Reminder 1.20 Documentation

Matti Rintala

April 19, 1993

Abstract

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

Contents

1 Copyright and Distribution **2**

2 How to Avoid Reading This Manual **3**

3 What is Reminder and Why? **3**

4 System Requirements **4**

5 How to Contact the Author **4**

6 Installing Reminder and ReminderCheck **4**
6.1 Installing from Workbench 4
6.2 Installing from shell 5

7 Reminder **5**
7.1 Workbench usage 5
7.2 Shell usage 5
7.3 Entering and Editing Events 6
7.3.1 The Date Gadgets 6
7.3.2 The Before and After Gadgets 7
7.3.3 Automatic Deletion of Acknowledged Events 7
7.3.4 Grouped Events 7
7.3.5 Event Text 8
7.4 Event List 8
7.5 Adding New Events 8

7.6 Removing Events 8
7.7 Updating Existing Event Information 8
7.8 Starting a new event 9
7.9 Quitting Reminder 9
7.10 ARexx Gadgets 9
7.11 Keyboard shortcuts 10

8 ReminderCheck **10**
8.1 Workbench usage 10
8.2 Shell usage 11
8.3 ReminderCheck Requesters 12
8.3.1 Group Requesters 12
8.4 ARexx with ReminderCheck 12
8.5 Keyboard shortcuts 13

9 Source Code **13**
9.1 Compiling under DICE 13
9.1.1 Changing the user interface with `GadToolsBox` 13
9.2 Compiling under SAS-C 14
9.2.1 Changing the user interface with `GadToolsBox` 14

10 Final Words and Acknowledgements **14**

1 Copyright and Distribution

[1] This section is modified from the similar section of `ToolManager 2.0` by *Stefan Becker*. I hope you don't mind, Stefan!

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!**

2 How to Avoid Reading This Manual

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

3 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`[2], and the user interface of Reminder was created with `GadToolsBox`[3]. The binaries in this distribution were compiled with `SAS/C6.2`[4] with optimization on, but the source files can be compiled with `DICE`[5] also.

[2] ©*Nico François*
[3] ©*Jan van den Baard*
[4] ©SAS Institute Inc.
[5] ©*Matthew Dillon*

4 System Requirements

REMINDER and REMINDERCHECK require at least **AmigaDOS 2.x** and `ReqTools.library`[6]. 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.

5 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

6 Installing REMINDER and REMINDERCHECK

REMINDER and REMINDERCHECK can be installed from both Workbench or shell.

6.1 Installing from Workbench

REMINDER V1.20 has an `Installer` [7] 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`[8], documentation and ARexx examples, asking you for confirmation if you don't select the Novice user level.

[6] A copy of the library is inluded in this distribution
[7] ©Commodore-Amiga, Inc
[8] Only if the version coming with this distribution is newer than yours

6.2 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[9] 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.

7 REMINDER

Reminder can be started from shell and from Workbench.

7.1 Workbench usage

REMINDER can be started from Workbench simply by double clicking its icon. In the icon you can define the database file[10] used by REMINDER with tooltype

```
FILE=filename
```

7.2 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[11] to be used.

[9] You can always check with `Version` command.

[10] Default name is "`S:Reminder.data`".

[11] Default name is "`S:Reminder.data`". *Repetitio est mater studiorum.*

7.3 Entering and Editing Events

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

7.3.1 The Date Gadgets

The date of an event consists of the Day , Month , Year and Weekday gadgets. Leaving any of these gadgets empty (Month and Weekday show `ANY` instead of empty gadget) means that any value is acceptable. This way you can enter repeating events. Weekday is different from other gadgets, however. **Selecting a weekday into Weekday means that the event is due on the next specified weekday ON OR AFTER the date selected.**

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

You can select a month into Month 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'.

Year accepts all numbers from 1993 to 2099[12]. 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'.

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

The text gadget above Day is a read-only gadget telling the weekday of non-repeating exact events[13] (i.e. events with Day , Month and Year set to specific values and Weekday 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 , `March` to Month , `1993` to Year and `ANY` to Weekday means that the event is on March 22, 1993.

[12] Upper limit is just to make you notice stupid typos like 2993.
[13] This feature was introduced in REMINDER V1.10.

- Entering `22` to Day , `March` to Month , empty to Year and `ANY` to Weekday means that the event is on March 22, every year.
- Entering `1` to Day , empty to Month , `1993` to Year and `Thursday` to Weekday means that the event is on first Thursday[14] of every month on year 1993.
- Entering `8` to Day , empty to Month , empty to Year and `Monday` to Weekday means that the event is on second Monday[15] of every month, every year.

7.3.2 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 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 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.

7.3.3 Automatic Deletion of Acknowledged Events

If you check the gadget ☑ `Delete event...` , 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!

7.3.4 Grouped Events

If you mark the event grouped[16] with gadget ☑ `Grouped` , ReminderCheck will put it into the same requester with **all other grouped events that are due on that specific day**.

[14] Actually first Thursday on or after the first day of the month, which is the same thing.

[15] The first Monday is always before the 8. day of the month and the second can't be before the 8. day.

[16] This feature was introduced in Reminder V1.20.

Note that if you want to be reminded about your event two days before the actual day, those remindings 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.

7.3.5 Event Text

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

7.4 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], [Month] and [Year] have all values and [Weekday] is `ANY`), the event list shows the calculated weekday of the event[17] 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[18]. The chronological order of repeating events is simply the order in which they will occur next time.

7.5 Adding New Events

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

7.6 Removing Events

Removing events is even easier. Select the event from the event list and click [Remove]. The event is removed from the event list and from the database.

7.7 Updating Existing Event Information

With [Update] 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.

[17] This feature was introduced in Reminder V1.10.
[18] This feature was introduced in Reminder V1.10.

Update any gadgets you want and then click [Update] . The event is now updated and the event list also shows the updated event information.

7.8 Starting a new event

Pressing [New] resets all event gadgets to their default values. This is a handy way to "start from scratch" when you enter another event.

7.9 Quitting REMINDER

There are three ways to quit REMINDER :

- Click [Quit and save] . The changes to the event database are saved and REMINDER terminates.
- Click [Quit, no save] . 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.

7.10 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] . 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`[19] 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).

[19] ©Jonas Pettersson

Command tells ReminderCheck to send to ARexx port specified in Port the command given in Comm/Script. This time no extra arguments are added.

NOTE: At the moment maximum length of text in Comm/Script is 28 characters, and maximum length of ARexx port name in Port 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).

7.11 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, pressing the key without shift-key selects next entry, pressing with shift-key selects previous entry.

There are two extra features:

- Weekday 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 ↑ and ↓ keys.

8 ReminderCheck

ReminderCheck can be started from shell and from Workbench. The best place for it is in your `SYS:WBStartup` drawer.

8.1 Workbench usage

ReminderCheck can be started from Workbench simply by double clicking its icon. In the icon you can define the database file[20] used by ReminderCheck with tooltype

```
FILE=filename
```

You can set the minimum time that has to pass before ReminderCheck reminds you again[21]. Default is to remind every time ReminderCheck is run. The interval is set with tooltype

[20] Default name is "`S:Reminder.data`".

[21] I added this feature because I happen to boot my machine quite often.

```
INTERVAL=minimum # of hours between two remindings
```

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 the ability of ReminderCheck 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 also include tooltype

```
DONOTWAIT
```

8.2 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[22] to be used. The 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 `RexxSysLib.library` and thus not use ARexx.

[22] Default name is "`S:Reminder.data`".

8.3 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** tells ReminderCheck that you have noticed the requester but want it to keep reminding you about this event and date[23]. For example, if the event says that you should pay your rent, you should use this button until you really have paid it.
- Acknowledge 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 event...` when you entered the event, the event is deleted from the database when you acknowledge it.
- Quit tells ReminderCheck to abort immediately.

8.3.1 Group Requesters

All the events which are marked "grouped" and whose date is "today" when ReminderCheck is run, are put into a single requester[24] (or to as few requesters as possible).

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.

8.4 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.

[23] Next time ReminderCheck is run, that is.
[24] This feature was introduced in ReminderCheck V1.20.

Note also that ARexx does not work if you have the `NOAREXX` tooltype in your ReminderCheck icon.

8.5 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**) is the default button which can be selected with Enter key also. You can use ESC key to select Quit .

9 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`[25]. These files are not included, but the complete ReqTools distribution can be found on many ftp sites and for example on Fish Disk #794.

9.1 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.

9.1.1 Changing the user interface with `GadToolsBox`

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.

[25] At least version 38 needed.

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 .

9.2 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.

9.2.1 Changing the user interface with `GadToolsBox`

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 .

10 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 fist 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 *Dvaid 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