

DATEBOOK

COLLABORATORS

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	DATEBOOK	1
1.1	main	1
1.2	legal_stuff	1
1.3	system_desc	2
1.4	cli_usage	3
1.5	reports	5
1.6	report_window	5
1.7	upcoming_events	6
1.8	month_mode	6
1.9	day_mode	7
1.10	cli_output	7
1.11	record_editor	8
1.12	ed_gadgets	8
1.13	ed_menus	9
1.14	menus_project	10
1.15	menus_mode	10
1.16	menus_edit	11
1.17	menus_move	11
1.18	menus_fkeys	12
1.19	memocal	13
1.20	mc_gadgets	13
1.21	mc_menus	15
1.22	mc_menus_project	15
1.23	mc_menus_mode	16
1.24	mc_menus_report	16
1.25	mc_menus_move	17
1.26	rexx_serv	17
1.27	rx_dbase	18
1.28	rx_opendb	18
1.29	rx_savedb	18

1.30 rx_sortdb	19
1.31 rx_record	19
1.32 rx_addrecord	19
1.33 rx_deleterecord	20
1.34 rx_nextrecord	20
1.35 rx_prevrecord	20
1.36 rx_gotorecord	21
1.37 rx_getfield	21
1.38 rx_setfield	22
1.39 rx_search	22
1.40 rx_findrecord	23
1.41 rx_nextmatch	23
1.42 rx_prevmatch	23
1.43 rx_reports	24
1.44 rx_summary	24
1.45 rx_events	24
1.46 rx_today	25
1.47 rx_upcoming	25
1.48 rx_date	26
1.49 rx_string2date	26
1.50 rx_getmonthname	26
1.51 rx_dayname	27
1.52 rx_expanddate	27
1.53 rx_isvalid	27
1.54 rx_gencontrol	28
1.55 rx_cngmode	28
1.56 rx_iconfy	29
1.57 rx_shutdown	29
1.58 iconify	30
1.59 appicon	30
1.60 clock	30
1.61 cl_menus_project	30
1.62 cl_menus_mode	31
1.63 calendar	32
1.64 db_help	32
1.65 configuration	32
1.66 convert	34
1.67 record_desc	34
1.68 date_field	35

1.69	time_field	35
1.70	pri_field	35
1.71	event_field	36
1.72	cycle_field	36
1.73	warning_field	37
1.74	freq_field	37
1.75	rexx_field	37
1.76	notes_field	39
1.77	installation	39
1.78	future	40
1.79	credits	41
1.80	history	41
1.81	about_cron	42
1.82	about_arexx	42
1.83	dun_dat	43

Chapter 1

DATEBOOK

1.1 main

DateBookIV (V37.30)

By Robert Hardy © 1988 - 96 HardKore Software

Internet: Robert_Hardy@dork.tvinet.com

WWW: <http://www.tvinet.com/~robert>

Fido: 1:153/250.0

i) Contents

1	
	Legal Stuff
	4
	Installation
	2
	System Description
	5
	Future
	3
	Record Description
	6
	Credits

1.2 legal_stuff

1) Legal Stuff

1.1) Disclaimer

DateBookIV is provided "AS IS", WITHOUT ANY WARRANTY to its

quality, performance or fitness for a particular purpose. In no event shall the author be liable or responsible to the user or any other person, for any kind of damage caused by the use of this software.

1.2) Conditions of Use and Distribution

DateBookIV is MessageWare, if you like and use DateBook send me a message (Internet or Fido). Feel free to include complaints or suggestions.

DateBookIV may be freely distributed provided:

1. The files are ALL left intact and unmodified.
2. No charge is made for DateBookIV (other than a reasonable copy charge).
3. DateBookIV is not packaged as a part of a commercial product without prior written consent.
4. DateBookIV may not be uploaded to any electronic service that claims a copyright to it's files and programs.

1.3) System Requirements and Limitations

DateBookIV requires AmigaDOS V2.04 or higher and 512K memory (~200K Free).

Battery backed clock or manual setting of the system time.

1.3 system_desc

2) System Description

DateBookIV is a reminder system that can be used to help you avoid the embarrassment of forgotten appointments, birthdays, etc. It provides numerous ways to automatically maintain permanent events. With this program you can specify how far ahead you want to be reminded of an event and how often. Reports can also be generated for a specific day or month.

Various reports or queries are available either to the standard output (shell), the graphic user interface or through the rexx port.

In the Record Editor, the Function Keys (F1 - F10) can execute Rexx macros or Rexx command strings. The Function Key Definition File will be loaded on startup, but if no name is supplied, then DateBook will use 'DateBook.dfk'. The search order for the Function Key Definition File is the current directory first, then S:, and finally PROGDIR:.

Various configuration options are normally loaded on startup. If no name is supplied DateBook will use 'DateBook.dfg'. The search order for the configuration file is the current directory first, then the contents of ENV:DateBook/StartDFG, then S: and finally PROGDIR:.

An

ARexx

command may be added to an event to allow more flexible announcement methods.

The file

'Dates.dat'

included, contains some common Canadian and

US holidays. If you feel that I'm missing some please EMAIL me your suggestions, I'm particularly interested in 'International xxx Day' type of dates. In any case, please include how they are determined each year, ie: yearly, second monday in november...

2.1

CLI Usage and WorkBench Tool Types

2.2

Reports and Queries

2.3

CLI Output

2.4

Record Editor

2.5

MemoCal Mode

2.6

Rexx Server

2.7

Iconify Mode

2.8

Help

2.9

Configuration Requester

2.10

ConvertDateBook

1.4 cli_usage

2.1) CLI Usage and WorkBench Tool Types

CLI arguments and Workbench Tool Types are exactly the same. The '=' is optional from the CLI.

```
DateBook MODE=MEMO|EDIT [FILE=<datefile>] [PUBSCREEN=<name>]
[FKEYS <filename>] [RXPORTE=<port_name>] [CONFIG <name>]
```

Or

```
DateBook MODE=CLI [DATE=<date>] [FILE=<datefile>]
[WAIT] [QUIET] [PRINT]
```

Or

```
DateBook MODE=SERVER|ICON|CLOCK|CAL [RXPORTE=<port_name>] [CONFIG <name>]
```

Where :

- FILE - Name of file to examine. If not specified, 'S:Dates.dat' will be used. If the file specified does not exist in the current directory, 'S:' is checked.
- PUBSCREEN - Name of public screen to open our window on.
- FKEY - Name of function key bindings file to load.
- RXPORT - Name of the Rexx port. Default is 'datebook_rx'
- CONFIG - Specifies the name of the configuration file to load on startup.
- MODE - Set the initial mode.
- MEMO - Start up in MemoCal mode (default)
- EDIT - Start up in Record Editor mode.
- SERVER - Start up in Rexx server mode, no GUI.
- CLI - Simply generates the desired report to the CLI window.
- ICON - Startup as an AppIcon on Workbench.
- CLOCK - Startup as a Clock/Memory meter.
- CAL - Startup as a Clock/Memory meter with a calendar of the current month.

NOTE: The remaining arguments are only valid when CLI mode is set.

- DATE - Date to report. This mode only prints events for specified day or month and does no cycling/deleting. If you specify only a month name, events for that month will be displayed.

You may specify Dates by typing the following:

- 1-jan-87 - Change the day, month and year
2-feb - Change the day and month only.
3 - Change the day only
871231 - Change year, month and day
1231 - Change month and day
31 - Change day only

The following set the date relative to today's date. You only need to type in the first 2 or three letters.

Today	Sunday	Thursday
Tomorrow	Monday	Friday

Yesterday Tuesday Saturday
 Wednesday

- WAIT - Wait for a response before finishing. This is always set when launched from WorkBench.
- PRINT - Echo date events to the printer as well.
- QUIET - Prints nothing if there are no events to display.

See the example Icons for more examples of Workbench tool types.

1.5 reports

2.2) Reports and Queries

DateBookIV's report generator also manages cycling or deleting expired events. It will report events for a specific day or month, or simply for upcoming events in the near future.

2.2.1

- Report Window
- 2.2.2
- Upcoming Events
- 2.2.3
- Month Mode
- 2.2.4
- Day Mode

1.6 report_window

2.2.1) Report Window

Reports are presented in a list view. Normal events are displayed with Pen #2 (In the standard WB palettes this pen is usually white). When an 'Upcoming Events' report is generated, expired events are displayed with Pen #3 (which is Blue in THE DEFAULT palette).

A page icon at the left of an event indicates that there is a note associated with the event. Double clicking on this event will display the contents of the

Notes Field

.

Reports may be printed or saved to a file by using the Print button on the report window.

2.2.1.1 Print Requester

Before printing a report you may set various printer parameters. These are as follows:

Print Quality, Pitch and Line Spacing which are read from Preferences on startup.

The state of the 'No Form Feed'/'Add Form Feed' gadget allows you to choose whether or not to print a form feed at the end of the report.

'Printer'/'File' allows you to set the destination for the report. If you select file, a file requester will open when you select 'Print'.

1.7 upcoming_events

2.2.2) Upcoming Events

DateBookIV searches through the database and checks for upcoming events. An event is displayed when the difference between the current date and the event's date is within the specified warning period. Records with no date or that have already passed are deleted, unless the Cycle field is valid, see section 3 for a complete explanation of the

```
CYCLE
    ,
    WARNING
    and
    FREQUENCY
    fields.
```

If the current date matches an event's date and the
REXX
field
is valid (not empty), the rexx field will be executed.

EG:

```
1> Datebook MODE cli
```

```
Upcoming On: Mon Dec 11 1995
```

```
    Mon Sep 18 1995  Gather Financial data for Donna
    Tue Dec 12 1995  Panorama Executive Meeting
    Tue Dec 12 1995  Dad's 73rd Birthday
19:30 Wed Dec 13 1995  Panorama WorkShop Meeting
```

1.8 month_mode

2.2.3) Month Report

DateBookIV searches through the data file and builds a report

containing all events that fall within the specified month. From MemoCal mode, this report can be generated by selecting 'Summary'.

EG:

```
1> Datebook MODE cli DATE jan
```

Summary For: January 1996

```
19:30 Tue Jan 02 1996 Panorama General Meeting
09:00 Wed Jan 03 1996 Chiropractor Appointment
09:40 Wed Jan 03 1996 Doctors Appointment
      Tue Jan 09 1996 Panorama Executive Meeting
19:30 Wed Jan 10 1996 Panorama WorkShop Meeting
      Mon Jan 15 1996 Pay Day!!
      Wed Jan 31 1996 Pay Day!!
```

1.9 day_mode

2.2.4) Day Mode

DateBookIV searches through the data file and builds a report containing all events that fall on the specified day. From MemoCal mode, this report can be generated by selecting one of the day gadgets or the Today gadget.

EG:

```
1> Datebook MODE=cli DATE=tomorrow
```

Upcoming On: Upcoming On: Up

```
      Tue Dec 12 1995 Panorama Executive Meeting
      Tue Dec 12 1995 Dad's 73rd Birthday
```

1.10 cli_output

2.3) Command Line Output

By using 'MODE=CLI' on the command line or tool type, the GUI is bypassed and the desired report is printed to the output window.

For example:

```
1> run DateBook MODE cli QUIET WAIT <>CON:10/10/640/120/DateBookIV/AUTO
```

If there are any events to display, a console window is opened and waits for a keypress.

1.11 record_editor

2.4) Record Editor

This mode allows you to enter, edit, delete, sort and search for DateBook events.

The gadgets and pulldown menus should be mostly self explanatory (famous last words :-). Press the underlined key on the gadget for the desired function or just click on the gadget. Cursor left and cursor right will step forward and backward through the file. Cursor up, cursor down, or <RETURN> allows you to edit the current record.

In Change or Add mode, use the cursor, DEL and Back Space keys to move around and edit fields. The Clipboard is supported for line cut and paste. Check the pulldown menus for editing options and keyboard shortcuts.

Pressing the HELP key or selecting HELP! from the pulldown menu will open a help window (surprise!). DateBook is still active while the help window is open.

2.4.1

Gadgets

2.4.2

Menus

1.12 ed_gadgets

2.4.1) Record Editor Gadgets

Add

Adds a new blank record to the end of the database.
Starts edit mode on the Date field.

Duplicate

Adds a new record to the end of the database with the same contents as the current record. Starts edit mode on the Date field.

Delete

Blanks the current record. The Sort option or requesting and Upcoming report in MemoCal mode will remove the record from the database.

Sort

Sorts the database by date and time. Blank records are not copied to the new database. The original database

is renamed with a '.bak' extension.

|<= Beginning

Go to the first record in the database.

<==

Go to the previous record.

==>

Go to the next record.

=>| End

Go to the last record.

Find

Set up which field to search and the string to search for.
The search will match any record that contains the string
in the selected field.

Next Match

Find the next matching record.

Prev Match

Find the previous matching record.

Help

AmigaGuide help (This File).
The main DateBookIV window is still active while the help
window is open.

MemoCal

Switch to MemoCal mode.

1.13 ed_menus

2.4.2) Record Editor Menus

2.4.2.1

- Project
 - 2.4.2.2
- Mode
 - 2.4.2.3
- Edit
 - 2.4.2.4
- Move
 - 2.4.2.5

Function Keys

1.14 menus_project

2.4.2.1) Project

Open...

Open a different database.

Set/Save Config...

Change, save and load configuration.
See

Configuration Requester
HELP!...

AmigaGuide help (This File).

The main DateBookIV window is still active while the help window is open.

About...

Bragging:-). Also shows version numbers and contact info.

Quit

Exit the program. Same as Close Gadget.

1.15 menus_mode

2.4.2.2) Mode

MemoCal

Switch to
MemoCal
mode.

Rexx Server Mode

Switch to
Rexx Server
mode.

Clock

Switch to title bar clock/mem-meter iconification mode.

Calendar

Switch to Clock/Calendar iconification mode.

AppIcon

Switch to AppIcon mode.

Iconify

Switch to the selected iconification mode.

1.16 menus_edit

2.4.2.3) Edit

Change Record

Begin modifying the current record. Starts the cursor in the Date field. Pressing <RETURN>, Cursor up or down will have the same effect.

Add Record

See

Add
gadget description.

Duplicate Record

See

Duplicate
gadget description.

Delete Record

See

Delete
gadget description.

Sort

See

Sort
gadget description.

1.17 menus_move

2.4.2.4) Move

Next Record

See

==>
gadget description.

Previous Record

See

<==
gadget description.

First Record

See

|<=
gadget description.

Last Record

See

=>|
gadget description.

Find

See

Find
gadget description.

Next Match

See

Next Match
gadget description.

Previous Match

See

Previous Match
gadget description.

1.18 menus_fkeys

2.4.2.5) Function Keys

Define FKey...

10 function keys (F1 - F10). See the
Rexx Field
for a full description of the contents. When you select

Allows the user to bind (attach) a rexx string or macro to

one of the 10 function keys (F1 - F10). See the
Rexx Field

for a full description of the contents. When you select this item, the window title bar will prompt you to press a function key. After you have selected the function key, a string requester will be presented for you to enter or change the contents. All internal rexx commands are available.

EG: `''setfield $$d $C'`

This will insert today's date into the date field of the current record. Note the `''$$d'`, this is done so that `'$d'` will be passed on to the rexx command rather than expanding to the contents of the date field.

Load FKeys...

Load a set of function key bindings.

Save FKeys...

Save the current function key bindings using the current filename. If there is no current filename, a file requester will open.

Save FKeys As...

Save the current function key bindings to a new filename.

1.19 memocal

2.5) MemoCal Mode

The DateBook Memo Calendar was inspired by those used in electronic pocket organizers, and gives you a calendar-style view of a given month. This mode is the default, and is handy for getting a quick look at your overall appointment situation.

Each day of the month is a gadget. If any events fall on a given day, the number of events will be displayed in the top left corner of its gadget.

2.5.1

Gadgets

2.5.2

Menus

1.20 mc_gadgets

2.5.1) MemoCal Gadgets

Day Gadgets

If any events fall on that day, the number of events will be displayed at the top left corner of the gadget. A bar graph across the bottom of the gadget shows the time of any timed events. The dot at the centre represents 12:00 noon.

When selected, DateBook presents a report showing all events for that day or a requester explaining that there are no events for that day. In either case, an 'ADD' button is available to add an event to that day. If 'ADD' is selected, DateBook switches to Edit mode and inserts the appropriate date into the Date field for you.

See

Day Reports

.

Year (String Gadget)

Allows you to set the year displayed.

Month (String Gadget)

Allows you to set the current month. Only the first 2 or 3 characters are needed, as Datebook will figure out which month you mean and expand it out for you.

<==

Go to the previous month.

Now

Set the calendar back to the current month.

==>

Go to the next month.

Upcoming

Presents a report showing events coming in the near future.

See

Upcoming Reports

.

Summary

Presents a report showing events for the displayed month.

See

Month Reports

.

Today

Presents a report showing events for the current day or a requester explaining that there are no events for that day. In either case, an 'ADD' button is available to add an event. If 'ADD' is selected, DateBook switches to Edit mode and inserts the today's date into the Date field for you.

See

Day Reports

.

Help

AmigaGuide help (This File).

The main DateBookIV window is still active while the help window is open.

Editor

Switch to

Record Editor
mode.

1.21 mc_menus

2.5.2) MemoCal Menus

2.5.2.1

Project
2.5.2.2
Mode
2.5.2.3
Report
2.5.2.4
Move

1.22 mc_menus_project

2.5.2.1) Project

Open...

Open a different database.

Set/Save Config...

Change, save and load configuration.

See
Configuration Requester
Help!

AmigaGuide help (This File).

DateBook is still active while the help window is open.

About

Bragging:-). Also shows version numbers and contact info.

Quit

Exit the program. Same as Close Gadget.

1.23 mc_menus_mode

2.5.2.2) Mode

Editor

Switch to
Record Editor
mode.

Rexx Server Mode

Switch to
Rexx Server
mode.

Clock

Switch to title bar clock/mem-meter iconification mode.

Calendar

Switch to Clock/Calendar iconification mode.

AppIcon

Switch to AppIcon mode.

Iconify

Switch to the selected iconification mode.

1.24 mc_menus_report

2.5.2.3) Report

Upcoming Events

See
Upcoming
gadget description.

Summary

See
Summary
gadget description.

Today's Events

See
Today
gadget description.

1.25 mc_menus_move

2.5.2.4) Move

Next Month

See
==>
gadget description.

Previous Month

See
<==
gadget description.

Current Month

See
Now
gadget description.

1.26 rexx_serv

2.6) REXX Server

While DateBook is running (except in CLI mode), there is a rexx port available. The default rexx port name is 'datebook_rx'. See 'DBRx.rxx' for examples of all commands.

- 2.6.1
 - Database Commands
 - 2.6.2
 - Record Commands
 - 2.6.3
 - Search Commands
 - 2.6.4
 - Reports
 - 2.6.5
 - Date functions
 - 2.6.6
 - General Control

1.27 rx_dbase

2.6.1) Database Commands

- 2.6.1.1
 - OPENDB
 - 2.6.1.2
 - SAVEDB
 - 2.6.1.3
 - SORTDB

1.28 rx_opendb

2.6.1.1) OPENDB <name> [CREATE]

Opens a Datebook database. Any open file is closed first.

<name> - name of the database to open. If no path is specified the normal search path will be used.

CREATE - If the database does not exist, then create it. Default is to fail if the database does not exist.

RC: 0 = SUCCESS
 1 = SUCCESS - File Created (only when CREATE specified)
 9 = FAIL - Does not exist

Result: 0 = "SUCCESS"
 1 = "SUCCESS - File Created"

Example: 'opendb s:dats.dat create'

1.29 rx_savedb

2.6.1.2) SAVEDB [<name>]

Write the database to disk.

<name> - New name and optional path for database.

RC: 0 = SUCCESS
 9 = Error Writing file

Result: "SUCCESS"

Example: 'savedb' /* save to current filename */
 or 'savedb temp:dats.dat'

1.30 rx_sortdb

2.6.1.3) SORTDB

Sort Database by date and time.

RC: 0 = SUCCESS
 9 = FAIL, no current database, database empty.

Result: "SORTED"

Example: 'sortdb'

1.31 rx_record

2.6.2) Record Commands

2.6.2.1

ADDRECORD
2.6.2.2
DELETERECORD
2.6.2.3
NEXTRECORD
2.6.2.4
PREVRECORD
2.6.2.5
GOTORECORD
2.6.2.6
GETFIELD
2.6.2.7
SETFIELD

1.32 rx_addrrecord

2.6.2.1) ADDRECORD [DUPE]

Add a new blank record to the end of the database.

DUPE - Duplicate the current record instead of blanking the new record.

RC: 0 = SUCCESS
9 = FAIL, out of memory, no current record with the DUPE option

Result: "<record number>"

Example: 'addrecord'

1.33 rx_deleterecord

2.6.2.2) DELETERECORD

Blank current record. It will be removed by sorting or Upcoming reports.

RC: 0 = SUCCESS
9 = FAIL, record number out of range?

Result: "DELETED"

Example: 'deleterecord'

1.34 rx_nextrecord

2.6.2.3) NEXTRECORD

Move to the next record in the database.

RC: 0 = SUCCESS
9 = FAIL, End of file.

Result: <record number>

Example: 'nextrecord'

1.35 rx_prevrecord

2.6.2.4) PREVRECORD

Move to the previous record in the database.

RC: 0 = SUCCESS
 9 = FAIL, At start of file.

Result: <record number>

Example: 'prevrecord'

1.36 rx_gotorecord

2.6.2.5) GOTORECORD <rec_number>|FIRST|LAST

Move to a specific record in the database.

<rec_number> - Make rec_number the current record.
FIRST - Make the first record the current record.
LAST - Make the last record the current record.

RC: 0 = SUCCESS
 9 = FAIL, Record number out of range.

Result: <record number>

Example: 'gotorecord 23'
 or 'gotorecord last'
 or 'gotorecord first'

1.37 rx_getfield

2.6.2.6) GETFIELD \$<field_token> [\$<field_token> ...]

Retrieve the contents of one or more fields from the current record.

<field_token> - Supported tokens are...

\$D - Contents of the Date field.
\$T - Contents of the Time field.
\$E - Contents of the Event field.
\$Y - Contents of the Cycle field.
\$W - Contents of the Warning field.
\$F - Contents of the Frequency field.
\$X - Contents of the Rexx field.
\$N - Contents of the Notes field.
\$R - Current record number.
\$C - Current Date.
\$(<name>) - Contents of an environmental variable
 '<name>'.
 '

RC: 0 = SUCCESS

5 = Bad field Token
 9 = FAIL, no current record.

Result: ' "contents" "contents"....'

Example: 'getfield \$d \$t %e' /* Get the contents of the date,
 * time and event fields)
 */

1.38 rx_setfield

2.6.2.7) SETFIELD \$<field_token> <contents> [\$<field_token> <contents> ...]

Update the contents of one or more fields in the current record.

<field_token> - Supported tokens are...

\$D	- Contents of the Date field.
\$T	- Contents of the Time field.
\$P	- Contents of the Priority field.
\$E	- Contents of the Event field.
\$Y	- Contents of the Cycle field.
\$W	- Contents of the Warning field.
\$F	- Contents of the Frequency field.
\$X	- Contents of the Rexx field.
\$N	- Contents of the Notes field.

<contents> - Every thing following the space up to next field token or the end of the line will be inserted into the specified field.

RC: 0 = SUCCESS
 5 = Bad field Token
 9 = FAIL, invalid token, no current record, record could not be saved...

Result: 'Done'

Example: 'setfield \$d 960601 \$t 12:30 \$e Eat Lunch'

1.39 rx_search

2.6.3) Search Commands

2.6.3.1

FINDRECORD
 2.6.3.2
 NEXTMATCH
 2.6.3.3
 PREVMATCH

1.40 rx_findrecord

2.6.3.1) FINDRECORD \$<field_token> <contents>

Setup a search and find the first record that matches <contents>.

<field_token> - Supported tokens are...

\$D	- Contents of the Date field.
\$T	- Contents of the Time field.
\$P	- Contents of the Priority field.
\$E	- Contents of the Event field.
\$Y	- Contents of the Cycle field.
\$W	- Contents of the Warning field.
\$F	- Contents of the Frequency field.
\$X	- Contents of the Rexx field.
\$N	- Contents of the Notes field.

<contents> - Every thing following the space up to the end of the line will be used to search the database.

RC: 0 = SUCCESS
 9 = FAIL, Not found, bad field token....

Result: <record number>

Example: 'findrecord %e PaNorAmA'

1.41 rx_nextmatch

2.6.3.2) NEXTMATCH

Find the next record that matches a previous call to FINDRECORD.

RC: 0 = SUCCESS
 5 = No search set up.
 9 = FAIL, Not found,

Result: <record number>

Example: 'nextmatch'

1.42 rx_prevmatch

2.6.3.3) PREVMATCH

Find the previous record that matches a previous call to FINDRECORD.

RC: 0 = SUCCESS
 5 = No search set up.
 9 = FAIL, Not found.

Result: <record number>

Example: 'prevmatch'

1.43 rx_reports

2.6.4) Reports

2.6.4.1

SUMMARY
 2.6.4.2
EVENTS
 2.6.4.3
TODAY
 2.6.4.4
UPCOMING

1.44 rx_summary

2.6.4.1) SUMMARY [<month> [<year>]]

Generate a

Month
report.

<month> - Month number for summary (1 to 12)
<year> - Year number for summary (>= current year)

RC: 0 = SUCCESS
 5 = No events.
 9 = Invalid Date...

Result: '<summary line>'CR'<summary line>'CR...

Example: 'summary 11 95'

1.45 rx_events

2.6.4.2) EVENTS <date>

Generate a

Day
report for the specified date.

<date> - Date to report on (YYYYMMDD).

RC: 0 = SUCCESS
 5 = No events.
 9 = FAIL, no current record, Invalid Date...

Result: '<summary line>'CR'<summary line>'...

Example: 'events 950611'

1.46 rx_today

2.6.4.3) TODAY

Generate a

Day
report for today.

RC: 0 = SUCCESS
 5 = No events.
 9 = FAIL, no current record, Invalid Date...

Result: '<summary line>'CR'<summary line>'...

Example: 'today'

1.47 rx_upcoming

2.6.4.4) UPCOMING [UPDATE]

Generate an

Upcoming
report.

UPDATE - Cycle expired events while creating report. The sorted file will be saved to the current filename.

RC: 0 = SUCCESS
 5 = No events.
 9 = FAIL, no current record, Invalid Date...

Result: '<summary line>'CR'<summary line>'...

```
Example: 'upcoming'           /* Just generate the report */
      or  'upcoming UPDATE'    /* Generate the report and update
                               * expired events
                               */
```

1.48 rx_date

2.6.5) Date functions

2.6.5.1

```
STRING2DATE
  2.6.5.2
GETMONTHNAME
  2.6.5.3
DAYNAME
  2.6.5.4
EXPANDDATE
  2.6.5.5
ISVALID
```

1.49 rx_string2date

2.6.5.1) STRING2DATE <string>

Convert a string into a packed date (YYYYMMDD).

<string> - See the
DATE
argument in Command line description.

RC: 0 = SUCCESS
 9 = FAIL, invalid string, can't make sense of it.

Result: <date>

```
Example: 'string2date yesterday'
      or  'string2date 10-nov'
```

1.50 rx_getmonthname

2.6.5.2) GETMONTHNAME <month_num>

Find the name of a month from it's ordinal number (1 - 12).

<month_num> - Month number.

RC: 0 = SUCCESS
 9 = FAIL, invalid month number.

Result: <month name>

Example: 'getmonthname 2'

1.51 rx_dayname

2.6.5.3) DAYNAME <date>

Get the name of the day of the week that the specified date falls on.

<date> - Valid date YYYYMMDD.

RC: 0 = SUCCESS
 9 = FAIL, invalid date.

Result: <dayname> (Monday...)

Example: 'dayname 960505'
 -> "Sunday"

1.52 rx_expanddate

2.6.5.4) EXPANDDATE <date>

Expand packed date to a more human readable form.

<date> - Valid date YYYYMMDD.

RC: 0 = SUCCESS
 9 = FAIL, invalid date.

Result: Tue Jan 1 1995

Example: 'expanddate 950101'

1.53 rx_isvalid

2.6.5.5) ISVALID <date>

Determine if the given packed date is valid.

<date> - Date YYYYMMDD.

```
RC:      0 = SUCCESS
         9 = FAIL, invalid date.
```

```
Result: "YES"
```

```
Example: 'isvalid 961302' /* this will be invalid */
```

1.54 rx_gencontrol

2.6.6) General Control

2.6.6.1

```
CHANGEMODE
```

```
  2.6.6.2
```

```
SETICONIFY
```

```
  2.6.6.3
```

```
SHUTDOWN
```

1.55 rx_cngmode

```
2.6.6.1) CHANGEMODE EDIT|MEMO|SERVER|ICONIFY|CALENDAR|CLOCK| ↔
APPICON
[PUBSCREEN <screen_name>]
```

Change DateBook's state.

```
EDIT
```

```
- Record editor.
```

```
MEMO
```

```
- MemoCal Mode.
```

```
SERVER
```

```
- REXX Server Mode.
```

```
ICONIFY
```

```
- Current Iconified Mode
```

```
CALENDAR
```

```
- Clock with Calendar Mode
```

```
CLOCK
```

```
- Title Bar Clock Mode
```

APPICON
- AppIcon Mode

PUBSCREEN - Open gui on the named Public screen.

RC: 0 = SUCCESS
 5 = Can't find screen, used Workbench.
 9 = FAIL,

Result: <old mode>

Example: 'CHANGEMODE mode edit'
 or 'CHANGEMODE mode memo pubscreen TrapDoor'

1.56 rx_iconfiy

2.6.6.2) SETICONIFY CLOCK|CAL|ICON

Change the Iconification type.

CLOCK
- Iconify to Clock/Memory meter.

CAL
- Iconify to Clock/Memory meter with calendar.

ICON
- Iconify to an AppIcon.

RC: 0

Result: <old mode>

Example: 'SETICONIFY CLOCK'

1.57 rx_shutdown

2.6.6.3) SHUTDOWN

Close Datebook down.

RC: 0

Result: 'BYE'

Example: 'shutdown'

1.58 iconify

2.7) Iconify Modes

DateBookIV has three methods of Iconifying. AppIcon, Clock and Calendar modes.

2.7.1

AppIcon
2.7.2
Clock
2.7.3
Calendar

1.59 appicon

2.7.1) AppIcon

If a file is dropped on the icon, the normal GUI will reopen after attempting to load the dropped file. Be careful to only drop DateBook database files on the icon.

Double clicking on the icon will reopen the GUI.

1.60 clock

2.7.2) Clock

DateBook's window is shrunk to title bar height. The time and available memory are displayed in the title bar.

Menus:

2.7.2.1

Project
2.7.2.2
Mode

1.61 cl_menus_project

2.7.2.1) Project

Set/Save Config...

Change, save and load configuration.

See

Configuration Requester

HELP!...

AmigaGuide help (This File).

The main DateBookIV window is still active while the help window is open.

About...

Bragging:-). Also shows version numbers and contact info.

Quit

Exit the program. Same as Close Gadget.

1.62 cl_menus_mode

2.5.2.2) Mode

Editor

Switch to

Record Editor
mode.

MemoCal

Switch to

MemoCal
mode.

Rexx Server Mode

Switch to

Rexx Server
mode.

Clock

Switch to title bar clock/mem-meter iconification mode.

Calendar

Switch to Clock/Calendar iconification mode.

AppIcon

Switch to AppIcon mode.

UnIconify

Switch back to the previous mode.

1.63 calendar

2.7.3) Calendar

DateBook's window is changed to display a small calendar for the current month with the current day highlighted. The date and time are displayed above the calendar. Available memory is displayed in the title bar.

Menus

See

Clock Mode

1.64 db_help

2.8) DateBook Help

This AmigaGuide document is available as 'online' help. It can be activated from the Help gadget, Help menu or (surprise!) the Help key. While the AmigaGuide window is open, DateBook is still alive and awaiting your command.

When DateBook starts, it first looks in the current directory for the Help file (DateBook.guide). If it's not there, it then looks for the environmental variable 'DateBook/Help', and finally checks PROGDIR:. The environmental variable contains the path to the help file. If you use the Installer script, this will be set for you.

```
EG: 1> setenv DateBook/Help Docs:
    1> setenv envarc:DateBook/Help Docs:
```

1.65 configuration

2.9) Configuration Options

All options can be overridden by command line or tooltypes.

2.9.1) Iconification Mode Gadgets

See

Iconify Modes
2.9.2) Misc Options Gadgets

Keypad Movement

Allows you to choose to use the keypad as a cursor pad.
See

Record Editor

.

Confirm Req's

By default, before performing important operations, DateBook opens a confirmation requester. Deselecting this gadget will skip these requesters.

Notification

This allows DateBook to know if the data file has been changed by any other program. If 'Confirm Req's' is turned on, a requester opens to give you the option of re-loading the database file. If 'Confirm Req's' is off the file is automatically re-loaded.

2.9.3) Save Window Positions Gadgets

By default, all windows open in the middle of the visible area of the screen. Setting any of the window type gadgets will save it's last screen position in the configuration file.

Since Intuition does not seem to provide any means of getting the AppIcon's location, two string gadget's are supplied to enter your preferred location by hand.

2.9.4) Startup Mode Gadgets

You may choose to start DateBook in any of the following modes:

Edit

Iconified

MemoCal

Rexx Server

CLI

2.9.5) Control Gadgets

Use

Accept configuration and return to previous mode.

Load...

Load a configuration file.

Revert

Undo any changes.

Save As...

Save the configuration to the user selected file.

Save

Save the configuration to the current filename.

1.66 convert

2.10) Converting DateBook Data

If you have an existing DateBook database from any previous release, make sure you use this program to update the database for use with this version of DateBook. The original database is renamed with a ``.bak`` extension.

2.10.1) CLI Usage

```
ConvertDateBook [<filename>]
```

Where :

```
<filename>    - The path and name to convert.  
               Default `S:Dates.dat`.
```

2.10.2) WorkBench Tools.

```
FILE    - Set this to the complete file path and name  
          of your data file.
```

```
EG: FILE=Sys:BlackBook/HotDates
```

1.67 record_desc

3) Record Description

The DateBook data is stored in a `'Flat File'` database. The file consists of a 4 byte header (magic cookie), for version identification, and one or more records. Each record is 274 bytes and consists of the following fields:

3.1

Date Field

3.2

Time Field

3.3
Priority Field
3.4
Event Field
3.5
Cycle Field
3.6
Warning Field
3.7
Freq Field
3.8
Rexx Field
3.9
Notes Field

1.68 date_field

3.1) Date Field

Type: Character.
Size: 8 bytes.

Stored in the format YYYYMMDD. For readability the DateBook Editor will display the date as YYYY/MM/DD.

The editor will only accept numeric input for this field.

1.69 time_field

3.2) Time Field

Type: Character.
Size: 4 bytes.

Stored in the format HHMM. For readability the DateBook Editor will display the date as HH:MM. In 24 hour format eg: 23:00.

The editor will only accept numeric input for this field.

1.70 pri_field

3.3) Priority Field

Type: Number.
Size: 2 bytes.

Not currently used by DateBook, it is intended for later expansion (Day Planner / Task Scheduler). It can be used by rexx macros in the mean time :).

The editor will only accept numeric input for this field.

1.71 event_field

3.4) Event Field

Type: Character.

Size: 40 bytes.

A string to describe the event. This string is printed by DateBook when the event is displayed in reports.

The event string can contain special tokens to substitute information calculated as the report is generated. Like the internal variable substitution, see {" ARexx Field " link REXX_FIELD}, a dollar symbol followed by a Key letter.

Supported Keys are:

`$Y<yyyy>` - Difference in years between specified year and the Event's year.

EG: Mom and Dad's \$Y1950 Anniversary.

Result: Mom and Dad's 45th Anniversary.

1.72 cycle_field

3.5) Cycle Field

Type: Character.

Size: 16 bytes.

Instructions for DateBookIV to handle expired events.

Valid entries in the Cycle field are:

K[keep] - Never automatically delete this record.

Y[early] [<number>] - Increment the year by <number>. Default is one year.

M[onth] [<number>] - Increment the month by <number>. Default is one month.

W[EEKly] [<number>] - Increment by <number> weeks. Default is one week

L[ast] [<dayname>] [<month>] - Set to last day of following month. If <dayname> is used then the

event will be cycled to the last occurrence of that day in the next or specified month.

<week #> <dayname> [<month>] - For example, '2 Sun May' would specify Mothers day. If no month is specified the month will be incremented by one and the appropriate day is inserted. 'week #' must be between 1 and 4.

E[aster] [F[riday]] - 'Easter' alone will cycle to the following Easter Sunday. If 'Friday' is specified then it will cycle to the following Easter Friday.

1.73 warning_field

3.6) Warning Field

Type: Number.
Size: 2 bytes.

Number of days ahead to start warning you of the event. DateBookIV will automatically warn you the day of the event, the day before and the day after.

The editor will only accept numeric input for this field.

1.74 freq_field

3.7) Freq Field

Type: Number.
Size: 2 bytes.

Days between warnings. For example, if the Warning field is '10' and the Freq field is '2' then DateBookIV will display the event 10 days ahead and every second day until the event has passed.

The editor will only accept numeric input for this field.

1.75 rexx_field

3.8) Rext Field

Type: Character.
Size: 80 bytes.

A Rexx command name or rexx command string. On the day of the event, the Rexx field will be executed. If the first character is a single quote (') the field is treated as a rexx command string. Otherwise, it is treated as a rexx command file name and arguments.

A rexx string is simply one or more valid rexx statements separated by semi-colons ';'.

For Example:

```
'address command 'Say "Don't forget your Doctors Appointment"'
```

will call the AmigaDOS Say command (assuming you still have it kicking around), passing it the string "Don't forget your Doctors Appointment".

The Rexx field can use internal and environmental variables. These variables are expanded prior to calling ARExx.

Variable Syntax:

```
$D      - Contents of the Date field.
$T      - Contents of the Time field.
$P      - Contents of the Priority field.
$E      - Contents of the Event field.
$Y      - Contents of the Cycle field.
$W      - Contents of the Warning field.
$F      - Contents of the Frequency field.
$X      - Contents of the Rexx field.
$N      - Contents of the Notes field.
$C      - Current Date.
$(<name>) - Contents of an environmental variable
          'name'.
```

For example with:

```
'say '$D $T $E $C $(font) and some text'
```

DateBookIV will produce (on my system, the day I wrote this):

```
950412 19:30 PaNorAmA WorkShop Meeting 950318 Siesta/11 and some text
```

One of the more useful things to do with the rexx field is to add an event to a

Cron program. For the following example, we assume that the Rexx field contains "Add_Alarm \$T 60 Something".

```
/* File : Add_Alarm.dbk
*
* Add an event to CyberCrons task list.
* Rexx field should be: 'Add_Alarm %T <lead_time> <message>'
*
*      Where:
*          %T      - DateBook will insert the event's
*                  time field.
*          lead_time - number of minutes prior to event
```

```

*                               time.
*           message - The rest of the line contains the
*                   message to display.
*
*/

parse arg event_time lead_time message

options results

Ding = '07'X

/* Set up the window to display our message */
window_string = '>con:10/10/400/50/Reminder/close/wait'

event_time = strip(event_time)

/* extract time */
hours = left(event_time, 2)
mins = right(event_time, 2)

mins = mins - lead_time /* Subtract lead time */

do while mins < 0 /* Adjust as needed */
  mins = mins + 60
  hours = hours - 1
end

message = Ding || Ding || message || Ding || Ding
command_string = echo "'message'" 'window_string

address 'CYBERCRON' /* talk to CyberCron's rexx port */

ADD_EVENT mins' 'hours' * * * 'command_string

exit 0

/* End Add_Alarm.dbk */

```

1.76 notes_field

3.9) Notes Field

Type: Character.
Size: 120 bytes.

This field is solely for the user's purposes. It is displayed if you double click on the event line in the report window.

1.77 installation

4) Installation

4.1) Automatic Installation

Simply double click on the InstallDateBook icon. This uses Commodore's Installer program (not included).

4.2) Manual Installation

To install DateBookIV manually, simply copy all files (or drag the icons) to a convenient place in your system.

If you have an existing database, run
ConvertDateBook
before
using DateBook.

Once DateBookIV is installed you can add the following to the end of your User-Startup.

```
Datebook MODE=CLI  
OR  
run <>NIL: DateBook MODE CLI WAIT QUIET <>CON:20/20/640/120/DateBook/AUTO  
OR  
run <>NIL: DateBook ; Let config file decide startup mode.
```

1.78 future

5) Future

This will be the last free version of DateBook (aside from bug fixes). DateBookV will be a commercial release, and of course, be greatly enhanced (if not re-written). Some of the planned features and enhancements are:

- Day Planner
 - Task Scheduler.
 - Completely font sensitive GUI.
 - Network aware, ie record locking...
 - More event cycling methods.
 - User defined event cycling methods.
 - Customized reports and queries.
 - Merge and extract database options.
 - Context sensitive Help.
-

Publisher's enquiries welcome :).

1.79 credits

6) Credits

Programming: Robert Hardy
Design: Robert Hardy
Consulting: Daryl Hegyi, Steve McClain,
Vince Hodges, Domenico Di Tomaso and
E. John Love

Thanks to my Beta Testers (and friends):

Gary Coleman, Tom Coleman, Ray Norman, Domenico Di Tomaso
Robert Robson and E. John Love.

1.80 history

7) Changes since V35.???

Changed Major revision to reflect minimum OS requirement (35 -> 37).

Eliminated the limitations imposed by storing the date in 6 digits.
I guess I'll hit this problem again in the year 10,000 :-).

Added a 'Notes' field to the database.

Added page icon to Report items if the event contains a note.

Added a 'Priority' field and shortened Time field.

Since I changed the record size I also changed the Magic Cookie, and updated
ConvertDateBook to handle all old formats.

Added full rexx port and server mode. See
Rexx Server

.

Added function key bindings. See
Function Keys

.

Added an 'ADD' button to the requesters for single day reports.

Added more options to the 'Last' cycle type. See
CYCLE FIELD
Added handling for calculating Easter.

More and hopefully better documentation.

Fixed crash bug when Rexx field was executed.

Turfed Zoom gadget and added an Iconify gadget.

Created three iconification modes.

Added configuration requester, with load and save options.

Added some measure of font sensitivity.

Removed initialization after printing since this causes some printers to eject an extra page.

Added configuration option to turn off confirmation requesters.

Added Keypad movement option for the record editor.

Added optional file notification on the database.

1.81 about_cron

What is a Cron?

A cron is a program that runs in the background (i.e., requires no user intervention) and executes commands at specific times, as defined in a "crontab file". It allows the automation of jobs that one runs regularly.

Numerous Cron programs exist for the Amiga including TPTCRON, CyberCron, AmiCron and DCRON.

If you leave your machine running all the time like I do, you may want to add a 'Cron' task to automatically run DateBookIV on a daily basis. For CyberCron I use the following CronTab entry:

```
0 8 * * * ex:DateBook "MODE*=CLI" QUIET <>CON:10/10/640/120/DateBookIV/AUTO/CLOSE/ ↵  
WAIT
```

1.82 about_arexx

ARexx is the standard interprocess language for the Commodore-Amiga line of microcomputers. ARexx was written by William S. Hawes in 1987, and is the Amiga implementation of the REXX language originally developed by M. F Cowlshaw at IBM.

Many Amiga applications are ARexx compatible, which means that ARexx can be used to automate or otherwise aid the user.

The ARexx language is included as part of Workbench Release 2.0 and higher, and is available as a separate product for Workbench 1.3 systems.

For more information, please consult the Release 2.0 manual, or the ARexx User's Reference Manual by William Hawes.

1.83 dun_dat

'Dates.dat' is an example of a '.dat' file used by DateBook to store the data entered by the user.

The '.dat' extension must be used in the filename in order for DateBook to recognize the file. Other than that, you may name your .dat file to anything you want. 'Bobs.dat', 'PaNorAmA.dat', or '123.dat' are all valid names.