

ShimmerDB

J. Tierney

Copyright © CopyrightÂ©1995-96 J. Tierney

COLLABORATORS

	<i>TITLE :</i> ShimmerDB		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	J. Tierney	July 1, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ShimmerDB	1
1.1	ShimmerDB Documentation	1
1.2	Welcome to ShimmerDB	2
1.3	ShimmerDB: Requirements	3
1.4	ShimmerDB: Legal Issues	3
1.5	ShimmerDB: Installation	4
1.6	ShimmerDB: Start Up Options	4
1.7	ShimmerDB: Field Types	5
1.8	ShimmerDB: Main Interface	7
1.9	ShimmerDB: Menus	8
1.10	ShimmerDB: Hotkeys	9
1.11	ShimmerDB: Search Requester	10
1.12	Menu: Project	10
1.13	Menu: Edit	12
1.14	Menu: Search	13
1.15	Menu: Sort	13
1.16	Menu: Settings	14
1.17	ShimmerDB GUI: Title Bar	14
1.18	ShimmerDB GUI: Position Gadget	14
1.19	ShimmerDB GUI: Filename Gadget	15
1.20	ShimmerDB GUI: Edit-Type Cycle Gadget	15
1.21	ShimmerDB GUI: Edit Listview	16
1.22	ShimmerDB GUI: Edit String Gadget	16
1.23	ShimmerDB GUI: Edit Mode Gadget	16
1.24	ShimmerDB GUI: Delete Gadget	17
1.25	ShimmerDB GUI: View Slider	17
1.26	ShimmerDB GUI: Field Data String Gadget	17
1.27	ShimmerDB GUI: Field Slider	18
1.28	Search GUI: Set Pattern	18
1.29	Search GUI: Find Next Button	18

1.30 Search GUI: Find Previous Button	19
1.31 Search GUI: Ignore Options	19
1.32 Search GUI: Scan Options	19
1.33 Search GUI: Clear Pattern List	20
1.34 AmigaDOS Pattern Matching	20
1.35 ShimmerDB: Limits	20
1.36 ShimmerDB: File Format	21
1.37 ShimmerDB: Clipboard	21
1.38 ShimmerDB: Known Bugs	22
1.39 ShimmerDB: 'There's somethin' screwy goin' on here...'	23
1.40 ShimmerDB: The Next Generation	23
1.41 ShimmerDB: The Early Years	24
1.42 ShimmerDB: The Name Game	24
1.43 ShimmerDB: The Shocking Truth	24
1.44 ShimmerDB: Credit Where Credit is Due	25
1.45 ShimmerDB: Beware of Romulans Bearing Gifts.	25
1.46 ShimmerDB: The Author	26

Chapter 1

ShimmerDB

1.1 ShimmerDB Documentation

```

                                     _/_/  _/_/  _/_/
                                     _/  _/_/_/_/
_/_/  _/_/  _/_/                    _/  _/  _/  _/
_/_/  _/_/  _/_/_/  _/  _/_/_/  _/  _/_/_/  _/  _/_/  _/_/  _/  _/  _/  _/  _/
  _/_/  _/  _/  _/  _/  _/  _/  _/  _/  _/  _/_/_/  _/  _/  _/  _/_/_/
_/_/  _/_/  _/  _/  _/  _/  _/  _/  _/  _/  _/  _/  _/  _/  _/  _/  _/
_/_/_/  _/  _/  _/  _/  _/  _/  _/  _/  _/  _/_/_/  _/  _/_/_/  _/_/_/

```

ShimmerDB v1.0
©1995-1996 J. Tierney

Introduction

What ShimmerDB is.

Requirements

What ShimmerDB wants.

Legalities

Copyright, Disclaimer, and Distribution information.

Installation

What goes where.

Startup

ShimmerDB startup options.

Field Types

Fields can do more than just hold a line of text...

The Interface

Main Window

Menus

Hotkeys

Search Requester
Technical Details

Limits

"Cap'n, the database canna take much more'a this!"

File Format

How ShimmerDB saves data.

Clipboard

How clipboard data is stored/handled.

Known Bugs

Known Oddities

"Ground control to Major Tom..."

To Do

Future goals.

History

Credits

Important influences.

ShimmerDB?!

"What's in a name..."

The Author

How to reach me.

1.2 Welcome to ShimmerDB

Introduction:

ShimmerDB is a flat file database, and a Gilligan's Island kind of project. It began as a simple three hour cruise when a friend, Greg Donner, asked for a program that would open a window with a few string gadgets in it and load and save the gadget's contents. The result was a program that resembled the left half of ShimmerDB's interface and behaved a lot like a line editor (when it behaved at all :-)

A suggestion of improvements by Greg resulted in an initial reaction of, "Eek! That's a database!", and eventually this program, ShimmerDB. Although still a work in progress, version 1.0 of ShimmerDB is stable and fully functional, with more than a few nifty features.

Features:

- Store multiple lines of text (up to 65,500) in any field.
 - A font sensitive, resizable, and (mostly) style guide compliant interface.
-

- 65,500 records per database, and each record may contain up to 65,500 fields.
- Clipboard support: cut, copy, and paste to and from units 0 - 9.
- Fields may be set to execute AmigaDOS commands or applications, and the data from the current or other fields may be passed as parameters to the command.
- Databases may easily be combined with other databases.
- Fields, records, and multi-line text each have a 100 item undelete buffer.
- Runs on any public screen.
- Tested with Enforcer and MungWall.

1.3 ShimmerDB: Requirements

Required:

- Workbench 2.04 or higher.

Recommended:

- CycleToMenu - Adds drop-down menus to cycle gadgets.
- MultiCX or NewEdit, for clipboard support in string gadgets.
- Memory - As ShimmerDB keeps all database data in memory, the amount of RAM in your computer determines how large your databases may become. However, ShimmerDB is memory efficient and dynamically allocates and frees memory as needed.

1.4 ShimmerDB: Legal Issues

Copyright Info:

ShimmerDB, copyright ©1995 - 1996 Jody J. Tiereny. All rights reserved.
...with an option on the lefts, too.

ShimmerDB is GiftWare. If you find ShimmerDB to be useful, and/or would like to support and encourage the author, then consider sending the

author
a
gift
.

Disclaimer:

No warranties, expressed or implied. Use this program at your own risk. Although much effort has been, and continues to be, put into this program to ensure safe and stable operation, I take no responsibility for any damages

and/or losses incurred through the use/misuse of this program.

However, I will take credit for anything you like about the program. :-)

Distribution:

ShimmerDB may be freely distributed in it's original form so long as no fee, beyond a nominal copying charge to cover time and media, which is not to exceed \$4 US, is attached.

ShimmerDB may not be included with any commercial product without written permission from the author.

Special permission is granted to include ShimmerDB in public domain collections such as Fred Fish's disks and CDs, and the Aminet CDs. A copy of the disk, while not mandatory, would be appreciated.

1.5 ShimmerDB: Installation

Installation:

ShimmerDB, and optionally it's icon, may be placed anywhere you see fit.

1.6 ShimmerDB: Start Up Options

Starting ShimmerDB:

ShimmerDB may be started from Workbench or the CLI. All arguments are optional.

CLI Arguments:

Format: [[FILE] <filename>] [X=LEFTEDGE <#>] [Y=TOPEDGE <#>] [W=WIDTH <#>]
[H=HEIGHT <#>] [VO=VIEWOFFSET <#>] [NO=NAMEOFFSET <#>] [F=FONT
<fontname>/<fontsize>] [PS=PUBSCREEN <screen name>] [NI=NOICONS]

Template: FILE,X=LEFTEDGE/K/N,Y=TOPEDGE/K/N,W=WIDTH/K/N,H=HEIGHT/K/N,
VO=VIEWOFFSET/K/N,NO=NAMEOFFSET/K/N,F=FONT/K,PS=PUBSCREEN/K,
NI=NOICONS/S

- FILE <filename> - Loads <filename>. The keyword "FILE" is optional.
 - X, LEFTEDGE
 - Y, TOPEDGE - The coordinates to open ShimmerDB's window at. Defaults to 0, 0.
 - W, WIDTH
 - H, HEIGHT - The dimensions of ShimmerDB's window. Defaults to the minimum size (which depends on the size of the font used).
 - VO, VIEWOFFSET - This value is added to the minimum size of the edit listview to determine how wide the edit listview and field views should be. Defaults to 0.
 - NA, NAMEOFFSET - This value is added to the minimum size allocated to
-

display field names in the field view. Determines how much space in the field view is for field names, and how much is for field data. Defaults to 0.

- F, FONT - The font name and size to use. Specified as <fontname>/<size> e.g. FONT Helvetica/13 or F Topaz/9. Defaults to the screen font.
- PS, PUBSCREEN - The name of the public screen to use. Defaults to the default public screen (usually Workbench).
- NI, NOICONS - Do not save icons with projects. By default, projects will be saved with an icon, and the LEFTEDGE, TOPEDGE, WIDTH, HEIGHT, VIEWOFFSET, and NAMEOFFSET tooltypes will be set to their current values at the time of the save.

ToolTypes:

The tool types function as their CLI equivalents.

LEFTEDGE
 TOPEDGE
 WIDTH
 HEIGHT
 VIEWOFFSET
 NAMEOFFSET
 FONT
 PUBSCREEN
 NOICONS

1.7 ShimmerDB: Field Types

Field Types Info:

In addition to just holding normal data, field's may also hold multiple lines of text, and be set to execute AmigaDOS commands or applications. Data from any field, including the one executing the command, may be used as parameters for launched program.

To achieve this, set the edit type to "Fields" and enter the desired field type (see below) and command in the field's string gadget. The field type must be the first text in the string gadget, and capitalized.

To execute a field type:

- Use the

Edit->Execute
 menu item to execute the field of the record

displayed in ShimmerDB's
 title bar

.

- When editing records, activate the desired field's string gadget and press <CTRL> <RETURN>.
- Switching the edit type to "Text" will display multi-line text for the record and field displayed in the title bar.

Executing a field that does not have a valid command switches to editing multi-line text (Edit type "Text").

Field Types:

- EXEC - Executes the rest of the line as an AmigaDOS command. Does not wait for the program to return.
- SYNCEXEC - Like "EXEC", only ShimmerDB will accept no more input until the executed command ends (synchronous execute).
- OUTPUT - Similar to "SYNCEXEC", "OUTPUT" executes the line as a command and waits for it to return. Any output from the executed command is stored in the specified field. Output may be stored in the field's string gadget, or as multi-line text (either overwriting, "-", the current text, or being inserted before, "<", or after, ">", it).
 - OUTPUT <fieldname>| - The first non-blank line generated by the command is stored in <fieldname>.
 - OUTPUT-<fieldname>| - Replaces the current multi-line text of <fieldname> with the output of the executed command.
 - OUTPUT<<fieldname>| - The executed command's output is inserted above <fieldname>'s multi-line text.
 - OUTPUT><fieldname>| - Command output is appended to <fieldname>'s text.

If <fieldname> is omitted, output goes to the field executing the "OUTPUT" command.

Control Sequences:

Special control sequences may be inserted into the command strings.

- %d - This inserts the current record's data for this field into the command string.
- %t - This inserts the current record's title (name) into the command string.
- %f<field name>| - Inserts the current record's data for <field name> into the command string.

Examples:

- Field Name	Field Data	Record Data
-----	-----	-----
Picture	EXEC SYS:Utilities/Viewtek %d	PICS:JPEGs/Wolverine.004.jpg

Result: "%d" is replaced with "PICS:JPEGs/Wolverine.004.jpg" and Viewtek is run.

- Field Name	Field Data	Record Data
-----	-----	-----

```

Player      EXEC %d %fPath|%fFile          C:Sound
Path        SYS:Music/Sounds/
File        ObeyDaleks.ss

```

Result: "%d" is replaced with "C:Sound", which is then executed with the argument "SYS:Music/Sounds/ObeyDaleks.ss".

```

- Field Name  Field Data          Record Data
-----
Picture      SYNCEXEC C:FastJPEG %fPath|%d    Storm.003.jpg
Path        PICS:JPEGs/

```

Result: "%fPath|" is replaced with "PICS:JPEGs/".
"%d" is replaced with "Storm.003.jpg".
C:FastJPEG is run with the argument "PICS:JPEGs/Storm.003.jpg".
ShimmerDB will wait until FastJPEG ends.

```

- Field Name  Field Data
-----
Say Hi       OUTPUT| Echo "Hi!*nWhat's new?"

```

Result: Field "Say Hi" will receive the string "Hi!"

```

- Field Name  Field Data
-----
Say Hi       OUTPUT-ReadMe| Echo "Hi!*nWhat's new?"
ReadMe      TEXT

```

Result: Field "ReadMe"'s multi-line text will read:
Hi!
What's new?

1.8 ShimmerDB: Main Interface

ShimmerDB's main window resembles the following:

```

-----
|o|
      Record: A. Rexx  Field: Address
      |'|#|
|-----|
|Pos:
      2
      / 3  File:
      RAM:SmegDB
      |
|
|Edit:
      @|Records
      -----|
|-----| Name

```

```

      |||
|
      Q. Biggles
|| |Address

      |||
|
      A. Rexx
|| |Phone

      |||
|
      D. Lister
|| |
|| |
|| |
|-----|
      | o |
      |
|
      -----|
|
      @|  Add

      Del

      |o |
      |
-----
|
      |,|
-----

```

1.9 ShimmerDB: Menus

Menus:

Project

Edit

Search

Sort

Settings

Project

Edit

Search

Sort

←

	Settings			
New	Execute	Set Pattern...	Sort	Insert/Append »
~~~~~	~~~~~	Find Next	~~~~~	~~~~~
Open...	Select All	Find Prev	./ Ignore Case	./ Create Icons?
Insert...	Select None	~~~~~	./ Sort By Title	
Append...	Toggle	Ignore »	Only Selected	
~~~~~	~~~~~	Scan Only »	~~~~~	
Save	Cut		Reverse	

```

Save As...  Copy
~~~~~
Paste
About...   Unit      »
~~~~~
Quit       Edit      »
          Mode     »
          ~~~~~
          Undelete
          Flush
          Flush All

```

1.10 ShimmerDB: Hotkeys

HotKeys

Keyboard hotkeys are unavailable when a string gadget is active; except for Tab-cycling and command-execution.

Menu shortcuts from within string gadgets may work, but may also result in extra data being entered, or may be intercepted by the string gadget itself. A little cleanup is needed here.

Main Window Hotkeys:

- <Tab> - Cycle to next string gadget.
- <Shift><tab> - Cycle to previous string gadget. A little more work is needed here.
- <Return> - From a data string gadget, cycle to next string gadget.
- <Shift><Return> - Cycle to previous string gadget.
- <Return> and <Shift><Return> currently work better than Tab cycling.
- <CRSR UP> - Selects previous item in listview.
- <CRSR DOWN> - Selects next item in listview.
- Multiple items may be selected by holding down the <Shift> key.
- f - Cycle forwards through field list.
- <Shift> f - Cycle backwards through field list.
- r - Cycle forwards through record list.
- <Shift> r - Cycle backwards through record list.
- e - Cycle forwards through the edit types (Fields, Records, Text).
- <Shift> e - Cycle backwards through the edit types (Fields, Records, Text).
- m - Cycle forwards through the edit modes (Add, Edit, Insert...).
- <Shift> m - Cycle backwards through the edit modes (Add, Edit, Insert...).

When the last/first item is reached, will wrap around to the first/last item.

Search Requester Hotkeys:

- f (or F) - Find the next occurrence of the current search pattern.
- n (or N) - Find the previous occurrence of the current search pattern.

Menu Shortcuts:

Listed in menu descriptions. Some may be changed in future versions.

1.11 ShimmerDB: Search Requester

The search requester, or a reasonable facsimile thereof:

```

-----
|o| Search                                     |'|#|
|-----|
|
|           *nn*
|           ||
||                                     |||
|-----|
|
|           Ke??i K*
|
|           Next
|           Find
|           Previous
|
|--Ignore----- ----Scan Only-----|
||
|
|           Case | |
|
|           Titles      ||
||
|           Titles| |
|
|           Selected Items ||
||
|           Text | |
|
|           Selected Fields||
|-----|
|-----|
|
|           Clear Pattern
|
|-----|
|                                     |,|
-----

```

1.12 Menu: Project

- Project:
- New - Clears the database.
HotKey: <R. Amiga> N
- ~~~~~
- Open - Loads a new database. If the
Edit Type
is "Text" an ASCII text
file may be loaded.
Hotkey: <R. Amiga> O
- Insert - Inserts new data into the current database. New records are
inserted above the current record. New fields (if used) are
appended to the end of the field list.
See the
Settings->Insert/Append
menu for options relating to how
the new data is handled.

If the
edit type
is "Text", a requester will appear asking if
you wish to insert new data into the current database, or ASCII
text into the current text.
Hotkey: <R. Amiga> I
- Append - Appends new data into the current database. New records and
fields will be appended to the end of their respective lists.
See the
Settings->Insert/Append
menu for options relating to how
the new data is handled.

ASCII text may also be appended if editing multi-line text.
Hotkey: <R. Amiga> +
- ~~~~~
- Save - Saves the database using the current file name (unless "unnamed").
Hotkey: <R. Amiga> S
- Save As - Brings up a file requester to save the database. If the
Edit Type
is "Text" then just the text may be saved as an ASCII
file.
Hotkey: <R. Amiga> A
- ~~~~~
- About - Guess. :-)
Hotkey: <R. Amiga> ?
- ~~~~~
-

- Quit - Another toughie. Will ask for verification if your work hasn't yet been saved.
Hotkey: <R. Amiga> Q

1.13 Menu: Edit

- Edit:
 - Execute - Takes the appropriate action (or inaction) depending on the field type (see field types for more info.).
Hotkey: <R. Amiga> E
- ~~~~~
- Select All - All titles are selected.
Hotkey: <R. Amiga> L
 - Select None - All titles are deselected.
Hotkey: <R. Amiga> 0
 - Toggle - Selected titles aren't, & unselected titles are. :-)
Hotkey: <R. Amiga> =
- ~~~~~
- Cut - Moves selected items to the clipboard.
Hotkey: <R. Amiga> X
 - Copy - Copies selected items to the clipboard.
Hotkey: <R. Amiga> C
 - Paste - Inserts the contents of the current clipboard unit. Contents are inserted above the currently selected item.
Hotkey: <R. Amiga> V
 - Unit - Selects the current clipboard unit (0 - 10) to use for cutting, copying, and pasting.
- ~~~~~
- Edit - Chooses the Edit Type
 ; just like the Edit cycle gadget For those who don't use CycleToMenu (why?! :-)).
 - Mode - Selects the Edit Mode
 ; same as the gadget. See Edit Modes for more info.
- ~~~~~
- Undelete - The last item deleted is restored, & placed after the current
-

item. An index (".001" ...) will be appended to the restored item in the event another item shares the restored item's name.
Hotkey: <R. Amiga> U

- Flush - Clears the undelete buffer of the current Edit Type (each Edit Type has its own undelete buffer).
- Flush All - Clears ALL undelete buffers.

1.14 Menu: Search

Search:

- Set Pattern - Brings up the search requester to set the current search pattern.
Hotkey: <R. Amiga> P
- Find Next - Finds the next occurrence of the current search pattern.
Hotkey: <R. Amiga>]
- Find Prev - Finds the previous occurrence of the current search pattern.
Hotkey: <R. Amiga> [

~~~~~

Ignore:

- Case - Searches are case insensitive ("A" = "a") if this item is checked.
- Titles - For records only. The title of a record is ignored; only its data is searched for the current pattern.
- Text - For records only. If checked, multiline text (field type "TEXT") is ignored during searches.

Scan Only:

- Titles - For records only. When checked, only a record's title is matched against the search pattern.
- Selected Items - Only selected items are searched.
- Selected Fields - For records only. Only the fields that are selected (in Edit: Fields) are matched against the search pattern if this item is checked. Otherwise all fields of a record are scanned.

## 1.15 Menu: Sort

Sort:

- Sort - Sorts the current Edit Type list.  
Hotkey: <R. Amiga> T

~~~~~

- Ignore Case - "A" = "a". This slows the sort somewhat.
- Sort By Title - Only for records. The list of records is sorted by title when this menu item is checked. Otherwise the list is sorted using the current field.
- Only Selected - Do you wish to sort all items, or just the selected ones?

~~~~~

- Reverse - Reverses the order of the current Edit Type list. Takes the "Ignore Case" and "Only Selected" menu items into account.  
Hotkey: <R. Amiga> R

## 1.16 Menu: Settings

Settings:

Insert/Append:

- Ignore New Fields? - No new fields will be added to the field list.

These options tell ShimmerDB what to do if it encounters a record with the same name as an existing record.

- Use Old Record - The new record is ignored.
- Use New Record - Data from the old record is discarded and replaced with data from the new record.
- Index New Record - The new record will be renamed to something like <record name>.001.

~~~~~

- Create Icons? - If checked, a database is saved with a project icon. The current dimensions of ShimmerDB's GUI (width, height, etc.) are also saved as tooltypes

1.17 ShimmerDB GUI: Title Bar

```
-----
|o| Record: A. Rexx  Field: Address      |'|#|
|-----|
```

The title bar displays the current record and field.

1.18 ShimmerDB GUI: Position Gadget

```
Pos: | 2 | / 3
-----
```

Displays your position, and the total number of items, in the current list. You may jump to any position by entering a number in the integer gadget.

1.19 ShimmerDB GUI: Filename Gadget

```
File: |RAM:SmegDB |
-----
```

The path and name of the database file you're editing.

1.20 ShimmerDB GUI: Edit-Type Cycle Gadget

```
Edit: |@|Records|
-----
```

Selects which list type to edit. Possible choices:

- Fields - Fields are the building blocks of any database. They are the categories each object in your database possesses. For instance, if you're creating an address book you would have fields such as "Name", "Address", "Phone", "E-Mail", etc. Fields may also hold multiple lines of text, and be set to execute commands. See field types for more information.
- Records - A record is a collection of fields. Using the address book example, each person would be a record, and have its own "Name" field, "Address" field, etc.
- Text - Multiple lines of text may be entered in the Edit listview gadget. Such text is associated with the record and field displayed in the title bar.

Note:

- Field and record names are case-sensitive. "ReadMe" and "readme" are seen as different names.
- If you enter a field or record name that has already been used, an index will be appended to it. For example, continually entering "readme" would result in the names: "readme", "readme.001", "readme.002", etc.

Hotkey: e (cycle forwards),
 <Shift> e (cycle backwards)

1.21 ShimmerDB GUI: Edit Listview

```

| Name          ||
| Address       ||

```

The edit listview displays the the titles of fields and records, and also multi-line text.

Multiple items may be selected by holding down the <SHIFT> key.

1.22 ShimmerDB GUI: Edit String Gadget

```

|D. Lister      |
-----

```

Data entered here will be added to the
 edit listview
 . The way it is added
 depends on the current
 edit mode
 .

Pressing <TAB> switches the cursor to the first field string gadget.

1.23 ShimmerDB GUI: Edit Mode Gadget

```

|@| Add |
-----

```

The edit mode gadget affects how mouse clicks in the edit listview or text entered in the edit string gadget will be processed.

Edit modes:

- Add - Any text entered in the edit string gadget is appended to the end of the current list.
 Duplicate field and record names will have an index appended to them (e.g. E-Mail Address.001)
- Edit - The selected item's name will be copied to the edit string gadget where it may be edited.
- Insert - Like "Add", only the text entered will be inserted above the currently selected item.

- Duplicate - Any items clicked on in the edit listview will be duplicated, and an index will be appended (unless editing text).

NOTE: Only the item's name is duplicated. Duplicating a record, for example, will not also duplicate any data contained within the record.

- Move Above - Moves selected items in the edit listview above an item. Select an item (or items) in the edit list view, and then click where (in the listview :-) you wish them to be moved too. The selected item(s) will be moved above the entry clicked on.
- Move Below - Works the same as "Move Above", only now the selected items will be located below the entry clicked on.

Hotkey: m (cycle forwards),
<Shift> m (cycle backwards)

1.24 ShimmerDB GUI: Delete Gadget

```
_____  
|Del|  
----
```

The delete button will delete all items selected in the edit listview.

Use the Edit->Undelete menu option to retrieve deleted data. Each list maintains its own undelete buffer (currently 100 items). Once the undelete buffer is full, the oldest item will be discarded (& gone for good) to make room for each newly-deleted item.

1.25 ShimmerDB GUI: View Slider

```
_____  
| o |  
-----
```

This slider gadget determines how much of the horizontal space in ShimmerDB's window is occupied by the edit listview and the field views (oh, just try it and see for yourself :-)).

With the knob all the way to the left, the field views will get the lion's share of the available window real estate. All the way to the right gives the nod to the edit listview. Somewhere in between lies perfection (perfection's a liar?! Hmmm, could be... (How many other program docs tackle the tough philosophical issues, eh? :-) (Erm, now you know why. :-))).

1.26 ShimmerDB GUI: Field Data String Gadget

```

Name   | Ima Goof   |
-----

```

The data for each field is entered here. If editing records, pressing <CTRL> <RETURN> will execute the field type (see field types).

<TAB> cycles between the various field data string gadgets and the edit string gadget.

1.27 ShimmerDB GUI: Field Slider

```

| o     |
-----

```

Divides horizontal space in field view, in the same manner as the view slider. Sets how much space is devoted to the field names and field data.

1.28 Search GUI: Set Pattern

```

| B?gg*   |
|         |
|-----|
| *ll*    |
|-----|

```

Enter a new search pattern in the string gadget, or retrieve a pattern from the listview. The pattern displayed in the string gadget is the current search pattern.

Search patterns may contain AmigaDOS pattern matching wildcards

1.29 Search GUI: Find Next Button

```

| Next |
-----

```

Finds the next occurrence of the current search pattern.

Hotkey: N

1.30 Search GUI: Find Previous Button

```
|Previous|
|-----|
```

Finds the previous occurrence of the current search pattern.

Hotkey: P

1.31 Search GUI: Ignore Options

```
                --Ignore---
||  | Case  |
||  | Titles|
||  | Text  |
|-----|
```

Ignore:

- Case - Searches are case insensitive ("A" = "a") if this item is checked.
- Titles - For records only. The title of a record is ignored; only its data is searched for the current pattern.
- Text - For records only. If checked, multiline text (field type "TEXT") is ignored during searches.

1.32 Search GUI: Scan Options

```
-----Scan Only-----
||  | Titles      |
||  | Selected Items |
||  | Selected Fields|
|-----|
```

Scan Only:

- Titles - For records only. When checked, only a record's title is matched against the search pattern.
- Selected Items - Only selected items are searched.
- Selected Fields - For records only. Only the fields that are selected (in Edit: Fields) are matched against the search pattern if this item is checked. Otherwise all fields

of a record are scanned.

1.33 Search GUI: Clear Pattern List

```
|Clear Pattern|
-----
```

Clears all search patterns in the
search pattern listview
.

1.34 AmigaDOS Pattern Matching

A brief summary of important AmigaDOS pattern matching wildcards:
(What, you don't already know these?! :-)

- #? (or "*", if enabled) - Match any characters.
- ? - Match a single character.
- #<p> - Match zero or more occurrences of <p>.
- ~(<p>) - Do not match <p>.
- (<p1>|<p2>) - Logical OR - Match <p1> or <p2>
- (<char>-<char>) - Set a character range (e.g. (A-E))

1.35 ShimmerDB: Limits

Maximum number of:

- Fields,
Records,
Lines of multi-line text,
Search patterns: 65,500 *, **
- Characters in a string gadget: 256 **, ***
- Items in undelete buffer: 100 **
- Clipboard buffers: 10 (units 0 - 9) **

* Also limited by the amount of free memory.

** Arbitrary. This may be expanded or user definable in future versions.

*** If ASCII text is loaded, lines with more than 256 characters are truncated. Future versions may/should split the line.

1.36 ShimmerDB: File Format

File Format:

ShimmerDB saves databases in an IFF format consisting of three FORMS and three chunks. Note: This format is still under construction (as is it's explanation :-)).

Forms:

- SHDB - The main wrapper.
- SHFD - Contains information describing one field.
- SHRC - Describes one record. May contain many SHFD chunks.

Chunks:

- SHDR - Header chunk containing two long words: the number of fields in the database, and the number of records. There should only be one SHDR in a FORM SHDB.
Note: ShimmerDB doesn't actually use this chunk yet. It may one day be used, expanded, or completely disposed of.
- TITL - Title string of a field or record.
- DSTR - A field or record's data string.
- TEXT - Standard TEXT chunk containing ASCII data. Used for multi-line text.

A typical file:

```

FORM SHDB
  SHDR          - Header chunk.

  FORM SHFD     - These SHFD chunks describe
  TITL          the database's fields.
  DSTR

  FORM SHFD     - Field title.
  TITL          - Field data (optional).
  DSTR

  FORM SHFD
  TITL
  DSTR

  FORM SHRC     - Now the records...
  TITL          - The record's name.
  FORM SHFD     - Records may contain many fields, or none.
  TITL          - Which field.
  DSTR          - The field's data.
  FORM SHFD
  TITL
  TEXT          - This field contains multi-line text, but no DSTR.
FORM SHRC
  TITL          - This record has no data.
FORM SHRC
.
.
.          - ...And so on.
```

1.37 ShimmerDB: Clipboard

ClipBoard Information:

When data is copied from a database to the clipboard it is stored as an IFF CAT CLIP containing two FORMs. The first FORM will be ShimmerDB's native

FORM SHDB
, the second FORM FTXT.

Within the FORM FTXT will be multiple CHRS chunks. Assuming a record was copied, the clipboard data will appear as:

```
FORM FTXT
CHRS - Record name.
CHRS - Field name (if the field holds any data).
CHRS - Field data.
CHRS - Multiline text (if present).
.
.
. - and so on.
```

When data is copied from the clipboard into a database ShimmerDB will first look for a FORM SHDB. If one is found, then data is inserted with fields being indexed if the name already exists, and records obeying the options selected in the

Settings->Insert/Append
menu. I'm not

satisfied with this behavior; it will probably change in future versions.

If no FORM SHDB is found, then FORM FTXT is searched for and used. Any FTXT data is used as field or record names (if the edit type selected is "Fields" or "Records").

CAT CLIP

FORM SHDB

<whatever...>

Note: The SHDR chunk will most likely be inaccurate. This will either be fixed in a future version, or the SHDB chunk will be abandoned totally.

FORM FTXT

<CHRS chunks...>

1.38 ShimmerDB: Known Bugs

Bugs:

At present, all known bugs have been corrected (or so they'd like me to believe...).

If you find a bug, please report it. When reporting bugs, be as descriptive as possible about the problem, and what steps lead to it. Details of your system (CPU, amount of RAM, OS version, other programs running at the time, etc.) would also be useful.

See

The Author
for mailing addresses.

1.39 ShimmerDB: 'There's somethin' screwy goin' on here..'

Oddities:

These aren't serious enough to be called bugs, but they could use some improvement.

- Tab cycling between string gadgets could be smoother. <Return> and <Shift><Return> work much better.
- Using menu shortcuts from within string gadgets is a little dodgy.
- In search patterns, maybe leading and trailing "*" wildcards should be implied to find partial strings? Judgement call; right now I'm thinking, "no."
- Clipboard pasting isn't as smooth as it could/should be.
- Clipboard operations haven't been extensively tested yet; there may be bugs here (why not, they've crept up everywhere else :-)) - let me know.
- Save often in low memory situations; ShimmerDB's default behavior is to exit (safely) when memory allocation fails. This will be improved.
- These docs (and you're reading the `_improved_` version). :-)

1.40 ShimmerDB: The Next Generation

Things to do (listed in no particular order):

- ARexx interface. *
 - Handle multiple projects/views.
 - Printing.
 - Localization.
 - Hide user-specified records and/or fields.
 - An auto-executing field type. Maybe two or three (execute whenever the current field, a specified field, or any field is modified).
 - More field-type control sequences. Let me know what you'd like to see.
 - Interface enhancements (some things worked better in theory than in practice). Possible rewrite using MUI or BGUI.
 - Find bugs, and replace them with newer, more interesting bugs. *
 - Import/export FTXT.
 - Power Packer/XPK support.
 - More, and more descriptive, error messages/requesters.
 - Maybe use style guide compliant "FONTNAME", "FONTSIZE" rather than DiskMaster-style /<size>. Depends on how many complaints I receive...
 - More editing features when dealing with multi-line text.
 - Install my own string gadget EditHook for clipboard and menu shortcut support. Maybe...
 - Recursion - fields that contain entire databases of their own.
 - Preferences file.
 - User-defined menu(s) to launch ARexx macros.
 - Raise limits to two to four billion items per list.
 - Improved documentation. *
 - AppWindow/AppIcon support.
-

- Improve behavior in low memory situations. *

* Priority.

1.41 ShimmerDB: The Early Years

History:

Millions of years ago, dinosaurs roamed the earth...

- 0.x - Private Beta versions. Primordial ShimmerDB. :-)

- 1.0 - First public release.

1.42 ShimmerDB: The Name Game

Q: "Why is this program named 'ShimmerDB'?!"
"What the @#%& were you thinking?"

A: Hmmmm, I wonder... I could explain how ShimmerDB came to be called "ShimmerDB", but it might be more fun for you to tell me. :-) So e-mail your best guesses to me along w/ the inevitable bug reports (<sigh>), and maybe I'll include them in future docs. :-) ...Heck, I may even use them instead of the
real story
. :-D

1.43 ShimmerDB: The Shocking Truth

A main goal of ShimmerDB was to have as many of the program's features as possible accessed by the same interface, as opposed to separate requesters for defining fields, entering records, etc. I felt this would be easier for the user to learn and get to grips with, and I knew it would be easier for me than programming 50 different requesters (laziness is the mother of invention). :-)

So now I needed a name that would match the program's polymorphic nature. "Sybill" came to mind, but I felt it was a bit too dark/disturbing, and I think it's been used before (it sounds vaguely familiar...). Then I remembered the "Shimmer" sketch from Saturday Night Live:

"New Shimmer is a dessert topping!"

"It's a floor wax!"

"Hold on folks, new Shimmer is a dessert topping and a floor wax!"

"Shimmer, the world's first non-dairy floor wax."

"Perfect!", says I. One "DB" (Data Base) suffix later, and some programming, and here we are.

1.44 ShimmerDB: Credit Where Credit is Due

Many thanks to the following:

- God - Life, the universe, and everything.
- Wouter van Oortmerssen - Amiga E. Programming ShimmerDB in any other language would have been possible, but not nearly as easy or as much fun. Also for the EasyGUI module, which was used to prototype the search requester.
- Jan van den Baard - GadToolsBox. Although I coded the interface by hand (I needed the practice/experience), GTB was very handy in prototyping the main window.
- Matt Dillon/Aaron Digulla - DME/XDME.
- Heinz Wrobel - HWGRCS.
- Greg Donner - Initial program ideas and beta testing.
- My cat - Various programming and interface solutions. "Sleep on it," being one of many valuable lessons.
- Phil Wright,
Ron Czarnecki,
Marc Stump,
Gary Willmot - Beta testing.
- Blondie,
Deborah Harry,
Ramones,
Pete Townshend,
The Rolling Stones,
the Police,
Animal Logic - Music to program by.

1.45 ShimmerDB: Beware of Romulans Bearing Gifts.

Some suggestions if you'd like to support ShimmerDB:

- Money - Cash or checks in US funds - Always useful. :-)
 - Amiga software - Nothing pirated, please.
 - CD-ROMs or disks - Graphics, PD.
 - Comic books - Make mine Marvel. Especially Spider-Man, Wolverine, Captain America, and X-Men.
 - Trading cards - Superheroes, sci-fi, fantasy...
 - Music CDs or cassettes - Blondie, Deborah Harry, Ramones, U2, The Rolling Stones, '80's...
-

- Videos (NTSC) - Sci-Fi, comedy, monster movies (Godzilla, etc.), oldies (Cary Grant, Jimmy Stewart, Humphrey Bogart...)...

1.46 ShimmerDB: The Author

Feel free to send comments, bug reports, suggestions, and/or gifts to:

Jody J. Tierney
29118 Lark Street
Elkhart, IN 46514-9579
USA

E-mail: 102673.1743@compuserve.com (Greg Donner will relay any messages)

- Fido: Um, it's kind up in the air right now.

- Pro-Net BBS - (219) 825-9164
Handle: Grey Wanderer

- Panoptic Network BBS - (219) 674-5292
674-9250
Handle: Grey Wanderer