3 savedatabase

**SYNTAX** SaveDatabase "Name"

**FUNCTION**

Writes current database to disk.

**PARAMETERS**

*Filename:* Name of Database file. If not supplied, this function uses default filename specified in **newdatabase**, else opens the Getfilename dialog to request filename from user.

**RETURN** RC = 0, if successful; RC > 0, if database not saved. with  
RESULT = <name of database>

**SEE ALSO**

Section 7.2.3.2 [loaddatabase], page 34, Section 7.2.3.1 [new-database], page 34

### 7.2.3.4 print

**NAME** Print

**SYNTAX** Print 'S=SelectedOnly/S,C=CurrentOnly/S,D=Dialog/S,P=PrefsFilename/K,OutFilename'

**FUNCTION**

Print a set of record. You may use the previously saved print preferences for printing.

**PARAMETERS**

*SelectedOnly:*

Only selected record will be printed.

*CurrentOnly:*

Only the current record will be printed.

*Dialog:* Open Print dialogbox.

*PrefsFilename:*

Name of print preferences to use for printing.

*OutFilename:*

Name of output file, supply only when printing to a file.

**RETURN** always RC = 0

**SEE ALSO**

Section 6.1 [Print dialogbox], page 23

### 7.2.3.5 newrecord

**SYNTAX** NewRecord T=UseTemplate/S,NameFieldData/F/A

**FUNCTION**

This function creates a new record, which is set as the current one. You may and have to specify only the **Name** data of the record. Use **setrecorddata** function to set the new record's other fields.

**PARAMETERS**

*UseTemplate:*

Switch to create record using Template data instead of empty record.

*NameFieldData:*

Record name entry.

**RETURN** always RC = 0

**SEE ALSO**

Section 7.2.3.7 [setrecorddata], page 36

**7.2.3.6 getrecorddata**

**SYNTAX** GetRecordData FieldID/N/A,R=Record/K/N,S=Selected/S

**FUNCTION**

This is the main function retrieving record data from a specific record entry. Use the numeric argument `FieldID` to access desired record field. memory.

**PARAMETERS**

*FieldID:* Record field id, ranging from 0 = Salutation to 14 = Remark 2. See above and example scripts for details.

*Record:* Number of record, counted from 0 for the first record in the current sort mode. If `Selected` is switched, only the selected records are counted. To sepcify the current record, do not supply this keyword.

*Selected:* See `Record`.

**RETURN** RESULT = <record field data>, if success (RC = 0)  
or RESULT = `result`, RC > 0

**SEE ALSO**

Section 7.2.3.7 [setrecorddata], page 36, Section 7.2.3.15 [format-recordstring], page 41

**7.2.3.7 setrecorddata**

**SYNTAX** SetRecordData FieldID/N/A,R=Record/K/N,S=Selected/S,Text/F

**FUNCTION**

This is the function for changing record data of a specific record entry. Use the numeric argument `FieldID` to access desired record field. memory.

**PARAMETERS**

- FieldID:* Record field id, ranging from 0 = Salutation to 14 = Remark 2. See above and example scripts for details.
- Record:* Number of record, counted from 0 for the first record in the current sort mode. If **Selected** is switched, only the selected records are counted. To sepcify the current record, do not supply this keyword.
- Selected:* See **Record**.
- Text:* Text data to assign to the record field. Supply line-feeds (LF) only in the Remark fields.

**RETURN** if success (RC = 0), else RC > 0

**SEE ALSO**

Section 7.2.3.6 [getrecorddata], page 36

### 7.2.3.8 duplicaterecord

**SYNTAX** DuplicateRecord Record/N,S=Selected/S

**FUNCTION**

This function duplicates the record specified by the numeric argument **Record** and the **Selected** switch. Give no arguments to duplicate current record.

**PARAMETERS**

- Record:* Number of the record; counted from 0, which is the first record, using the current sort mode. If **Selected** is switched, only the selected records are counted.
- Selected:* See **Record**.

**RETURN** if success (RC = 0), else RC > 0

**SEE ALSO**

Section 5.3 [Record Menu], page 19

### 7.2.3.9 removerecord

**SYNTAX** RemoveRecord Record/N,S=Selected/S

**FUNCTION**

This function removes the record specified by the numeric argument **Record** and the **Selected** switch from database memory. Give no arguments to remove current record. The record will be removed immediately without any confirmation dialogbox!

**PARAMETERS**

*Record:* Number of the record; counted from 0, which is the first record, using the current sort mode. If **Selected** is switched, only the selected records are counted.

*Selected:* See **Record**.

**RETURN** if success (RC = 0), else RC > 0

**SEE ALSO**

Section 5.3 [Record Menu], page 19

**7.2.3.10 savetemplate**

**SYNTAX** SaveTemplate Record/N,S=Selected/S

**FUNCTION**

This function saves the specified record as **Record template**. Record templates are useful, when creating new records which a similar data. Give no arguments to save current record, or use the numeric argument **Record** and the **Selected** switch to specify the desired record.

**PARAMETERS**

*Record:* Number of the record; counted from 0, which is the first record, using the current sort mode. If **Selected** is switched, only the selected records are counted.

*Selected:* See **Record**.

**RETURN** if success (RC = 0), or RC > 0 if no record found.

**SEE ALSO**

Section 4.14 [Record template], page 15

**7.2.3.11 copytoclip**

**SYNTAX** CopyToClip Record/N,S=Selected/S

**FUNCTION**

This function copies the data of the specified record to the Amiga clipboard. Give no arguments to save current record, or use the numeric argument **Record** and the **Selected** switch to specify the desired record.

**PARAMETERS**



*Record:* Number of the record; counted from 0, which is the first record, using the current sort mode. If **Selected** is switched, only the selected records are counted.

*Selected:* See **Record**.

**RETURN** if success (RC = 0), or RC > 0 if no record found.

**SEE ALSO**

Section 5.3 [Record Menu], page 19

### 7.2.3.12 getrecordgroups

**SYNTAX** GetRecordGroups R=Record/K/N,S=Selected/S

**FUNCTION**

Use this function to retrieve record group data from a specific record entry. Give no arguments to retrieve data from current record, or use the numeric argument **Record** and the **Selected** switch to specify the desired record.

**PARAMETERS**

*Record:* Number of the record; counted from 0, which is the first record, using the current sort mode. If **Selected** is switched, only the selected records are counted.

*Selected:* See **Record**.

**RETURN** if success (RC = 0), or RC > 0 if no record found.

**SEE ALSO**

Section 4.11 [Record groups], page 14

### 7.2.3.13 setrecordgroups

**SYNTAX** SetRecordGroups 'R=Record/K/N,S=Selected/S,A=AllSelected/S,G1/S,...,G10/S,G=GroupsID/

**FUNCTION**

This function changes group data either of the specified or of all selected records. There are two ways to supply the record group data: The easiest way is to use the Gx switches for the groups you like to be assigned to the record. However, this way you get a lot of function parameters. Therefore you may give directly the numeric value **GroupsID**, which incorporates all the group data in one value. See below to learn how the **GroupIDs** value is used.

If the `Record / Selected` and `AllSelected` keywords are not used, this function will change the current record.

#### PARAMETERS

*Record:* Number of the record; counted from 0, which is the first record, using the current sort mode. If `Selected` is switched, only the selected records are counted.

*Selected:* See `Record`.

*AllSelected:*

Check to edit all selected and not only the current one.

*G1..Gx...G10:*

Use the `Gx` switches to assign a record to one or several groups, where `x` is the ID of group counting from 1 to 10 equal the number in `AA`.

*GroupsID:* A numeric value, that incorporates a set of Groups. `GroupsID` is defined as follows. Use `1=G1`, `2=G2`, `4=G3`, say  $2^{(x-1)}=Gx$  as group values and do a simple addition of the desired group values. Example:  $G3+G4+G10 = 524 = 2^2 + 2^3 + 2^9$ .

**RETURN** if success (`RC = 0`), or `RC > 0` if no record found.

#### SEE ALSO

Section 4.11 [Record groups], page 14

### 7.2.3.14 removeduplicaterecords

**SYNTAX** `RemoveDuplicateRecords D=Dialog/S`

#### FUNCTION

This function removes duplicate (redundant) records from the current database.

#### PARAMETERS

*Dialog:* Use this switch get a dialogbox appearing for each record to be removed. Otherwise all duplicate records will be removed without any confirmation.

**RETURN** `RC = 0`, if there are no more duplicate records in the database. Otherwise `RC > 0`.

#### SEE ALSO

Section 5.5 [Process Menu], page 21

### 7.2.3.15 formatrecordstring

**SYNTAX** FormatRecordString R=Record/N/K,S=Selected/S,FormatString/F/A

**FUNCTION**

This function is useful to retrieve record data formatted in a C-language (sprintf) way. See below for supported shortcuts. Give no arguments to retrieve data from current record, or use the numeric argument **Record** and the **Selected** switch to specify the desired record.

**PARAMETERS**

*Record:* Number of the record; counted from 0, which is the first record, using the current sort mode. If **Selected** is switched, only the selected records are counted.

*Selected:* See **Record**.

*FormatString:*

Text with control sequences to retrieve record data. Currently only the following control sequence are supported: %fx, where x is the Record field id, ranging from 0 = Salutation to 14 = Remark 2. Example %f2 %f1, living at %f3 in %f5.\ne-mail is %f8.

**RETURN** if success (RC = 0), or RC > 0 if no record found.

**SEE ALSO**

Example ARexx scripts.

### 7.2.3.16 selectall

**SYNTAX** SelectAll

**FUNCTION**

This function adds all records to the selection.

**PARAMETERS**

**RETURN** always RC = 0.

**SEE ALSO**

Section 7.2.3.17 [selectnone], page 42, Section 7.2.3.18 [select-toggle], page 42  
 , Section 7.2.3.19 [selectstring], page 42, Section 7.2.3.20 [select-group], page 43  
 Section 4.12 [Record selection], page 14

### 7.2.3.17 selectnone

**SYNTAX** SelectNone

**FUNCTION**

This function removes all records from the selection.

**PARAMETERS**

**RETURN** always RC = 0.

**SEE ALSO**

Section 7.2.3.16 [selectall], page 41, Section 7.2.3.18 [selecttoggle], page 42  
, Section 7.2.3.19 [selectstring], page 42, Section 7.2.3.20 [selectgroup], page 43  
Section 4.12 [Record selection], page 14

### 7.2.3.18 selecttoggle

**SYNTAX** SelectToggle

**FUNCTION**

This function selects or deselects the current record, depending on its selection state.

**PARAMETERS**

**RETURN** always RC = 0.

**SEE ALSO**

Section 7.2.3.16 [selectall], page 41, Section 7.2.3.18 [selecttoggle], page 42  
, Section 7.2.3.19 [selectstring], page 42, Section 7.2.3.20 [selectgroup], page 43  
Section 4.12 [Record selection], page 14

### 7.2.3.19 selectstring

**SYNTAX** SelectString X=Exclude/S,T=Toggle/S,D=Dialog/S,F=FieldFlags/N/K,MatchString/F

**FUNCTION**

Use this function to filter records to or from a selection using the string match filter (Field string match).

**PARAMETERS**

*Exclude:*

*Toggle:* Action switches. Use **Exclude** to remove matching records from selection or **Toggle** to switch their selection state. If no action switch is given, matching records will be added to the selection.

*Dialog:* Use this switch to open the Filter dialogbox from ARexx scripts.

*FieldFlags:*  
This numerical argument is a representation of a set or record fields to match. Omit this keyword to search in all record fields.

*MatchString:*  
Match string with control sequences. See Section 6.4.1 [Field string match], page 29, for match string format.

**RETURN** RC = 0 if success, RC > 0 when failing.

#### **KNOWN BUGS**

The FieldFlags keyword does not work correct. This will be fixed in the next release.

#### **SEE ALSO**

Section 6.4 [Filter dialogbox], page 28

### **7.2.3.20 selectgroup**

**SYNTAX** SelectGroup X=Exclude/S,T=Toggle/S,D=Dialog/S,G1/S,G2/S,G3/S,G4/S,G5/S,G6/S,G7/S,G8/S

#### **FUNCTION**

Use this function to filter records to or from a selection by their group assignment (Group match). There are two ways to indicate the set of groups: The easiest way is to use the Gx switches for the groups. However, this way you get a lot of function parameters. Therefore you may give directly the numeric value **GroupsID**, which incorporates all the groups in one value. See below to learn how the **GroupIDs** value is used.

#### **PARAMETERS**

*Exclude:*

*Toggle:* Action switches. Use **Exclude** to remove matching records from selection or **Toggle** to switch their selection state. If no action switch is given, matching records will be added to the selection.

*Dialog:* Use this switch to open the Filter dialogbox from ARexx scripts.

*G1..Gx...G10:*

Use the Gx switches to indicate the set of groups that should match. x is the ID of group counting from 1 to 10 equal the number in AA.

*GroupsID:* A numeric value, that incorporates a set of Groups. GroupsID is defined as follows. Use 1=G1, 2=G2, 4=G3, say  $2^{(x-1)}=Gx$  as group values and do a simple addition of the desired group values. Example:  $G3+G4+G10 = 524 = 2^2 + 2^3 + 2^9$ .

**RETURN** RC = 0 if success, RC > 0 when failing.

**SEE ALSO**

Section 6.4 [Filter dialogbox], page 28

### 7.2.3.21 removeselection

**SYNTAX** RemoveSelection D=Dialog/S

**FUNCTION**

This function removes all selected records from the database.

**PARAMETERS**

*Dialog:* Use this switch to open a dialogbox for user prompt.

**RETURN** RC = 0 if success, RC > 0 when failing.

**SEE ALSO**

Section 7.2.3.16 [selectall], page 41, Section 7.2.3.18 [selecttoggle], page 42  
 , Section 7.2.3.19 [selectstring], page 42, Section 7.2.3.20 [selectgroup], page 43  
 Section 4.12 [Record selection], page 14

### 7.2.3.22 find

**SYNTAX** Find Key/A,F=Field/N/K,N=Next/S,S=Selected/S,D=Dialog/S

**FUNCTION**

This function performs the string search to find a record.

**PARAMETERS**

*Key:* Key string.

- Field:* Numerical value with field ID, indicating the field that should match the key string.
- Next:* Supply this switch to find next record, using current key string.
- Selected:* If this switch is supplied, only the selected records will be searched. The number of the return record applies to the selected records.
- Dialog:* Use this switch to prompt the users for key string.

**RETURN** Number of the record found. If **Selected** is switched, then the returned number applies to the selected records.

**SEE ALSO**

Section 4.7.2 [Find a record], page 12

### 7.2.3.23 sort

**SYNTAX** Find Index1/K/N,Descend1/K/S,Index2/K/N,Descend2/K/S,Index3/K/N,Descend3/K/S,D=Dialog  
**FUNCTION**

Use the **Sort** function change sort index and order of the database. You may call this function several times to sort with multiple indices and directions.

**PARAMETERS**

*IndexX:* Numerical representation of the record field to sort for. X is the number of the sort order. The record field value ranges from 0 = Salutation to 14 = Remark 2.

*DescendX:*  
 Use this switch to sort in descending order.

*Dialog:* Use this switch to open the sort dialogbox.

**RETURN** RC = 0 if success, RC > 0 when failing.

**SEE ALSO**

Section 4.8 [Database sort], page 13  
 Section 6.5 [Sort dialogbox], page 30

### 7.2.3.24 next

**SYNTAX** Next

**FUNCTION**

Function moves to next record in the database and sets it as the current one.

**PARAMETERS**

**RETURN** always return RC = 0

**SEE ALSO**

Section 7.2.3.25 [previous], page 46, Section 7.2.3.26 [first], page 46, Section 7.2.3.27 [last], page 46

**7.2.3.25 previous**

**SYNTAX** Previous

**FUNCTION**

Function moves to the previous record in the database and sets it as the current one.

**PARAMETERS**

**RETURN** always return RC = 0

**SEE ALSO**

Section 7.2.3.24 [next], page 45, Section 7.2.3.26 [first], page 46, Section 7.2.3.27 [last], page 46

**7.2.3.26 first**

**SYNTAX** First

**FUNCTION**

Function moves to the first record in the database and sets it as the current one.

**PARAMETERS**

**RETURN** always return RC = 0

**SEE ALSO**

Section 7.2.3.24 [next], page 45, Section 7.2.3.25 [previous], page 46, Section 7.2.3.27 [last], page 46

**7.2.3.27 last**



**SYNTAX** Last

**FUNCTION**

Function moves to the last record in the database and sets it as the current one.

**PARAMETERS**

**RETURN** always return RC = 0

**SEE ALSO**

Section 7.2.3.24 [next], page 45, Section 7.2.3.25 [previous], page 46, Section 7.2.3.26 [first], page 46

### 7.2.3.28 current

**SYNTAX** Current Record/N,S=Selected/S

**FUNCTION**

Function moves to specified record and sets it as the current one.

**PARAMETERS**

*Record:* Number of the record; counted from 0, which is the first record, using the current sort mode. If *Selected* is switched, only the selected records are counted.

*Selected:* See *Record*.

**RETURN** always RC = 0

**SEE ALSO**

Section 7.2.3.24 [next], page 45, Section 7.2.3.25 [previous], page 46

### 7.2.3.29 getgroupname

**SYNTAX** SetGroupname G1/S, . . . ,G10/S

**FUNCTION**

Use this function to retrieve the name of a specific record group.

**PARAMETERS**

*G1..Gx...G10:*

Use exactly one of the Gx switches to indicate the group; where x is the ID of the group counting from 1 to 10.

**RETURN** if success, RC = 0 with RESULT = <name of group>  
on error, RC > 0

**SEE ALSO**

Section 4.11 [Record groups], page 14

### 7.2.3.30 setgroupname

**SYNTAX** SetGroupname G1/S, . . . , G10/S, Name/F

**FUNCTION**

This function changes the name of the record group specified with one Gx switch.

**PARAMETERS**

*G1..Gx...G10:*

Use exactly one of the Gx switches to indicate the group; where x is the ID of the group counting from 1 to 10.

*Name:* Text parameter, containing the new name of the group.

**RETURN** if success, RC = 0 else RC > 0

**SEE ALSO**

Section 4.11 [Record groups], page 14

### 7.2.3.31 getfreename

**SYNTAX** GetFreename F1/S, . . . , F4/S

**FUNCTION**

Use this function to retrieve the name of a specific FreeX record field.

**PARAMETERS**

*F1/S, . . . , F4/S:*

Use exactly one of the Fx switches to indicate the field to be changed; where x is the ID of the FreeX record field counting from 1 to 4.

**RETURN** if success, RC = 0 with RESULT = <name of field>  
on error, RC > 0

**SEE ALSO**

Section 4.5.2 [User definable fields], page 10

### 7.2.3.32 setfreename

**SYNTAX** SetFreename F1/S, . . . , F4/S, Name/F

**FUNCTION**

This function changes the name of the FreeX record field specified with one Fx switch.

**PARAMETERS**

*F1/S, . . . , F4/S:*

Use exactly one of the Fx switches to indicate the field; where x is the ID of the FreeX record field counting from 1 to 4.

*Name:* Text parameter, containing the new name of the field.

**RETURN** if success, RC = 0 else RC > 0

**SEE ALSO**

Section 4.5.2 [User definable fields], page 10

### 7.2.3.33 getnumofrecords

**SYNTAX** GetNumofRecords S=Selected/S

**FUNCTION**

Use this function to retrieve the number of all or only the selected records.

**PARAMETERS**

*Selected:* If the Selected switch is supplied, the number of records in the selection is returned. Else the total number of records in the database is returned.

**RETURN** always RC = 0 with RESULT = <number of records>

**SEE ALSO**

### 7.2.3.34 savepath

**SYNTAX** SavePath

**FUNCTION**

This function saves the path of the current database as the default startup path, which is equal to using the Save path menu item from the Project Menu.

**PARAMETERS**

**RETURN** always RC = 0

**SEE ALSO**

### 7.2.3.35 setlistview

**SYNTAX** SetListview A=All/S,S=SelectedOnly/S,U=UnselectedOnly,G1/S,...,G10/S,G=GroupsID/K/N

**FUNCTION**

This function changes the set of records displayed in the Database list view.

**PARAMETERS**

*All:* All records will be displayed.

*SelectedOnly:*

Only selected records will be displayed.

*UnselectedOnly:*

Only records not in the selection will be displayed.

*G1,...,G10:*

Use the Gx switches to indicate the set of groups that should match. x is the ID of group counting from 1 to 10 equal the number in AA.

*GroupsID:* A numeric value, that incorporates a set of Groups. GroupsID is defined as follows. Use 1=G1, 2=G2, 4=G3, say  $2^{(x-1)}=Gx$  as group values and do a simple addition of the desired group values. Example:  $G3+G4+G10 = 524 = 2^2 + 2^3 + 2^9$ .

**RETURN** always RC = 0

**SEE ALSO**

Section 4.6 [Database view], page 11

### 7.2.3.36 setview

**SYNTAX** SetView L=ListView/S,F=FormView/S

**FUNCTION**

Use this function to change the Database view type from List view to Form view or vice versa. Do not supply to any argument to toggle view type. Use the switches to ensure a specific view type.

**PARAMETERS**

*ListView:*

*FormView:*

Switches the database view to the desired view type.

**RETURN** always RC = 0

**SEE ALSO**

Section 4.6 [Database view], page 11

### 7.2.3.37 remark1window

**SYNTAX** Remark1Window 0=Open/S,C=Close/S

**FUNCTION**

This function opens or close the window containing the Remark1 field.

**PARAMETERS**

*Open:*

*Close:* Action switches.

**RETURN** always RC = 0

**SEE ALSO**

Section 4.6 [Database view], page 11

### 7.2.3.38 remark2window

**SYNTAX** Remark2Window 0=Open/S,C=Close/S

**FUNCTION**

This function opens or close the window containing the Remark2 field.

**PARAMETERS**

*Open:*

*Close:* Action switches.

**RETURN** always RC = 0

**SEE ALSO**

Section 4.6 [Database view], page 11

### 7.2.3.39 exportselection

**SYNTAX** ExportSelection D=Dialog/S,OutFilename

**FUNCTION**

This function exports the currently selected records as an AA database file. You may load these records as an own database project or import the into other database. In contrary to ASCII transfer, all preferences data (Group and FreeX names) as well as Record group information will be preserved.

**PARAMETERS**

*Dialog:* Use this switch to open the filename dialogbox.

*OutFilename:*

Text argument containing filename of Database selection file.

**RETURN** RC = 0, if selection exported, else RC > 0

**SEE ALSO**

Section 4.12 [Record selection], page 14

**7.2.3.40 exportascii**

**SYNTAX** ExportASCII S=SelectedOnly/S,C=CurrentOnly/S,D=Dialog/S,P=FormatFilename/K,OutFile

**FUNCTION**

This function exports a set of records as an ASCII text file. Use the name of previously saved Export preferences ('\*.xpref') to define the format of the export file.

**PARAMETERS**

*SelectedOnly:*

*CurrentOnly:*

Use on of these switches to export only a specific set of records: **Selected** or **Current**. Ommit to export all records.

*Dialog:* Use this switch to open the Section 6.2 [Transfer ASCII dialogbox], page 27.

*FormatFilename:*

Name of the format file. Supply either full path or only the filename. AA searches first in the current dir, the in the local ('Prefs/') and finally in the global ('envarc:AddressAssist/') preferences directory.

*OutFilename:*

Text argument containing filename of export file.

**RETURN** RC = 0, if selection exported, else RC > 0

**SEE ALSO**

Section 4.12 [Record selection], page 14

**7.2.3.41 importselection**

**SYNTAX** ImportSelection S=Select/S,G1/S,...,G10/S,G=GroupsID/K/N,D=Dialog/S,InFilename  
**FUNCTION**

Use this function to import the contents of an AA database file.

**PARAMETERS**

*Select:* Switch to add imported records to the selection. Otherwise the imported records will not be selected.

*G1,...,G10:* Switch a set of Gx switches, to assign the imported records to these groups. Note: The specified set of groups will be superimposed to the set of groups already assigned to the imported records. I.e. by omitting Gx switches, the record groups will be preserved.

*GroupsID:* Numerical value indicating the set of groups to assign to the imported records. This group information will be superimposed to the existing group assignments. See also G1,...,G10.

*Dialog:* Use this switch to prompt user.

*InFilename:* Text argument containing filename of Database file.

**RETURN** RC = 0, if selection exported, else RC > 0

**SEE ALSO**

Section 4.12 [Record selection], page 14

**7.2.3.42 importascii**

**SYNTAX** ImportASCII S=Select/S,G1/S,...,G10/S,G=GroupsID/K/N,D=Dialog/S,P=FormatFilename/K  
**FUNCTION**

Use this function to import the contents of an ASCII transfer file. Use the name of previously saved Export preferences ('\*.xpref') to define the format of the ASCII transfer file.

**PARAMETERS**

*Select:* Switch to add imported records to the selection. Otherwise the imported records will not be selected.

*G1,...,G10:*

Switch a set of Gx switches, to assign the imported records to these groups. Note: The specified set of groups will be superimposed to the set of groups already assigned to the imported records. I.e. by omitting Gx switches, the record groups will be preserved.

*GroupsID:* Numerical value indicating the set of groups to assign to the imported records. This group information will be superimposed to the existing group assignments. See also *G1, . . . ,G10*.

*Dialog:* Use this switch to prompt user.

*FormatFilename:*

Name of the format file. Supply either full path or only the filename. AA searches first in the current dir, then in the local ('Prefs/') and finally in the global ('envarc:AddressAssist/') preferences directory.

*InFilename:*

Text argument containing filename of Database file.

**RETURN** RC = 0, if selection exported, else RC > 0

**SEE ALSO**

Section 4.15.2 [Import/Export ASCII], page 15

**7.2.3.43 dialogbox**

**SYNTAX** DialogBox C=ControlString/K,Message/A/F

**FUNCTION**

This function opens a simple dialogbox for user prompt, information or selection. The dialogbox consists of a Message and a set of control buttons, specified by a control string.

**PARAMETERS***ControlString:*

Text representation of the buttons in the dialogbox. Format: \*But1...But\_X, where ButX are the names of the x buttons in the set. The button marked with a preceding \* is the default button. The char preceded by an underscore (\_) in the name will be underlined. Example: \*\_OK\_A11\_Cancel will create 3 buttons OKAllCancel.



*ControlString:*

Text string containing dialogbox message.

**RETURN** depends on the button used to close the dialogbox: The button to the left side will return RC = 1, the button on the righthand side returns RC = 0. The other buttons return their ID: RC=2 for the second button from left, say RC=i for the i th button from left.

**SEE ALSO**

Example ARexx scripts.

**7.2.3.44 getfile**

**SYNTAX** GetFilename I=InitialFilename/K,F=FilterString/K,Title/A/F

**FUNCTION**

This function prompts the user to enter a filename.

**PARAMETERS***InitialFilename:*

An initial filename may be supplied with this keyword.

*FilterString:*

This is an AMIGA-DOS file pattern string, to filter the files displayed. Default: #? display all. Example: #?.arx displays only ADDRESSASSIST ARexx scripts.

*Title:* Supplies the title of the file requester.

**RETURN** On OK: RC = 0 and RESULT = <path & filename>, else RC > 0

**SEE ALSO**

Example ARexx scripts.

## 8 Preferences

AA is highly configurable. Besides the application and database specific configuration, you may change the look and feel of ADDRESSASSIST's GUI through MUI's powerful GUI preference program. This works only, if you use a registered version of MUI (another reason to register this favourable product).

This section covers only AA and database specific preferences. See in the MUI docs for information about the GUI preferences.

AA distinguishes between application and database preferences, which may be configured by the user:

### Database (project):

- the names of the 4 user definable record fields (Section 6.7 [Preferences dialogbox], page 31)
- the names of the 10 groups (Section 4.11 [Record groups], page 14), (Section 6.7 [Preferences dialogbox], page 31)
- the format of the Database Listview (Section 6.7 [Preferences dialogbox], page 31)
- the stuff in the **Application** page of the preferences editor (Section 6.7 [Preferences dialogbox], page 31)
- the Record template specific setup (Section 4.14 [Record template], page 15).
- the current sort direction and index (Section 6.5 [Sort dialogbox], page 30)
- the name of the actual print setup (Section 4.9 [Database print], page 13),
- the name of the actual ASCII transfer setup (Section 4.15.2 [Import/Export ASCII], page 15).

### Application:

- All the Database preferences, used when a new database is created.
- the path of the Help file (Section 4.3 [Help!], page 8),
- the path of the database, which is automatically reloaded on startup (**Save path** from `<undefined>` [Project menu], page `<undefined>`)
- the ARexx script menu and the ARexx console definition Chapter 7 [ARexx], page 32

## 9 Program history & future developments

### 9.1 Program history

- **Release 3.0 (29.1.97), version 41.1:**
  - Mayor update: no detailed list available.
- **Release 2.1 (5.9.95), version 40.17:**
  - Completed english and german documentation.
  - Added a french translation of the documentation and the catalogs. A big bunch of thanks go to Alexandre Sforza
  - Added an italian tranlation of the catalog file. Thanks go to Alessandro Basso
  - Pressing return in the database listview will change to the record view of the current record.
  - Some bugs fixed and internal optimisations.
- **Release 2.1 beta 1 (21.6.95), version 40.16:**
  - Added 2 multi-line remark fields in separate so called remark windows.
  - Enhanced database views: now you can choose the set of record to be viewed.
  - Added record template function.
  - Some bugs fixed and internal optimisations.
- **Release 2 (11.6.95), version 40.15:**
  - German documentation added and English documentation enhanced.
  - Norwegian catalog translation added (by Eirik Boggsnes)
  - Many minor bugs fixed and internal optimisations.
- **Release 2 beta 4 (30.5.95), version 40.14:**
  - Record form view: Field data is now saved without pressing ENTER each time you change a field string.
  - Record form view: View adjusted to be height flexible
  - Preferences window: Auto move to next text field added.
  - Print Preferences: Changed pattern of print prefs files to #?.ppref instead of \*.ppref
  - Key search: now searches a string in all record fields.
  - General: Check Name AND Name2 field for record addition. Now only one of the Name fields, Name OR Name2, must contain data to add a record to the database.
  - General: Localized Continue and MUI copyright string in AboutWindow as well as the Title/Status string.
  - General: Cleanup Config behaviour, separated program/database prefs data
  - Many minor bugs fixed and internal optimisations.

- **Release 2 beta 3 (23.5.95), version 40.12:**
  - Preferences dialog: Added Save button to save changed preferences directly
  - Preferences: Added switch to turn off Record Save dialog.
  - Printer Interface: Added switch to suppress FormFeeds (FF), ignored when printing to file.
  - Documentation: updated
  - Changed preferences file format for future extensions!!
  - General: Added function to delete selected records.
  - General: Added Phone1 and Phone2 index for sorting (missing Name2 was a bug!).
  - General: 'Turbo' version no longer needs FPU. Why should it????
  - Many bugs fixed
  
- **Release 2 beta 2 (15.5.95), version 40.11:**
  - Bug fixed: Selection toggle button/menu no work properly
  - Bug fixed: List/label print field length bug fixed
  - Documentation: updated to release 2
  
- **Release 2 beta 1 (9.5.95), version 40.10:**
  - General: Added 2 new database-field: Firstname and Group field for 10 different groups
  - Views: Added view concept by providing 2 different views switchable in one window: database list view and record form view
  - Printer Interface: Added user definable record field ordering
  - Printer Interface: Added database field length for line list printing
  - Printer Interface: Load/Save individual print setup
  - Printer Interface: Added switch to print only used fields of an address (thanks to A. Becker for its idea)
  - Selections: Added powerful selection filters: field string match and group match
  - Some bug fixes
  - Sorting: Added more all sort index, except for phone fields
  - General: New preferences file structure, save preferences in address project files. Added support for project icon and WB-Start with project file.

- Some bug fixes
  
- **Release 1.0 (26.10.94), version 38.15:**
  - First public release.

## 9.2 Future developments

This is a list of future enhancements for ADDRESSASSIST. Remember, the future of this program also depends on you, the user: register immediately and make suggestions, bug reports.

- **Enhance ASCII Import/Export:**
  - Add <IGNORE> fields to the ASCII format definitions.
- **Enhance printing**
  - Add postscript / graphics printing abilities
- Enhance Remark windows to hold external data as well: text, graphics, sound and document links (f.e. foe PageStream)
- Add more flexible output format, user definable, f.e. "printf"-format using specific markers
  
- Add string list salutation, place and code fields.
- Add datatype to display AA data with tools like Multiview.
  
- Any ideas are welcome.

## 10 Copyright and Registration

ADDRESSASSIST is copyrighted 1994-96 by Quarz Development / Alex H. Schneider. All rights reserved. The program is distributed as **Shareware**.

The **freely distributable** archive contains the unregistered version of the program. It is **limited to work with only 15 records**.

The freely distributable archive may be freely redistributed, as long as no charges other than reasonable copying and handling fees are collected. The program may be distributed only as originally released, in the complete archived form. No changes of any kind may be made to the ADDRESSASSIST binary, documents or archive, and I will not answer for any problems that may result from a modified binary.

The suggested basic registration conditions is listed in the 'Orderform.txt'. Registration gets you an entry in the users and a personal **key file**. This key file disables the restrictions of 15 records and will work with all updates of the program that will be available on the networks in the future. This key file is *personal* and definitely *not* freely distributable. You will receive the **key-file** immediately after I got the registration fee by email or by mail.

Take a look at 'Orderform.txt' to register or for more information about registering.

Register now!!!! Thank you for supporting Shareware.

### Trademarks

Amiga, Amiga Workbench and Amiga Kickstart are registered trademarks of Commodore-Amiga Inc.

Installer and Installer project icon are © Copyright 1991-93 Commodore-Amiga, Inc. All Rights Reserved.

MUI is © Copyright 1993-96 Stefan Stuntz (see 'Readme.MUI') for details.

Textfield.gadget is © Copyright 1995 Mark Thomas (see 'TextFieldGadget.txt') for details.

All other named trademarks reference to their appropriate manufacturer or seller.

## 11 Disclaimer

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDER AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## 12 Acknowledgements

Despite all the efforts of me (the author), ADDRESSASSIST would never be what it is now without the help of several people and institutions, who should be thanked very much for their contribution at this place:

- My **wife** and my little daughter **Ramona** for their patience.
- A very special thanks goes to **Arto Becker**, **Eirik Bogsnes**, **Kevin Kirk** and **Alexandre Sforza** (in alphabetical order) for their great, critical beta testing and the many suggestions, flames and blames:) !
- An extra thanks goes to **Alexandre Sforza**, who did the great work of translating the catalogs *and* the manual to *french*. All those who have ever written a GNU-Texinfo file know what I mean! .
- **Eirik Bogsnes** for his Norwegian translation of the catalog file.
- **Alessandro Basso** for his Italian translation of the catalog file.
- The **SAS Institute** for their excellent C/C++ compiler. Hope they will come back!!
- **Stefan Stuntz** for his great Magic User Interface (MUI)
- **Mark Thomas** for his more than only useful `textfield.gadget`.
- **Escom/AT** and **Viscorp** for buying the AMIGA and last but not least **Commodore** for ...!



## 13 Contacting the author

For questions, registration, suggestions or bug reports please contact me by mail:

Alex H. Schneider  
Kruggasse 179  
CH-5462 Siglistorf  
Switzerland / Europe

or by email, internet (no registration):

`schneider@igt.baum.ethz.ch`

or by phone (no registration):

++41 56 243 18 13 (at home, NEW!!!!)

++41 1 633 30 40 (at work)

# Index

## A

ARexx commands .....	32
ARexx dialogbox .....	30
ARexx Introduction .....	32
ARexx Menu .....	22
ARexx, Application specific commands .....	33
ARexx, command argument standards .....	32
ARexx, commands .....	32
ARexx, menu .....	32
ARexx, MUI standard commands .....	33
ASCII exchange file format .....	16

## C

Change path of helpfile .....	8
Contacting the author .....	63
Copy record .....	19
Copyright .....	60

## D

Database format .....	9
Database navigation .....	12
Database print .....	13
Database sort .....	13
Database view .....	11
Database view types .....	11
Dialogbox Reference .....	23
Duplicate record .....	19

## E

Edit record .....	19
-------------------	----

## F

Field string match, selection filter .....	29
Filter dialogbox .....	28
Filter, selection .....	28
Find a record .....	12
FreeX fields .....	10
Future .....	57

## G

Group dialogbox .....	28
Group match, selection filter .....	29
Groups of records .....	19

## H

Helpfile .....	8
History .....	57

## I

Icon, change data icon .....	5
Image-buttons, toolbar .....	9
Import and export .....	15
Import/Export ASCII .....	15
Import/Export Casio .....	17
Import/Export selection .....	15
Installation, application .....	5
Introduction .....	1

## M

Menu strip .....	18
MUI, Magic User Interface .....	2

## N

Navigation controls .....	12
New record .....	19

## P

Preferences dialogbox .....	31
Preferences window .....	56
Preferences, database and application ..	56
Print dialogbox .....	23
Print record .....	19
Process Menu .....	21
Program startup .....	6
Project Menu .....	18

## Q

Quick Tour .....	6
------------------	---

**R**

Record edit .....	13
Record field string match, selection filter .....	29
Record group match, selection filter ...	29
Record groups .....	14
Record links .....	14
Record Menu .....	19
Record selection.....	14
Record template .....	15
Record templates .....	19
Registration.....	60
Remark fields, windows .....	10
Remark windows .....	11
Remove record .....	19

**S**

Save record .....	19
Selection Filter dialog.....	28

Selection Menu.....	20
Selective record view .....	12
Sort dialogbox .....	30

**T**

Templates, for records.....	19
textfield.gadget.....	2
The remark fields .....	10
Toolbar .....	9
Transfer ASCII dialogbox .....	27
Tutorial.....	6

**U**

user definable fields .....	10
-----------------------------	----

**V**

View Menu .....	19
Views, view types.....	11

# Table of Contents

<b>1</b>	<b>Introduction</b> .....	<b>1</b>
1.1	Program features .....	1
1.2	Magic User Interface (MUI).....	2
1.3	Textfield gadget .....	2
<b>2</b>	<b>What is new</b> .....	<b>3</b>
2.1	Release 3.0 (January 29th 1997).....	3
<b>3</b>	<b>Installation</b> .....	<b>5</b>
<b>4</b>	<b>Overview</b> .....	<b>6</b>
4.1	Program startup.....	6
4.2	Quick Tour .....	6
4.3	Help!.....	8
4.4	Image-button toolbar .....	9
4.5	Database format.....	9
4.5.1	The record fields.....	10
4.5.2	User definable fields .....	10
4.5.3	The remark fields.....	10
4.6	Database view .....	11
4.6.1	Database view types.....	11
4.6.2	Remark windows .....	11
4.6.3	Selective record view.....	12
4.7	Database navigation .....	12
4.7.1	Navigation controls.....	12
4.7.2	Find a record .....	12
4.8	Database sort.....	13
4.9	Database print.....	13
4.10	Record edit .....	13
4.11	Record groups.....	14
4.12	Record selection .....	14
4.13	Record links.....	14
4.14	Record template .....	15
4.15	Import and export.....	15
4.15.1	Import/Export selection .....	15
4.15.2	Import/Export ASCII.....	15
4.15.2.1	ASCII file format.....	16
4.15.3	Import/Export Casio.....	17

<b>5</b>	<b>Menu Reference</b> .....	<b>18</b>
5.1	Project Menu .....	18
5.2	View Menu .....	19
5.3	Record Menu .....	19
5.4	Selection Menu .....	20
5.5	Process Menu .....	21
5.6	ARexx Menu .....	22
<b>6</b>	<b>Dialogbox Reference</b> .....	<b>23</b>
6.1	Print dialogbox .....	23
6.1.1	Page setup .....	24
6.1.2	Field options page .....	24
6.1.3	Header & Footer page .....	25
6.1.3.1	Header & Footer control chars .....	26
6.1.4	Printer setup page .....	26
6.2	Transfer ASCII dialogbox .....	27
6.3	Group dialogbox .....	28
6.4	Filter dialogbox .....	28
6.4.1	Field string match .....	29
6.4.2	Group match .....	29
6.5	Sort dialogbox .....	30
6.6	ARexx dialogbox .....	30
6.7	Preferences dialogbox .....	31
<b>7</b>	<b>ARexx</b> .....	<b>32</b>
7.1	ARexx Introduction .....	32
7.2	ARexx commands .....	32
7.2.1	Command argument standards .....	32
7.2.2	MUI standard commands .....	33
7.2.3	Application specific commands .....	33
7.2.3.1	newdatabase .....	34
7.2.3.2	loaddatabase .....	34
7.2.3.3	savedatabase .....	34
7.2.3.4	print .....	35
7.2.3.5	newrecord .....	35
7.2.3.6	getrecorddata .....	36
7.2.3.7	setrecorddata .....	36
7.2.3.8	duplicaterecord .....	37
7.2.3.9	removerecord .....	37
7.2.3.10	savetemplate .....	38
7.2.3.11	copytoclip .....	38
7.2.3.12	getrecordgroups .....	39
7.2.3.13	setrecordgroups .....	39
7.2.3.14	removeduplicaterecords .....	40

7.2.3.15	formatrecordstring .....	41
7.2.3.16	selectall .....	41
7.2.3.17	selectnone .....	42
7.2.3.18	selecttoggle .....	42
7.2.3.19	selectstring .....	42
7.2.3.20	selectgroup .....	43
7.2.3.21	removeselection .....	44
7.2.3.22	find .....	44
7.2.3.23	sort .....	45
7.2.3.24	next .....	45
7.2.3.25	previous .....	46
7.2.3.26	first .....	46
7.2.3.27	last .....	46
7.2.3.28	current .....	47
7.2.3.29	getgroupname .....	47
7.2.3.30	setgroupname .....	48
7.2.3.31	getfreename .....	48
7.2.3.32	setfreename .....	49
7.2.3.33	getnumofrecords .....	49
7.2.3.34	savepath .....	49
7.2.3.35	setlistview .....	50
7.2.3.36	setview .....	50
7.2.3.37	remark1window .....	51
7.2.3.38	remark2window .....	51
7.2.3.39	exportselection .....	51
7.2.3.40	exportascii .....	52
7.2.3.41	importselection .....	53
7.2.3.42	importascii .....	53
7.2.3.43	dialogbox .....	54
7.2.3.44	getfile .....	55
<b>8</b>	<b>Preferences .....</b>	<b>56</b>
<b>9</b>	<b>Program history &amp; future developments ....</b>	<b>57</b>
9.1	Program history .....	57
9.2	Future developments .....	59
<b>10</b>	<b>Copyright and Registration .....</b>	<b>60</b>
	Trademarks .....	60
<b>11</b>	<b>Disclaimer .....</b>	<b>61</b>
<b>12</b>	<b>Acknowledgements .....</b>	<b>62</b>

**13 Contacting the author ..... 63**

**Index ..... 64**