**MFR**

| | TITLE : MFR | | |
|---|---|---|---|
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | | November 11, 2024 | |

**COLLABORATORS**

**REVISION HISTORY**

| NUMBER | DATE | DESCRIPTION | NAME |
|---|---|---|---|
| | | | |

# Contents

# Chapter 1

# MFR

## 1.1   MFR Help

```
==============================================================================
MagicFileRequester V2.0c Documentation                           15-Jul-1992
==============================================================================


                  (C) Copyright 1992 by Stefan Stuntz


   For informations about license and warranty see the file 'License'.



==============================================================================
                                Roadmap
==============================================================================


   1.      Introduction

           1.1.   Motivation
           1.2.   Features
           1.3.   Basics
           1.4.   System Requirements

   2.      Installation

           2.1.   From Workbench
           2.3.   From CLI
           2.3.   Removing MFR

   3.    Using MFR

           3.1.   Window Arrangement
           3.2.   Mouse Usage
           3.3.   Keyboard Usage
           3.4.   Menus
           3.4.1. Project Menu
           3.4.2. Edit Menu
           3.4.3. DOS Menu
           3.4.4. Settings Menu
```

## 1.2  MFR/Motivation

```
1.1. Motivation
---------------
```

A  file  requester  is  probably  the most used program inside an Amiga.
Unfortunately,  all  currently  available requesters lack some important
features, such as

– Keyboard support.
  Since  I  operate my computer mainly from an arm–chair with the keyboard
  on my knees, my mouse is far away.  I really need keyboard support.

– Font support.
  I  don't  see  a  reason  to force the user to a non proportional system
  default font or – even worse – to topaz/8 in a file requester.

– Resolution support.
  The  Amiga  has  many  different resolutions with different pixel aspect
  ratios.  What  do  I  want  with  a courier/13 font on a low resolution
~ 320x200 screen?

– Speed.
  This one's very important. Just look at 2.0 ASL and you know what I mean

These were the main reasons for me to write my own file requester.
MFR is the result and I hope, you will like it as much as I do.

## 1.3  MFR/Features

```
1.2. Features
-------------
```

- MFR is completely controllable via keyboard, so you will never have to touch your mouse when selecting files (of course you can, if you really want to).

- MFR will use any font you want in its display. Proportional fonts like helvetica look really marvellous.

- MFR has the ability to quickly search your hard disk for any file, so you needn't remember each file's directory.

- MFR offers a powerful directory caching mechanism.

- MFR can display pictures, play sounds, view archives, etc. with the aid of some external programs.

- MFR looks very nice, especially on high resolution screens.

- MFR has a sizing gadget.

- MFR takes advantage of new Kick 2.0 features like file notification.

- MFR can be customized through a wide range of tooltypes and menu items.

- MFR is fast.

- MFR is simply magic :-)


## 1.4 MFR/Basics

1.3. Basics
-----------

MFR is not distributed like other file requesters. You do not get an additional library for your LIBS: drawer and you may not call MFR directly from within your own applications. All you get is a single program that replaces some library vectors at installation time. So every program that uses one of the supported file requesters will use MFR instead.

MFR is especially designed for the new Kick 2.0 colors and resolutions. It will look best on a high resolution display with a pixel aspect ratio of nearly 1:1 (e.g. Hires + Interlace). Although MFR can be used on low resolution screens too, the 'feeling' won't be there.

MFR is a commodity. You can control it like other commodities with the Exchange program from your Workbench Disk.


## 1.5 Requirements

1.4. System Requirements
------------------------

As you probably noticed, MFR requires at least Kickstart 2.04 (also known as V37.175) to work properly. If you are still using Version 1.3 of the operating system, it's time to update now. You won't regret it.

Particularly, the following libraries are needed:

```
Library            Min.Version  Place

intuition.library    37        ROM
graphics.library     37        ROM
gadtools.library     37        ROM
utility.library      37        ROM
workbench.library    37        ROM
icon.library         37        Libs:
commodities.library  37        Libs:
asl.library          37        Libs:
iffparse.library     37        Libs:
diskfont.library     36        Libs:
req.library           1        Libs:
```

All these libraries are part of the OS 2.04 distribution, except req.library, which can be found in the distribution archive of MFR.

## 1.6  MFR/From Workbench

```
2.1. From Workbench
-------------------
```

Couldn't be easier. A simple double klick on MFR's icon will install it in memory. From now on, every ASL and REQ file requester will be replaced with a MFR. Of course you can (and I really hope you will) move MFR into "SYS:WBStartup/", so it will be launched every time you boot.

## 1.7  MFR/From CLI

```
2.2. From CLI
-------------
```

Since there seem to be quite many people out there, who do not use workbench, MFR can also be started from a CLI environment. The command mfr install will do this job. MFR automatically detaches so there is no need to – run – it.

## 1.8  MFR/Removing MFR

```
2.3. Removing MFR
-----------------
```

If you really want to remove MFR from your system, you can do this by
double clicking its icon a second time, by calling "mfr install" from
CLI again or with the Commodities Exchange program. A requester will
pop up informing you that MFR has been removed.

MFR will refuse to quit when there are some other programs still hanging
around with open or just with allocated file requesters. In this case,
MFR will inform you how many times it's locked. You will have to exit
all these locking programs, before you can remove MFR from your system.


## 1.9  MFR/Window Arrangement

3.1. Window Arrangement
----------------------

After installing MFR, you should try pressing <Control><LeftAlt><M>.
This key combination (of course adjustable) will pop up a MFR to play
with, so let's start playing...

MFR offers a window with a big list view for the directory on the left
side and a smaller one for devices and volumes on the right side. At
the bottom, you will find the drawer and file gadgets as well as a
pattern gadget. 'Okay' and 'Cancel' buttons are located where you would
expect them. You should be familiar with this outfit since it's almost
like the req.library requester.

Some words about the directory list now. On a standard workbench,
directories will appear white and files will appear black. Right of a
file entry, MFR shows its size, creation date and time and the file's
flags. The amount of information shown depends on the size of MFR's
window and on the width of the font. Try sizing the window to see how
it works.

The device list contains an alphabetically sorted list of all devices
(df0:, df1:, dh0:, ram:, ...), all volumes (Work:, Ram Disk:, ...) and
all assigned directories available in your system. Usually, you won't
need all of these entries while working with MFR, so MFR offers a
possibility to prevent certain entries from being displayed. More on
that later.

The widths of directory and device list are adjustable. Just click
between both lists (right of the directory scroll bar) and drag the
appearing bar around. If you size the device list big enough, some
device information will appear.

MFR allows customizing its complete outfit (Fonts, Scrollbars, Gadgets,
Background, etc.) according to your personal taste. Refer to section
 3.4.4. for details.


## 1.10  Usage

3.2. Mouse Usage
----------------


Using  MFR  with a mouse is not too much different from using other file
requesters.   To  select  a  file, just click on it and press the 'Okay'
gadget or just double click the file.  To avoid overwriting of important
files, the double click feature is disabled in save mode.

Entering  a  directory (or  a  device,  volume) is done by clicking and
releasing  the  left  mouse  button  on  the  appropriate entry.  If you
release the button somewhere else, nothing will happen.

The 'Parent' gadget (labeled '/') is located right of the drawer gadget.
Since  this  place  is  sometimes  far away from the directory list, MFR
offers another parent gadget.  It's located (invisible) between the left
window  border  and the left side of the directory list and acts exactly
like  the  '/'  gadget.   For  users  with a three button mouse, the mid
button can also be used to access the parent directory.

When the application allows selection of multiple files, you can do this
by  holding  down  the  <shift> key and clicking on the desired entries.
You  can  also  hold  down  <shift> and move the mouse (with left button
pressed)  around.   Note  that  you will have to hold shift even for the
first  entry  you  select.  This is different from other file requesters
behaviour  and  from workbench multi select, but it seems to be the only
reasonable way for MFR.

Maybe you already noticed the '!' gadget near the file gadget.  This one
will  cause  MFR to perform a so called file action on the current file.
Refer to section  3.4.3.   for details.


# 1.11   Usage


3.3. Keyboard Usage
-------------------


Using MFR with the keyboard needs a little more explanation.  First some
general things.

The  file gadget of MFR is connected with the directory list, the drawer
gadget  is  connected  with  the  device list. So if you have e.g.  an
active file gadget, you will also notice a cursor in the directory list.
Try this by pressing <TAB> or <Shift><TAB> to cycle through the gadgets.

Tab  cycling is a little bit different from standard kick 2.x behaviour.
Pressing  <TAB>  will  always  toggle  between  drawer  and file gadget,
<Shift><TAB> will activate the pattern gadget (if visible).

The  list cursor always moves according to your actions in the connected
string  gadget.   You should now try moving up and down in the directory
list  with  the  cursor keys (and maybe with some qualifiers).  MFR will
always  update the file gadget with the name of the entry under the list
cursor.  So if you want to select a file, just move the cursor on top of
it and press <Return>.  When the list cursor is placed over a directory,

MFR will append a '/' to the directory's name before updating the file
gadget. Thus, pressing <Return> over a directory will simply enter it.
The shortcut <Amiga><P> is used to change to parent directory.
Alternatively, you can use <Alt><Right> and <Alt><Left> to enter/leave
directories.

The same procedure applies to the device list. Activate the drawer
gadget, select a device by moving the cursor and press <Return>. Easy,
isn't it?

Now things get a little bit complicated, but more interesting. Imagine
you have a directory with lots of files (e.g. a directory with IFF
graphics) and you want to load the file 'Stefan_With_Amiga.pic'. Of
course you could try to reach the file with your cursor keys, but this
may take a while. Just try typing a 'S' into the (empty) file gadget.
You will notice the list cursor magically moving on the first file
starting with the letter 'S'. This could e.g. be
'Sabine_Without_Clothes.pic'. Now type a 't' and MFR will move to the
first entry with 'St' at the beginning, e.g. 'Stars&Stripes.pic'.
After typing an 'e', you will find yourself directly over
'Stefan_With_Amiga.pic'. Thus, you only needed three keystrokes instead
of many cursor downs or mouse moving. Due to MFR's auto completion, you
will only have to press <Return> now to select this file.

Remember, although your string gadget only contains 'Ste', MFR will
return the full name 'Stefan_With_Amiga.pic' to the application. If you
don't like this auto complete feature, you can of course switch it off.

Now, type some filename into the file gadget that does not exist in your
current directory (e.g. 'grmblwuz'), or just clear the gadget. You
will notice the list cursor changing from a bar to a rectangle. This
indicates, that no auto complete will take place when you type <Return>,
the file will be returned without any changes.

Multi selection of files can be done by the key combination
<Amiga><Space>. This toggles the selection state of the active file
(the one under the list cursor). Additionaly, <Amiga><Shift><Space> can
be used to select/deselect the entire directory.

MFR can also be controlled with the numeric keypad (if enabled). Key
'5' is for multiselection, the other keys work as expected.

Leaving MFR without selecting files can be done with the <Escape> key.
This is equivalent to pressing the 'Cancel' gadget.

## 1.12  MFR/Menus

3.4. Menus
----------

As every real Amiga program, MFR offers a couple of pulldown menus.
Here they are...

## 1.13  Menu

```
3.4.1. Project Menu
-------------------
```

- About

Display some information about program version, author and registration.
When you register, your name will be displayed here.

- Statistics

Display statistics about all previous file requester calls.  Seems to be
a  little  bit  senseless,  but  maybe  someone  is  interested  in  this
information.

- Flush Memory

Free  some  allocated memory.  With Workbench 2.1, this function will be
called automatically when memory gets low.

- Okay

Simulate 'Okay' gadget.

- Cancel

Simulate 'Cancel' gadget.

## 1.14  Menu

```
3.4.2. Edit Menu
----------------
```

- Copy Path

Paste  complete  current  path (concatenated contents of drawer and file
gadget) to clipboard.

- Copy File

Paste contents of file gadget to clipboard.

- Copy Dir

Paste contents of drawer gadget to clipboard.

## 1.15  MFR/Dos Menu

```
3.4.3. DOS Menu
---------------
```

- Parent Directory

Change to the parent directory. MFR will display an error in its
status line when there is no parent directory.

- Reload Directory

Load and display the current directory again. This is useful if you
turned of file notification for some reason and something in the current
directory changed.

- Rescan Devices

After mounting an additional device, you may wish to recreate the device
list. This menu item will do that for you.

- Search File

MFR has the ability to search your harddisk for some files. Before you
can use this feature, you will have to create a data base consisting of
all your directories and all your files. This can be done with the
"UpdateDB" program from Larry Phillips "Find" package, which is included
in MFR's distribution archive. If you are not familiar with "UpdateDB",
please refer to Larry's documentation for details. Once you have
created the data base, you can easily find any file you want. Just type
the (partial) name (or pattern) into the string gadget and select the
'Search File' menu item. MFR will list all occurances of the concerning
file together with the complete path in the file list. Just double
click on the desired entry and you will be moved to its directory with
the appropriate information filled in the drawer and file gadgets.

- File Action

MFR has the ability to show pictures, play sounds, view archives, etc..
If you want to use this feature, you will have to tell MFR what to do
with certain file types. This is done via the CLASS_0 - CLASS_9
tooltype entries.

The syntax of a CLASS entry looks like this:

CLASS_<n>=[<namepattern>],[<contentspattern>],<action>

If you select the "File Action" menu item or you press the file action
gadget (labeled '!'), MFR checks the current file against <namepattern>.
If this one matches, MFR checks the first 256 Bytes of the file against
<contentspattern>. In case of match, the command <action> is performed
on this file. Note that <namepattern> is case insensitive but
<contentspattern> is case sensitive. An ommited pattern matches always.

The parameter <action> must be a valid command, exactly as if typed into
a CLI. The first "%s" in <action> will be replaced with the full path
specification of the current file. Note that formatting of the command
is done via sprintf(), so you should be careful not to use other printf
style formatting commands such as "%d" or "%x". Your machine may crash
if you do so! If you use commands that print to the current console
(e.g. all archivers), you must use IO-redirection.

Some examples for file classes are:

```
CLASS_0=#?.(lha|lzh),,lz >con:120/50/500/250/LH-Archive/CLOSE/WAIT v %s
CLASS_1=#?.zoo,,zoo >con:120/50/500/250/Zoo-Archive/CLOSE/WAIT l %s
CLASS_2=#?,FORM????ILBM#?,display %s
CLASS_3=#?,FORM????8SVX#?,play %s
CLASS_4=#?.doc,,more %s
```

- History List

With 'History List', you will get a list of the last files you selected
with MFR. This can be useful if you want to load some files again. If
the history list is not empty, a window will pop up and allow you to
select one entry. Note that the entries of the history list are not
sorted, the last selected file always appears on top of the list.

- Make Directory

Create a new drawer in the current directory.

- Delete File/Dir

Delete a file or an empty directory in the current directory. Note that
MFR is no CLI tool, so you won't be able to delete non empty directories
or write protected files.

- Rename File/Dir

Rename a file or a drawer. You may move files out ot the current
directory buy specifying a complete path for the new name.

* Status submenu

The 'Status' submenu is used to control some status bits of MFR.
Normally, these bits are set to reasonable values by the calling
application, but you can override these settings at any time. Note that
none of these flags will be saved.

- Status / Display Pattern Gadget

Controls whether the pattern gadget is visible or not.

- Status / Disable Pattern

Controls whether the pattern specified in the pattern gadget shall be
used to hide certain files. This has nothing to do with the visibility
of the gadget.

- Status / Hide Icons

Controls whether icons (.info files) shall be displayed or not. Even
with icon display enabled, .info files may be hidden by the pattern
setting.

- Status / Match Directories

Controls whether directories must also match the specified pattern to be

displayed.  This is not useful in general.


## 1.16  MFR/Settings Menu

3.4.4. Settings Menu
-------------------

The 'Settings' menu  contains some items to customize MFR according to
your personal taste.  MFR's settings are global for all file requesters.
If you e.g.  select a new font, all open requesters will be informed and
updated.

Many  settings  can  hold two different values, according to the current
screen  resolution.  For  example, changing the gadget font on a lowres
screen  affects  the  lowres  setting,  changing  it on a highres screen
affects the highres setting.  These menu items will be marked as [dual].

Low  resolutions displays are:  Lores, Hires, SuperHires, HAM, LoresDPF,
LoresDPF2,  HiresDPF2,  SuperHiresDPF2,  ExtraHalfBrite,  VGAExtraLores,
VGAHAM, VGAExtraLoresLace, VGALoresDPF, VGAExtraLoresDPF2, VGALoresDPF2,
and  VGAExtraHalfBrite.  All  other  modes  will  be considered as high
resolution.

* Window submenu

The  'Window'  submenu  is  used to control size, position and outfit of
MFR's window.

- Window / Override Dimension

Normally,  the  dimensions of a file requester window will be set by the
calling  program.  However, MFR seems to need more space than other file
requester.  Thus,  it  may  be  a  good  idea  to  let MFR override the
application's  size  suggestion  and use its  own one.  With this switch
set, MFR will always use the configured size when opening its window.

- Window / Override Position

MFR  offers the ability to open its window either centered on the screen
or  centered  under  the mouse pointer.  To use these features, you will
have  to set this switch.  Without 'Override Position', MFR will use the
application specified position.

- Window / Centered Window

MFR's  window  will  always appear centered on the (visible part of the)
screen.

- Window / Moused Window

MFR's window will always appear under the mouse pointer.

- Window / Rastered Background [dual]

The  background  of MFR'S window will be rastered with the darkest color

of  the current palette.  In save mode, background will be rastered with
the two darkest colors.

- Window / Inverted Gadgets

Flip front and back pen of string gadgets.

- Window / Use as Default Dimension [dual]

Set MFR's default dimension to the current window size.

* Scrollbar submenu

MFR  offers  four  different scrollbar designs, each of them either with
symmetrical or asymmetrical placement of arrow gadgets.

- Scrollbar / Default [dual]

Scrollbars look like standard Kick 2.x scrollbars on highres screens.

- Scrollbar / Default-Lores [dual]

Scrollbars look like standard Kick 2.x scrollbars on lowres screens.

- Scrollbar / Goonie [dual]

Scrollbars  and  arrows  look  3-dimensional,  with a little knob in the
middle.  Nice!

- Scrollbar / Stuntzi [dual]

Scrollbars use PROPNEWLOOK, looks a bit like workbench's scrollbars.

- Scrollbar / Symmetrical

Scrollbar uses symmetrical placement of arrow gadgets.

- Scrollbar / Set Arrow Delay...

Here,  you  can adjust the repeat rate for the scrollbars arrow gadgets.
With a value of 0, MFR scrolls very fast.

* Font submenu

- Font / Select List Font... [dual]

A  font  requester  will  open up and let you specify the font to use in
directory  and  device  list. Proportional fonts are perfectly okay and
recommended, since they allow displaying of longer filenames.  Suggested
ist helvetica/11 on a highres display, but that depends on your taste.

- Font / Use Screen Font [dual]

With  this item enabled, MFR will always use the parent screen's default
font for displaying its lists.

- Font / Select Gadget Font... [dual]

A  font  requester  will  open up and let you specify the font to use in
MFR's  string gadgets.   Proportional  fonts  are  perfectly  okay  and
recommended, since they allow displaying of longer filenames.  Suggested
ist helvetica/13 on a highres display, but that depends on your taste.

– Font / Use Screen Font [dual]

With  this item enabled, MFR will always use the parent screen's default
font in its string gadgets.

* Sorting submenu

MFR is able to sort the directory list in many different ways.  The menu
items should be self explaining, if not, just try them.

– Sorting / Sort by Name
– Sorting / Sort by Date
– Sorting / Sort by Size
– Sorting / Dirs First
– Sorting / Dirs Last
– Sorting / Don't care
– Sorting / Low to High
– Sorting / High to Low

* DOS submenu

– DOS / Set Default Pattern...

When the calling application doesn't specify a pattern, MFR will use the
configured  default  pattern as initial contents for the pattern gadget.
For example ~(#?.info|#?.o|#?.bak) would be a good choice.

– DOS / Set Vol. Show Pattern...
– DOS / Set Vol. Hide Pattern...

MFR  allows  you  to  specify which devices/volumes should appear in the
device  list.   Just set some show and hide pattern to suite your needs.
Example:  Show Pattern = (dh#?|df#?|ram#?)

– DOS / Set History Entries...

This sets the maximum number of entries in the history list.  A value of
0 disables the history feature completely, saving a bit of memory.

– DOS / Set Cache Entries...

MFR  remembers  up  to  100 directory contents, considerably speeding up
operation.   With  this caching, you will be able to browse through your
directory tree without delay.  A cache is automatically invalidated when
the datestamp of the associated directory changes (this happens whenever
files are changed within that directory).  With 'Set Cache Entries', you
can  specify  the  maximum  number  of cached directories.  A value of 0
disables  caching completely.  Note that this change will not affect the
current  session.   You will have to save settings, exit MFR and restart
it to see the difference.

In case of memory shortage, MFR will free its caches. Therefore, it
hooks into asl.library's Expunge() function. Unfortunately, this
feature won't work with Workbench 2.0's asl library. But when Workbench
2.1 becomes available, it will magically start working.

- DOS / Set Read Timeout...

With 'Read Timeout' it's possible to make MFR multitask while reading a
directory. By default, its value is set to zero. In this mode, MFR
will read a directory completely before displaying anything and before
reacting to user actions. This is the recommended mode, since it's a
lot less confusing and since MFR's directory caching reduces waiting
time to a minimum. However, if you really want, you can set the timeout
to another value (timeout is measured in 1/50 seconds). Then, MFR will
update its display every time the period expires or when you perform
some action on the directory list (e.g. cursor movement). To avoid
messing up the display, new entries will be appended unsorted at the
bottom of the list. As you probably noticed, I don't like this mode and
it surely would need some more testing and programming. You really
should use a timeout value of 0 for now.

- DOS / File Notification

'File Notification' finally toggles MFR's file notification feature.
With file notification enabled, MFR automatically recognizes changes in
the displayed directory and reloads it immediately. Since this can be
really confusing sometimes, you can switch it off here.

* Search submenu

This submenu influences the way MFR compares a file with the entries of
the find data base. Note that when you specify an AmigaDOS pattern to
search for, only the 'Case Sensitive' menu item will affect matching.

- Search / Case Sensitive

When this item is selected, matching will be case sensitive.

- Search / Match Exactly

The file must match the data base entry completely, e.g. 'mary' won't
match 'primary' or 'marys_daughter'.

- Search / Match Beginning

All data base entries starting with the file gadget contents will be
displayed.

- Search / Match Partial

If the data base entry contains the file gadget contents somewhere, it
will match.

* Usage submenu

This menu contains various flags that control how MFR responds to user
actions.

- Usage / Auto Complete Load
- Usage / Auto Complete Save

A  file  requester  can  be in load or in save mode, as specified by the
application.   MFR allows you to toggle its  auto completion feature for
both  of  these modes.  Normally, 'Auto Complete Load' should be set and
'Auto Complete Save' should be unset, but this depends on your taste.

- Usage / Numeric Pad Control

Used to toggle cursor control via numeric keypad.

- Usage / Close after Copy

When  this  item  is  set, MFR's window will automatically close after a
copy operation to clipboard.

- Load Settings
- Save Settings

These  two  menu items (available only when you register), allow loading
and  saving of MFR's settings.  MFR will automatically load the settings
at  startup.  For that, it will try to locate its .info file and get the
configuration  out  of the tooltypes.  If you start MFR from CLI and the
.info  file  is  not  present, MFR  will look for an 'envarc:mfr.prefs'
config  file  and  take  its configuration from there.  In any case, MFR
remembers  the  configuration's  place  and  saves it to the appropriate
position when 'Save Settings' is selected.


## 1.17  MFR/Tooltypes

4.1. Tooltypes
--------------

Most  of  MFR's  configuration  can  be  adjusted  with  the menu items.
However,  there  are  some  things that haven't made their way to a menu
yet, so you will need to adjust them via tooltypes.

CX_POPKEY
defines  the  key sequence used to pop up a Magic File Requester.  Don't
know  if it's really useful to pop up a file requester, but if you want,
you can.

CX_PRIORITY
defines  MFR's  priority  in  commodities  input  chain.  There  should
normally be no reason to alter this value.

PATCHASL
turns  the  patch for asl.library on and off.  There should be no reason
to turn it off, I hope.

PATCHREQ
does the same thing for req.library.

```
PATCHREQTL
same for reqtools.library

MINPIXEL
defines  the minimum space reserved to display a file name.  MFR adjusts
the  amount  of  displayed  information  according  to the width of  its
window.  Try  resizing  the window to see how it works.  MINPIXEL pixel
will  always  be reserved to display the file name, unless the window is
too small.

REJECT
defines  a  case insensitive pattern for task names.  If you encounter a
program  that  does  not work with MFR, you can make MFR drop to the old
file requester when being called from this task.  Just find out the task
name and add it to the REJECT pattern.  You can use the program TaskList
(see Goodies section) to list all tasks currently running.

CLASS_0 - CLASS_9
Here  you  can  configure  some  file classes for file actions.  See the
description of the 'File Action' menu item for details.

FKEY_0 - FKEY_20
MFR  lets  you define 20 directories for quick access with the functions
keys.  You  can  specify a file name after the directory, in which case
the file gadget will be filled with this name.  F-Keys 11 through 20 are
the shifted F-Keys 1 through 10.

The  two tooltypes DONOTWAIT and STARTPRI are dedicated to workbench and
should be left untouched, unless you know what you're doing.

All  other  tooltypes  are  maintained  with MFR's menu items.  There is
absolutely no need to alter them by hand.
```

## 1.18  Interface

```
4.2. CLI Interface
------------------
```

MFR  has  a  CLI  interface  that allows using the requester from within
script files.  Calling MFR with the following template

```
  mfr [DIR=<dir>] [FILE=<file>] [PATTERN=<pattern>] [TITLE=<title>]
      [OKTEXT=<oktext>] [CANCELTEXT=<canceltext>]
      [LEFT=<left>] [TOP=<top>] [WIDTH=<width>] [HEIGHT=<height>]
      [MULTI] [MATCHDIRS] [SAVE] [PATGAD]
```

will  pop  up a file requester, wait for the user to select some file(s)
and  print  the  results  (full  path  specification) to stdout.  If the
requester is cancelled, nothing is displayed.

You  can  use this feature from a shell or arexx script, redirecting the
result to a file for further usage.  For example, the command

```
  mfr >env:result DIR=cap: PATTERN=*.cap TITLE="Select a capture file"
```

will create an environment variable named 'result' which consists of the
selected file.  If you don't select a file, 'result' will be empty.

MFR  will  return  0  if  a  file  was  selected, 5 if the requester was
cancelled.  A  return  code of 10 indicates an error (wrong parameters,
out of memory, ...).

Note that MFR will internally call the asl.library file requester, so if
you  didn't  allow  it  to  patch this library, you will just see an ASL
requester popping up.

## 1.19  MFR/UnArp

5.1. UnArp
----------

UnArp  is  a  tool by Martin Berndt, that tries to make arp.library more
compatible  to  Kickstart  2.0.  Many ARP calls are replaced by calls to
dos.library, the file requester is replaced by an ASL requester.  If you
put it somewhere in your startup-sequence or user-startup, you'll notice
MFR replacing all your old ARP file requesters.

## 1.20  MFR/TagScreens

5.2. TagScreens
---------------

With  Kick  2.04, there is a new method for programmers to specify which
colors  should  be  used for gadget and window rendering on their custom
screen.  MFR uses exactly this colors in its window.

Programs  that  were developed before Kick 2.04 was released, don't know
about  this feature.  Their windows and gadgets, including MFR will look
a little bit ugly.

"TagScreens"  from  Martin  Berndt  tries  to patch the old OpenScreen()
function  to  make  these  old  programs  use the new 3d look. Refer to
"TagScreens.txt" for details.

## 1.21  MFR/Find

5.3. Find
---------

"Find" is a tool for fast file searching, written by Larry Phillips.  It
works  in  conjunction  with a data base, including all your directories
and files.  MFR needs this data base for its search functions. Refer to
"Find.DOC" for details.

## 1.22  MFR/TaskList

```
5.4. TaskList
-------------
```

MFR  has the ability to prevent some specific tasks from using it.  This
is  hopeful  if you e.g.  encounter a program that doesn't work with MFR
or needs some special features of asl.library, not supported by MFR yet.

When  a  file  requester  is  called,  MFR compares the specified REJECT
pattern  with  the  name  of  the  calling  task  (ExecBase->ThisTask->
tc_Node.ln_Name).  In case of match, MFR will call the old requester.

To  find  out the name of the calling task, you can use a system monitor
such  as  "ARTM"  or "XOper" or the supplied utility TaskList.  TaskList
exactly  displays  the  task  name,  checked by MFR.  "ARTM" and "XOper"
display some other things sometimes.

Finding  out  the calling task's name may sometimes get a little tricky.
The  normal  task name of "MultiPlayer" is e.g.  "MultiPlayer", but this
program  creates  a  second  task  named "File Requester" to call a file
requester.  To prevent "MultiPlayer" from calling MFR, you would have to
specify "REJECT=File Requester".

## 1.23  Bugs

This  section  is  intended  to  avoid  unnecessary  bug  reports  and
suggestions.  Please read this before you write a mail.  But if you have
additional ideas, bug reports or just some comments, write me.

```
6.1. Known Bugs
---------------
```

- MFR won't hide its file gadget in a directory only requester.  This does
  not affect functionality.

- Still some missing ASL Tags.  Since they are not too important, MFR will
~ never drop to an ASL requester.

```
    ASLFR_Screen            /* Address of parent screen     */
    ASLFR_PubScreenName     /* Name of public screen        */
    ASLFR_Locale            /* Locale ASL should use for text */
    ASLFR_RejectPattern     /* preparsed reject pattern      */
    ASLFR_AcceptPattern     /* preparsed accept pattern      */
    ASLFR_TextAttr          /* Font to use for requester     */
```

- Missing reqtools tags:

```
    RT_IDCMPFlags
    RT_ReqPos
    RT_PubScrName
    RT_Screen
    RT_ReqHandler
```

```
    RT_DefaultFont
    RT_ScreenToFront
    RT_UnderScore
    RT_TextAttr
    RT_IntuiMsgFunc
    RT_Locale
    RTFI_AddEntry
    RTFI_RemoveEntry
    RTFI_FilterFunc
    RTFI_AllowEmpty
    Flags: FREQF_NOBUFFER, FREQF_SELECTDIRS, FREQF_DOWILDFUNC

    RTFI_VolumeRequest

    /* This one's making trouble. Why didn't Nico add an extra
       function rtVolumeRequest() instead of this strange tag?
       MFR will try to drop to reqtools, but this may be
       dangerous in some cases. */
```

- MFR is not totally clean against specifiying a too big font.  You should
  be careful experimenting with euro/90 or something like that.

## 1.24  MFR/Compatibility

```
6.2. Compatibility
------------------
```

I  tried  hard to make MFR as compatible as possible.  However, patching
libraries  has  always  been  and will always be a little bit dangerous.
You  should  assure  that no other programs patch file requester library
vectors  when  MFR runs.  If you e.g.  use 'ReqChange' (V1.09, by Magnus
Holmgren)  to forward some string/number requesters to reqtools.library,
you should specify "reqchange NOASL NOARP" to start it.  In general, MFR
should  be  the  last  started requester patch tool.  To accomplish that
with 'WBStartup/' files, you can e.g. use the STARTPRI tooltype.

## 1.25  Features

```
6.3. Future Features
--------------------
```

Here are some plans for future expansion of MFR.  Comments and additions
welcome.

- A  hotkey  to  paste MFR's gadgets contents to the input stream would be
~ nice.

- Drawing of the rastered background could be optimized.

- Entries in the find list tend to become very long.  Since the right part
  of a path is usually more interesting than the left part, I should maybe
  display it this way.

– Once file classes are supported by the system software, they will be
~ taken out of MFR. Hurry up, Commodore :)

– Localization. Ever wanted a norwegian file requester?

– Replacing of some more libraries (arp, rct, kd_freq). Well, UnArp does
~ his job well, rct and kd_freq are rarely used...

– Better handling of the zoom gadget, including complete iconification and
~ adjustable zoom size.

– Some more multi select hilight styles (italics, different colors, ...).

– ... to be continued ...

## 1.26  MFR/Acknowledgements

6.4. Acknowledgements
--------------------

Finally, I want to thank


Andreas Schildbach  for many good ideas, for debugging, for some artwork
                    and for the pacman. :-)

Stefan Becker       for debugging CLI interface and eliminating
                    some enforcer hits.

Martin Berndt       for debugging resolution detection and for the
                    permission to include his programs "TagScreens"
                    and "UnArp".

Stefan Hochmuth     for debugging memory allocation system.

Bryan Ford          for his startup code.

Michael Berger      for the set of Lores Icons.

Ralf Kemle &        for some more ideas and for testing MFR with
Oliver Kilian       good old 68000er.

Oliver Vaross       for additional debugging.

Larry Phillips      for his permission to include the "Find" package.

Colin Fox &         for req.library. I only need RealTimeScroll()
Bruce Dawson        but this one's really nice.

Maximilian Hantsch  for some useful code fragments and for their
& Martin Laubach    trapdoor license agreement. I cloned it nearly 1:1.

Stefan Boberg       for the LhA documentation. I cloned the part about
                    registration for MFR. Hope he doesn't mind.

```
Ludwig Gritsch        for the AmigaGuide manual.

...                   and all the other guys and girls who reported bugs
                      and made suggestions.
```

## 1.27  MFR/How to reach the author

```
6.5. How to reach the author
----------------------------

If you have some additional ideas and/or bug reports, feel free to send
a letter (e-mail preferred) or to give me a call.


You can reach me via


 Internet: stuntz@informatik.tu-muenchen.de

  Fidonet: 2:246/46.10 (Stefan Stuntz)

   Z-Netz: S.STUNTZ@AMC.ZER

SnailMail: Stefan Stuntz
           Eduard-Spranger-Straße 7
           D-W-8000 München 45
~          Germany
           Phone: +49-(0)89-313-1248



That's all for now... Goodbye, and thanks for all the fish...
                                                          Stefan


                              ||
                              \/
```

```
   Remember: Always expect the unexpected and don't panic :-)
   --------------------------------------------------------
```