

**Remider**

**COLLABORATORS**

	<i>TITLE :</i> Reminder		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 5, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Reminder</b>	<b>1</b>
1.1	Reminder V1.20 . . . . .	1
1.2	Copyright And Distribution . . . . .	2
1.3	How to Avoid Reading This Manual . . . . .	2
1.4	What is Reminder and Why? . . . . .	3
1.5	System Requirements . . . . .	3
1.6	How to Contact the Author . . . . .	3
1.7	Installing Reminder and ReminderCheck . . . . .	4
1.8	Reminder . . . . .	5
1.9	Reminder: Workbench Usage . . . . .	5
1.10	Reminder: Shell usage . . . . .	5
1.11	Entering and Editing Events . . . . .	6
1.12	The Date Gadgets . . . . .	6
1.13	Day-gadget . . . . .	8
1.14	Month-gadget . . . . .	8
1.15	Year-gadget . . . . .	9
1.16	Weekday-gadget . . . . .	9
1.17	The Before and After Gadgets . . . . .	9
1.18	Automatic Deletion of Acknowledged Events . . . . .	9
1.19	Grouped Events . . . . .	10
1.20	Event Text . . . . .	10
1.21	Event List . . . . .	10
1.22	Adding New Events . . . . .	11
1.23	Removing Events . . . . .	11
1.24	Updating Existing Event Information . . . . .	11
1.25	Quitting Reminder . . . . .	11
1.26	ARexx Gadgets . . . . .	12
1.27	Reminder: Keyboard shortcuts . . . . .	13
1.28	remindercheck . . . . .	13
1.29	ReminderCheck: Workbench usage . . . . .	14

---

---

1.30	ReminderCheck: Shell usage . . . . .	15
1.31	ReminderCheck Requesters . . . . .	15
1.32	Group Requesters . . . . .	15
1.33	ARexx with ReminderCheck . . . . .	16
1.34	ReminderCheck: Keyboard shortcuts . . . . .	16
1.35	Source Code . . . . .	16
1.36	dice . . . . .	17
1.37	Changing the user interface with GadToolsBox (under DICE) . . . . .	17
1.38	Compiling under SAS-C . . . . .	18
1.39	Changing the user interface with GadToolsBox (under SAS-C) . . . . .	18
1.40	Final Words and Acknowledgements . . . . .	18

---

# Chapter 1

## Reminder

### 1.1 Reminder V1.20

REMINDER 1.20 DOCUMENTATION

-----  
by Matti Rintala  
April 19, 1993

Abstract

This document describes the usage of Reminder and ReminderCheck programs. Both programs and the source code are copyright 1993 Matti Rintala. They can be distributed freely however, see section Distribution below.

Table of Contents:

-----  
Copyright and Distribution  
How to avoid reading this manual  
What is Reminder and why?  
System requirements  
How to Contact the author  
Installing Reminder and ReminderCheck  
Reminder  
ReminderCheck  
Source Code  
Final words and Acknowledgements

---

## 1.2 Copyright And Distribution

### COPYRIGHT AND DISTRIBUTION

Reminder and ReminderCheck are (C) 1993 Matti Rintala (This section is modified from the similar section of ToolManager 2.0 by Stefan Becker. I hope you don't mind, Stefan!)

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

No guarantee of any kind is given that the programs described in this document are 100 % reliable. You are using this material at your own risk. The author CAN NOT be made responsible for any damage which is caused by using these programs.

This package is freely distributable, but still copyright by Matti Rintala. This means that you can copy it freely as long as you don't ask for a more than nominal copying fee.

Permission is granted to include this package in Public-Domain collections, especially in Fred Fish's Amiga Disk Library (including CD ROM versions of it). The distribution file may be uploaded to Bulletin Board Systems or FTP servers. If you want to distribute this program you MUST use the contents of the original distribution archive 'Reminder-1.20.lha'. The archive may be distributed unpacked or packed with a different archiver, if necessary.

None of the programs nor the source code (nor parts of it) may be included or used in commercial programs unless by written permission from the author.

NONE of the programs NOR the source code (nor parts of it) may be used on any machine which is used for the research, development, construction, testing or production of weapons or other military applications. This also includes any machine which is used for training persons for ANY of the above mentioned purposes. The same goes for unnecessary animal testing.

You may modify the source code as you wish, provided that my name still exists there and you add a comment to the code explaining that the source is modified. If you think the change you made is useful, send it to me so that it can be included in the next official version (I promise to mention your name in the docs). PLEASE DO NOT DISTRIBUTE MODIFIED VERSIONS OF REMINDER AND REMINDERCHECK YOURSELF, AS THIS ONLY CREATES CONFLICTING VERSION NUMBERS ETC!

## 1.3 How to Avoid Reading This Manual

### HOW TO AVOID READING THIS MANUAL

Read QuickRef.doc. Try programs. Experiment. Guess.

---

## 1.4 What is Reminder and Why?

### WHAT IS REMINDER AND WHY?

For some time I have been searching for a program which would remind me of important events like birthdays, days when rent has to be paid etc. There are already several PD or shareware programs like CyberCron, which do mostly what I want - but they stay in the memory as background tasks and consume precious RAM.

Then I tried MemoMaster which could be made to run once in startup, but its 1.3-like user interface and certain other features (for example, the lack of monthly events) made it unsuitable for me.

After that I sighed and wrote Reminder. The system consists of two programs. Reminder is for entering your events to a database file, and ReminderCheck is put into your WBStartup, where it is run every time you boot and reminds you about the events and then quits.

Both programs use the excellent ReqTools.library ((C) Nico François), and the user interface of Reminder was created with GadToolsBox ((C) Jan van den Baard). The binaries in this distribution were compiled with SAS/C6.2 ((C) SAS Institute Inc.) with optimization on, but the source files can be compiled with DICE ((C) Matthew Dillon) also.

## 1.5 System Requirements

### SYSTEM REQUIREMENTS

Reminder and ReminderCheck require at least AmigaDOS 2.x and ReqTools.library (A copy of the library is included in this distribution). They were developed and have been tested on Amiga 2000 with GVP G-Force 68030/25MHz, 1 MB Chip and 5 MB Fast running AmigaDOS 2.04. They worked also with the G-Force turned off (i.e. normal Amiga 2000 with 1 MB Chip and no Fast). There is no reason why they shouldn't run on any Amiga with at least 2.x, however.

## 1.6 How to Contact the Author

### HOW TO CONTACT THE AUTHOR

First, if you are using Reminder, please send me e-mail, a postcard, Amiga 4000 or whatever. I'm interested in knowing that someone else is using Reminder, too. The feedback may also encourage me to make improvements etc, so all suggestions are wellcome.

The best way to contact me is via e-mail. My mail address is

bitti@cs.tut.fi

The postcards and the A4000 should go to address

---

Matti Rintala  
Pyykkiojankatu 2 C 24  
33710 TAMPERE  
FINLAND

## 1.7 Installing Reminder and ReminderCheck

### INSTALLING REMINDER AND REMINDERCHECK

Reminder and ReminderCheck can be installed from both Workbench or shell.

#### Installing from Workbench

Reminder V1.20 has an Installer (© Commodore-Amiga, Inc) script which takes care of the whole installation. If you don't have Installer, Commodore has made it freely available, so it should be found on many ftp sites. If you can't find it, then you'll have to install Reminder from shell.

Make sure Installer is in the AmigaDOS path ('`C:`' is where I have it), then double click on the ``Install-Reminder`` icon. The script installs Reminder and ReminderCheck, ReqTools.library (Only if the version coming with this distribution is newer than yours), documentation, and ARexx examples, asking you for confirmation if you don't select the Novice user level.

#### Installing from shell

Recommended way to install Reminder from shell is to use Installer, i.e. to cd in the directory where you unpacked this distribution and then say

```
Installer Install-Reminder
```

If you don't have Installer, however, read further.

First, cd into the directory where Reminder is. If you know that the ReqTools.library coming with this distribution is newer than yours (You can always check with Version command.) or you don't have ReqTools.library, copy it to correct place with

```
copy ReqTools.library LIBS:
```

It is recommended that you always get the whole ReqTools distribution, however. It can be found on Fish-disks and on most ftp sites.

Then copy the executables to appropriate places with

```
copy Reminder Reminder.info your_directory  
copy ReminderCheck ReminderCheck.info SYS:WBStartup
```

Finally copy the documentation and ARexx example files to appropriate places.

---



## 1.8 Reminder

### REMINDER

With Reminder you enter the events you want to be reminded about to the database which

ReminderCheck  
reads on every boot.

Workbench usage

Shell usage

Entering and Editing Events

The Date Gadgets

The Before and After Gadgets

Automatic deletion of Acknowledged Events

Grouped Events

Event Text Gadget

Event list

Add Gadget

Remove Gadget

Update Gadget

ARexx Gadgets

Quitting Reminder

Keyboard shortcuts

## 1.9 Reminder: Workbench Usage

Workbench usage

Reminder can be started from Workbench simply by double clicking its icon. In the icon you can define the database file (Default name is ``S:Reminder.data'') used by Reminder with tooltype

FILE=filename

## 1.10 Reminder: Shell usage

---

### Shell usage

Reminder can be started from shell simply by typing Reminder and pressing Enter. You can also type ``FILE=filename`` as a command line argument to specify the database file (Default name is ``S:Reminder.data``) to be used.

## 1.11 Entering and Editing Events

### Entering and Editing Events

After starting Reminder a window opens, presenting a list of existing events and gadgets needed to add and edit events.

The events are sorted in chronological order, repeating events in order they occur next (this is new in V1.10). Clicking on an event in event list copies information for that event to appropriate gadgets, in which you can edit them (don't forget to

Update  
the event

after editing).

If you have edited the information in the gadgets and try to perform an action which would make you lose the information (you click on event in event list, or click New,

Quit, no save  
or  
Quit and save  
) ,

Reminder asks you to confirm that you wish to abandon the current values of gadgets.

Use

Add  
to add the event in gadgets to event list.

Click on event, edit it and then click

Update  
to update event data.

Click on event and click

Remove  
to remove event.

Click on New to clear the gadgets to default values.

## 1.12 The Date Gadgets

### The Date Gadgets

The date of an event consists of the

Day-gadget

,

Month-gadget

,

Year-gadget

and

Weekday-gadget

. Leaving any of these gadgets empty

(

Month-gadget

and

Weekday-gadget

show ANY instead of empty gadget)

means that any value is acceptable. This way you can enter repeating events.

Weekday-gadget

is different from other gadgets, however.

SELECTING A WEEKDAY INTO

Weekday-gadget

MEANS THAT THE EVENT IS DUE ON

THE NEXT SPECIFIED WEEKDAY ON OR AFTER THE DATE SELECTED (on the same month, however).

The text gadget above

Day-gadget

is a read-only gadget telling

the weekday of non-repeating exact events (i.e. events with

Day-gadget

,

Month-gadget

and

Year-gadget

set to specific values and

Weekday-gadget

as

``ANY``). I myself have very little use for this gadget, but I was requested it. The event list shows the weekday of non-repeating exact events, also. There is a colon after the weekday indicating that the weekday is not entered by you, but calculated by Reminder.

A few examples should clarify the usage of date gadgets.

- Entering 22 to

Day-gadget

, March to

Month-gadget

, 1993 to

Year-gadget

and ANY to

Weekday-gadget

means that the event is on March

22, 1993.

- Entering 22 to  
Day-gadget  
, March to  
Month-gadget  
, empty to  
  
Year-gadget  
and ANY to  
Weekday-gadget  
means that the event is on March  
22, every year.
- Entering 1 to  
Day-gadget  
, empty to  
Month-gadget  
, 1993 to  
Year-gadget  
and Thursday to  
Weekday-gadget  
means that the event is on first  
Thursday (Actually first Thursday on or after the first day of the  
month, which is the same thing.) of every month on year 1993.
- Entering 8 to  
Day-gadget  
, empty to  
Month-gadget  
, empty to  
  
Year-gadget  
and Monday to  
Weekday-gadget  
means that the event is on  
second Monday (The first Monday is always before the 8. day of the  
month and the second can't be before the 8. day.) of every month,  
every year.

### 1.13 Day-gadget

Day-gadget accepts all numbers from 1 to 31, empty gadget representing 'any day'.

### 1.14 Month-gadget

You can select a month into Month-gadget from the list above it. You can of course enter the month by hand, too. In this case Reminder selects the first month starting with the letters you entered (I.e. entering 'ap' would select April). Entering the month this way is case-insensitive. You can also enter into the gadget just the number of the month (I.e. entering 10 would select October). Leaving the gadget empty (or entering or selecting ANY) means 'any month'.

---

## 1.15 Year-gadget

Year-gadget accepts all numbers from 1993 to 2099 (Upper limit is just to make you notice stupid typos like 2993.). You can also enter a number from 93 to 99 to represent years 1993 - 1999 or a number from 0 to 92 to represent years 2000 - 2092. Leaving the gadget empty means 'any year'.

## 1.16 Weekday-gadget

You can enter a value to Weekday-gadget by selecting a weekday ←  
from  
the list above the gadget. Selecting ANY means 'any weekday'.

NOTE THAT SELECTING A WEEKDAY MEANS THAT THE EVENT IS DUE ON THE NEXT CORRECT WEEKDAY ON OR AFTER THE DATE SPECIFIED WITH OTHER DATE GADGETS. (See examples in  
The Date Gadgets  
).

## 1.17 The Before and After Gadgets

The Before and After Gadgets

With these gadgets you can select how long before the event ReminderCheck starts reminding you about it, and how long after the event ReminderCheck should still remind about it, if you haven't acknowledged the event before that.

Before-gadget accepts any value from 1 to 365, defining how many days before the event ReminderCheck should start reminding. Leaving the gadget empty means that reminding should start on the day the event is due.

After-gadget accepts any value from 1 to 365, defining how many days after the event ReminderCheck should still remind about it if the event has not been acknowledged. Leaving the gadget empty means that reminding should stop after the day the event is due.

## 1.18 Automatic Deletion of Acknowledged Events

Automatic Deletion of Acknowledged Events

If you check the 'Delete when...'-gadget, ReminderCheck automatically deletes the events you acknowledge. If you leave this gadget unchecked, events are not deleted unless you remove them yourself with Reminder.

NOTE THAT USING THE AUTOMATIC DELETION WITH REPEATING EVENTS IS NOT

---

USUALLY WISE AS THE EVENT IS DELETED AFTER THE FIRST ACKNOWLEDGEMENT OF THE EVENT!

## 1.19 Grouped Events

### Grouped Events

If you mark the event grouped (This feature was introduced in Reminder V1.20) with 'Grouped'-gadget, ReminderCheck will put it into the same requester with all other grouped events that are due on that specific day.

Note that if you want to be reminded about your event two days before the actual day, those reminders will be done in normal fashion. The event will be put into the group requester ONLY on the day it is due, no earlier or later.

Grouping is a handy way to handle minor ``notifications``, which you always acknowledge immediately when they are first reminded about.

## 1.20 Event Text

### Event Text

Into Text-gadget you should write the text you want to appear in the reminding requester. The maximum length of the text is 80 characters.

## 1.21 Event List

### Event List

The listview gadget in the upper left corner shows all events currently in the event database. The list shows the weekday and date of the event (or stars for 'wildcards') and the beginning of the event text. If your event is an non-repeating exact event (

```
Day-gadget
,
Month-gadget
and
Year-gadget
have all values and
Weekday-gadget
is
```

``ANY``), the event list shows the calculated weekday of the event (This feature was introduced in Reminder V1.10) plus a colon to tell you that the weekday was calculated, not entered by you. I.e. if the event list shows ``Thu:01-Apr-1993 Test``, it means that you haven't requested a specific weekday, but first of April happens to be Thursday. ``Sun 01-Apr-1993 Test2`` tells, on the other hand, that

your event is due on the first Sunday on or after first of April.

The events in the event list are sorted in chronological order (This feature was introduced in Reminder V1.10). The chronological order of repeating events is simply the order in which they will occur next time.

## 1.22 Adding New Events

### Adding New Events

Adding new events is easy. Just enter the event parameters using the gadgets described above and click Add-gadget. The event appears in the

event list  
.

## 1.23 Removing Events

### Removing Events

Removing events is even easier. Select the event from the event list and click Remove-gadget. The event is removed from the

event list  
and  
from the database.

## 1.24 Updating Existing Event Information

### Updating Existing Event Information

With Update-gadget you can change parameters of existing events. Just select the event from the

event list  
. All event gadgets now show the  
values for that event. Update any gadgets you want and then click  
Update-gadget. The event is now updated and the  
event list  
also shows  
the updated event information.

## 1.25 Quitting Reminder

### Quitting Reminder

There are three ways to quit Reminder:

---

- Click 'Quit and save'-gadget. The changes to the event database are saved and Reminder terminates.
- Click 'Quit, no save'-gadget. If the database has been changed, you are first asked whether you really want to exit Reminder without saving the changes. If you answer Yes, the changes are lost, otherwise you are returned to Reminder.
- Click the close gadget of Reminder window. If the database has been changed, you are asked whether you want to save the changes. If you answer Yes, the changes are saved, otherwise they are lost. Reminder then terminates.

## 1.26 ARexx Gadgets

### ARexx Gadgets

#### ReminderCheck

V1.20 is capable of launching ARexx scripts and commands with every event.

With the radio buttons in the ARexx box (''Nothing'', ''Script'' and ''Command'') you can select the type of ARexx action associated with current event.

''Nothing'' naturally means no ARexx action at all.

''Script'' means that

#### ReminderCheck

is to start the given ARexx script

when the event is due (the requester pops up, as usual, too). The name of the script (and parameters) should be entered to the Comm/Script-gadget. The script gets the date of the event and the event text (both enclosed in double quotes) as its last two parameters (i.e. after the parameters specified in the gadget). This way, if you use Upd (© Jonas Pettersson) like I do, you can easily create a script which uses narrator to speak the event text. Such script is included as an example. Upd itself is not provided, however. It is available via ftp. Another example provided can be used to play an audio sample (with Upd, again).

''Command'' tells

#### ReminderCheck

to send the command given in

Comm/Script-gadget to ARexx port specified in Port-gadget. This time no extra arguments are added.

NOTE: At the moment maximum length of text in Comm/Script-gadget is 28 characters, and maximum length of ARexx port in Port-gadget name is 10 characters. These values are far too small, but it was the only way to keep the database file format compatible, so that Reminder V1.00 database files can be used directly with V1.20. This will change in V1.30 (the release time of which depends on the feedback I get).



## 1.27 Reminder: Keyboard shortcuts

### Keyboard shortcuts

Reminder can be used with keyboard only, too. Most of the gadgets have underlined characters, which activate the gadget (you don't have to press Alt or Amiga keys or anything).

With

```

    event list
    and
    Weekday-gadget
    , pressing the key without shift-key selects
next entry, pressing with shift-key selects previous entry.
```

There are two extra features:

- - Weekday-gadget
    - can also be selected with number keys 0 - 7. 0 means ANY, 1 is ←
    - Monday etc.
- Events may be selected from the
  - event list
  - using the cursor-up and
 cursor-down keys.

## 1.28 remindercheck

### REMINDERCHECK

ReminderCheck should be placed into your WBStartup drawer or called from your s:user-startup. It checks events entered with

```

Reminder
on every boot (although you can set the minimum interval) and ←
reminds
you about the events which are due.
```

Workbench usage

Shell usage

ReminderCheck requesters

Keyboard shortcuts

ARexx with ReminderCheck

## 1.29 ReminderCheck: Workbench usage

### Workbench usage

ReminderCheck can be started from Workbench simply by double clicking its icon. In the icon you can define the database file (Default name is ``S:Reminder.data``.) used by ReminderCheck with tooltype

FILE=filename

You can set the minimum time that has to pass before ReminderCheck reminds you again (I added this feature because I happen to boot my machine quite often.). Default is to remind every time ReminderCheck is run. The interval is set with tooltype

INTERVAL=minimum # of hours between two reminders

For example, if you set INTERVAL=3, run ReminderCheck and it reminds you, then running ReminderCheck again within next three hours does nothing.

If you do not use ReminderCheck's ability to run AREXX scripts and commands, you may not want ReminderCheck to open RexxSysLib.library at all. In that case include tooltype

NOAREXX

Some people run ReminderCheck from their user-startup so early that Workbench screen hasn't been opened yet. In that case ReqTools.library can't open its requesters. ReminderCheck uses Intuition's standard EasyRequest requester, if you include tooltype

NOREQTOOLS

You can specify the maximum number of events in a group requester with tooltype

MAXGROUP=maximum # of events in one group requester

The default value is 4. The value should be small enough so that the requester will fit the screen.

In a similar way you can specify the maximum width of line in requesters with tooltype

MAXWIDTH=maximum # of characters per line in requester

The default value is 30, and it can't be under 19 (because that's the length of the date string).

If you use ReminderCheck in your SYS:WBStartup, you should include tooltype

DONOTWAIT

---

## 1.30 ReminderCheck: Shell usage

### Shell usage

ReminderCheck can be started from shell simply by typing ReminderCheck and pressing Enter. You can also type ```FILE=filename``` as a command line argument to specify the database file (Default name is ```S:Reminder.data```.) to be used. Other possible command line arguments are ```INTERVAL=# of hours``` which defines the minimum numbers of hours between two remindings, ```NOREQTOOLS``` which causes ReminderCheck not to use ReqTools.library, ```MAXGROUP=# of events``` to specify maximum number of events in group requester, ```MAXWIDTH=# of chars``` to specify maximum width of line and ```NOAREXX``` which causes ReminderCheck not to open RexasLib.library and thus not use ARExx.

## 1.31 ReminderCheck Requesters

### ReminderCheck Requesters

If ReminderCheck finds a event it has to remind you about, it pops up a requester telling you the date of the event and the event text you have entered for the event. The requester has three buttons:

- ```Go away```-gadget tells ReminderCheck that you have noticed the requester but want it to keep reminding you about this event and date (Next time ReminderCheck is run, that is.). For example, if the event says that you should pay your rent, you should use this button until you really have paid it.
- ```Acknowledge```-gadget tells ReminderCheck that you've dealt with this event, so reminding you again is not necessary. Of course, if the event is a repeating one (has other ```wildcards``` than the weekday), ReminderCheck will remind you again the next time the event is due. If you checked the ```Delete when...```-gadget when you entered the event, the event is deleted from the database when you acknowledge it.
- ```Quit```-gadget tells ReminderCheck to abort immediately.

### Group Requesters

## 1.32 Group Requesters

### Group Requesters

All the events which are marked ```grouped``` and whose date is ```today``` when ReminderCheck is run, are put into a single requester (or to as few requesters as possible) (This feature was introduced in ReminderCheck V1.20.) .

Acknowledging this requester means that you acknowledge ALL the events in that requester. There is no way to acknowledge (or not to acknowledge) single events in group requester.

NOTE: The ARexx event associated with group requester is ANY of the ARexx events of the events in the requester. There is no way to specify which one. Only one ARexx event is generated for each group requester.

### 1.33 ARexx with ReminderCheck

#### ARexx with ReminderCheck

If an event has an ARexx script or command associated with it, ReminderCheck sends the command or starts the script before it opens the requester for the event. The requester is opened immediately, however (i.e. ReminderCheck does not wait for the ARexx script to terminate).

NOTE: In order to get ARexx working, REXXMAST must be running at the time ReminderCheck is run. Because both these programs are usually launched from WBStartup, you can't be sure which of these will be run first. The only way is simply to try. If ARexx does not seem to work, you have to start REXXMAST from your user-startup.

Note also that ARexx does not work if you have the NOAREXX tooltype in your ReminderCheck icon (see Workbench usage).

### 1.34 ReminderCheck: Keyboard shortcuts

#### Keyboard shortcuts

Like Reminder, ReminderCheck has also keyboard shortcuts for the buttons. Just press the key that is underlined in the button text. The button whose text is in boldface ('Go away'-gadget) is the default button which can be selected with Enter key also. You can use ESC key to select Quit-gadget.

### 1.35 Source Code

#### SOURCE CODE

The distributed source should compile under both SAS/C6.x and DICE. IF YOU ARE PLANNING TO MAKE CHANGES TO SOURCE AND WANT TO DISTRIBUTE THE ALTERED VERSION, READ THE ``COPYRIGHT AND DISTRIBUTION`` SECTION.

---

To compile Reminder and ReminderCheck you will also need the .h and .lib files for ReqTools.library (At least version 38 needed.). These files are not included, but the complete ReqTools distribution can be found on many ftp sites and for example on Fish Disk #794.

Compiling under DICE

Compiling under SAS-C

## 1.36 dice

Compiling under DICE

The provided DMakefile should do most of the work, so just type ```dmake``` to compile both programs or ```dmake Reminder``` or ```dmake ReminderCheck``` to compile only one.

The source files have some parts between ```#ifdef _DCC``` and ```#endif```. These parts are not needed when compiled with SAS-C, so if your change applies to DICE only, use the same method to keep the source consistent.

Similarly the parts between ```#ifdef __SASC``` and ```#endif``` are not used when compiled with DICE, so make sure you don't write anything DICE needs into these parts.

Notice also that the files RemSASc.c, RemSAS.c and RemSAS.h are not used under DICE. They are GadToolsBox generated source files (and an additional prototype file) for SAS-C only.

Changing the user interface with GadToolsBox

## 1.37 Changing the user interface with GadToolsBox (under DICE)

Changing the user interface with GadToolsBox (under DICE)

The file ```Reminder.gui``` contains the description of the user interface in GadToolsBox2.0 format. If you change this with GadToolsBox, use the following steps to produce the new source:

1. Enter the C Source Preferences window.
2. Make sure ```Include pragmas```, ```Static``` and ```Aztec C``` are off and that ```Generate IDCMP handler``` is on (I also use ```Use System font```).
3. Select Use.
4. Select C source generation and enter ```RemDICE.c``` as the output file. The source is now generated in correct format and into the

correct file.

5. Exit GadToolsBox and type ```dmake Reminder``` to compiled Reminder.

## 1.38 Compiling under SAS-C

### Compiling under SAS-C

The provided smakefile should do most of the work, so just type ```smake``` to compile both programs or ```smake Reminder``` or ```smake ReminderCheck``` to compile only one.

The source files have a some parts between ```#ifdef __SASC``` and ```#endif```. These parts are not needed when compiled with DICE, so if your change applies only to SAS-C, use the same method to keep the source consistent.

Similarly the parts between ```#ifdef _DCC``` and ```#endif``` are not used when compiled with SAS-C, so make sure you don't write anything SAS-C needs into these places.

Notice also that the files `RemDICE.c` and `RemDICE.h` are not used under SAS-C. They are GadToolsBox generated source files for DICE only.

### Changing the user interface with GadToolsBox

## 1.39 Changing the user interface with GadToolsBox (under SAS-C)

### Changing the user interface with GadToolsBox (under SAS-C)

The file ```Reminder.gui``` contains the description of the user interface in GadToolsBox format. If you change this with GadToolsBox, use the following steps to produce the new source:

1. Enter the Preferences window.
2. Make sure ```Static``` and ```Aztec C``` are off and that ```Generate IDCMP handler``` and ```Include pragmas``` are on (I also use ```Use System font```).
3. Select Use.
4. Select C source generation and enter ```RemSAS.c``` as the output file. The source is now generated in correct format and into the correct file.
5. Exit GadToolsBox and type ```smake Reminder``` to compiled Reminder.

## 1.40 Final Words and Acknowledgements

---

## FINAL WORDS AND ACKNOWLEDGEMENTS

That's it, really! I hope you have use for Reminder, at least I do.

I'd like to thank the following people, without whom writing Reminder would have been impossible or at least much more painful:

- The folks at Commodore for creating Amiga in the first place
  - Matthew Dillon for DICE
  - SAS Institute for SAS-C
  - Nico François for ReqTools.library
  - Jan van den Baard for GadToolsBox
  - Steve Koren for SK-shell
  - Free Software Foundation and David Gay for GnuEmacs
  - The people who wrote RCS and Un\*x dmake and those who ported them to Amiga
  - And all people writing software (especially PD and freeware) to Amiga
-