

01ca0398-0

David Ermac

COLLABORATORS

	<i>TITLE :</i> 01ca0398-0		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	David Ermac	March 1, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	01ca0398-0	1
1.1	MBench Help	1
1.2	MBench Overview	2
1.3	Registering MBench	2
1.4	Installing MBench	3
1.5	Running MBench	3
1.6	Technical Support	4
1.7	Bug Reports	4
1.8	Credits	5
1.9	Copyright	5
1.10	Author	6
1.11	MBench History	6
1.12	Using MBench	11
1.13	Layout	11
1.14	New Features	12
1.15	MB Extra Commands	13
1.16	MBCommand	13
1.17	MBInfo	13
1.18	Open Location Window	14
1.19	Execute Command Window	14
1.20	Execute File Window	14
1.21	Select Pattern Window	15
1.22	Information Window	15
1.23	Rename Window	15
1.24	Menu	16
1.25	ARexx	18
1.26	Examples	19
1.27	Command List	20
1.28	About	23
1.29	Clean Up	23

1.30 Close Window	23
1.31 Disk Copy	24
1.32 Duplicate	24
1.33 Delete	25
1.34 Execute Command Window	25
1.35 Execute File Window	25
1.36 Empty Trash	26
1.37 Flush Buffers	26
1.38 Format Disk	26
1.39 GetSelectedList	27
1.40 Help	27
1.41 Information	28
1.42 Last Message	29
1.43 Leave Out	29
1.44 Open Assign Window	29
1.45 New Drawer	30
1.46 Open Open Location Window	30
1.47 Open Location	31
1.48 Open	31
1.49 Open Parent	32
1.50 Put Away	32
1.51 Quit	32
1.52 Rename	33
1.53 Redraw All	33
1.54 Run	34
1.55 Snapshot All	34
1.56 Show All Files	35
1.57 Set Backdrop Mode	35
1.58 Select Contents	36
1.59 Snapshot	36
1.60 Sort List	36
1.61 Select Next Window	37
1.62 Select Previous Window	37
1.63 Scroll Window Bottom	38
1.64 Snapshot Window	38
1.65 Scroll Window Down	39
1.66 Scroll Window Far Left	39
1.67 Scroll Window Far Right	40
1.68 Scroll Window Left	40

1.69	Scroll Window Page Up	41
1.70	Scroll Window Page Down	41
1.71	Scroll Window Page Left	42
1.72	Scroll Window Page Right	42
1.73	Scroll Window Right	43
1.74	Scroll Window Top	43
1.75	Scroll Window Up	44
1.76	Update All	44
1.77	Update	45
1.78	UnSnapshot	45
1.79	WBRun	45
1.80	Windows Horizontal	46
1.81	Windows Stacked	46
1.82	Window To Back	47
1.83	Window To Front	47
1.84	Windows Vertical	48

Chapter 1

01ca0398-0

1.1 MBench Help

MBench V1.0

Official MBench guide written by Dave Ermac
(C) Copyright 1996-1197 Mark Hewitt

Welcome, and thank you for using MBench,
the new workbench replacement for your Amiga.

Overview

What is MBench.

Registration

Obtaining the full version.

Installation

How to install MBench.

Running MBench

How to run MBench

Technical support

Getting help.

Using MBench

Using MBench.

Bug reports

How to report bugs.

Credits

Thanks to.....

Copyright

All the boring legal stuff.

The author
About the author.

History
Version history of MBench.

1.2 MBench Overview

MBench Overview

MBench is designed to directly replace the workbench graphical interface to the Amiga's filing system. It is 100% compatible in operation and functionality but has a number of more advanced features that stand out.

Some of the most notable features are:-

- Internally multitasking, load files, open windows, copy files etc... all at the same time.
- System wide file notification. Delete etc. files in one program and watch as MBench updates the display.
- User definable quick start menu for launching programs or scripts. Just drag icons to the SYS:StartMenu drawer to add new items.
- Cached directories.
- More user friendly, progress requesters and full context sensitive online help.
- Improved list view modes, sort items by clicking on the headers.
- Arexx support to allow easy and powerfull expansion.
- Works on any Amiga running OS2.04+.
- Optional full display of files in list mode, so you can manipulate just icons, files or both.
- Floating tool bars.
- Fully font sensitive.
- Keyboard short cuts.
- Directly open a window for any location.
- Enhanced information requesters.
- History buffers in execute file, open location, ... windows.
- Modular approach to allow easier upgrades and memory usage as only modules that are currently needed are loaded.
- Context sensitive help, including floating help bubbles.
- Localised support.
- WBStartup configuration window. Hold down keys to pop up a window or cancel all WBStartup programs.
- New transparent icons.

1.3 Registering MBench

Registering MBench

MBench is shareware. The unregistered version of MBench has the following limitations:-

- You can only load up two programs, not including those loaded in the WBStartup drawer.

The registration fee is £10 UK pounds + package and postage (see below). Register users will receive a personalised key file that will enable all the features of MBench. This personalised key file will also work with updates to MBench which will be freely available to registered users.

How To Register

Print out the file order form(or write using the same format), fill in the details and send it with the registration fee including P&P to me. The key file will then be sent to you via e-mail or floppy disk as you request.

Orders should be sent to:

MBench Registration
Church Farm House
Cublington Road
Aston Abbotts
Bucks
HP22 4NB
ENGLAND

Any cheques or postal orders should be made payable to - Mark Hewitt

Package and Postage costs

To the registration fee please also add the following amounts:

e-mail - Free
UK - £1.50
Europe AirMail - £2.50
Rest Of World AirMail (Zone 1) - £3.00
Rest Of World AirMail (Zone 2) - £3.50
Rest Of World (Surface Mail) - £2.50

1.4 Installing MBench

MBench Installation

To install MBench simply run the provided install script. Full help is provided on the installer screen.

1.5 Running MBench

Running MBench

The installation script will automatically cause MBench to be loaded instead of workbench from the disk it was installed to. MBench can also be run manually from the CLI with the following format:

Format:

MBench [PUBSCREEN] [SETTINGS] [PORTNAME] [STARTUP] [CX_PRIORITY] [CX_POPKEY]

Template:

PUBSCREEN/K, S=SETTINGS/K, P=PORTNAME/K, STARTUP/K, CX_PRIORITY/N/K, CX_POPKEY/K

Specification:

All the additional parameters to MBench are optional. The meaning of each of these parameters is as follows.

PUBSCREEN=<Screen> where screen is the name of the public screen to open on. If this screen can't be found, MBench will open on the default screen.

SETTINGS=<Filename> where filename is the name of a configuration file for MBench to use.

PORTNAME=<Name> where name is the required name for MBench's Arrex port.

STARTUP=<Filename> where filename is an Arrex script to execute on startup.

CX_PRIORITY=<Priority> where priority is the commodities priority to run MBench at.

CX_POPKEY=<Description> where description is the key press template to bring MBench to the front.

Example:

MBench PUBSCREEN=MainScreen SETTINGS=MBench.config

1.6 Technical Support

Technical Support

If you have any problems then please write to either of these address's. (If a reply is needed then please include a stamped addressed envelope)

Mbench Support
Church Farm House
Cublington Road
Aston Abbotts
Bucks
HP22 4NB
ENGLAND

e-mail:cs95mah@exeter.ac.uk

1.7 Bug Reports

Reporting Bugs

If you come across any bugs then please let me know details of the problem. Please include as much detail as possible, including machine type, expansions and the software you are using etc. Then send a report to me at

this address, and if need be changes will be made available and a new version distributed.

None limitations and bugs include:

- Scrolling isn't too good.
- Untested that well on GFX boards.

1.8 Credits

MBench Thanks

I would like to thank the following people for their support and help in developing MBench

Dave Ermac who has written the MBench guide you are reading now and for the ideas and help he has injected into the software through out the developing stages, such as designing all the graphical elements of this software. Sarah Bond for initial beta testing and her great support during the project

All those who sent brilliant ideas, suggestions and bug reports of who there are to many to name but their help was greatly appreciated.

Everyone who has registered MBench and made it the success it now is.

You for using MBench.

And Amiga for being the best computer ;-)

1.9 Copyright

Copyright Information

MBench is Copyright (c) 1997 by Mark Hewitt but may be freely distributed in a non profit way provided that the full distribution is included in an unmodified form.

Dissassembly or modification of the code is strictly prohibited.

Use on magazine coverdisks or other commercial distribution by compact disk or otherwise requires written permission from the author.

The KeyFile that is provided for registered users however may NOT be transmitted or reproduced in any form and any attempt to do so will be viewed as copyright theft and prosecuted.

If you want any further information then please contact me.

1.10 Author

MBench's Author

MBench is written by Mark Hewitt. He can be contacted at the following address's:

E-mail
cs95mah@exeter.ac.uk

WWW
<http://www.geocities.com/SiliconValley/Lakes/1258/mbench.html>

Snail mail (any replies needed, except registrations should include a SAE)

MBench
Church Farm House
Cublington Road
Aston Abbotts
Bucks
HP22 4NB
England

1.11 MBench History

MBench History

23-June-1997

- Installer script finished.

11-June-1997

- Bug fixed that would leave unfreed memory for icon selected images.

10-June-1997

- Leave Out/ Put Away finally work and record the correct values to disk.
- Bug fixed where in list mode the vertical scroller would show the wrong line.

09-June-1997

- Disk and memory update now takes place at a user definable interval.
 - Fixed bug in list mode that would cause memory trashing when you dragged an item where the name was longer than the name column.
 - Set the screen title to a reasonable default.
 - Fixed bug where duplicating a drawer would rename all its contents to copy_of_... aswell
 - Leave out bug fixed.
 - Default tool field fixed in project info requesters.
 - Fixed bug that could cause a crash if a projects default tool field was invalid.
 - Added notification on all start menu items, inc. sub drawers.
 - Fixed notification bug that could cause a crash.
 - Fixed bug that would cause MBench to crash if you changed the pattern from a picture to a pattern and then changed it again.
 - Clicking on directories is now more intelligent. New drawers open in the
-

same mode as the parent one. And opening up a buffered directory opens it back in the same mode that it was opened in before.

- Fixed bug with MFS (MultiFileSystem) that would cause MBench to hang.

08-June-1997

- Horay!! Finally finished my exams.
- Fixed icon positioning problems. Now consideration is made to the icons text and other bits fixed.

02-June-1997

- More fixes to disk detection, code prepared so that regular checks for new/unmounted/busy etc. disks can be made without having to reinsert a disk for any changes to be detected.
- Option to only unmount disks where they don't have a cached directory on them.
- Oops - pen locking didn't work. It works properly now - fixes some of the icon colour problems.
- Fixed bug where if a project icon's tool wasn't a full path, it wasn't found, it correctly searches the path list now.
- StartMenu items now have » rather than >> to show submenus. Fixes a problem with MagicMenus.
- Workbench windows now declared as such - allows, gauge bar from MCP et al. to work properly

01-June-1997

- Reworked the way that disks are detected. Now things such as the ENV: icon for HappyENV no longer show up. Should work more like Workbench now.

31-May-1997

- Added more options to the title format.
- You can now close all windows (except the main one) by holding down shift when clicking on a window's close gadget.

30-May-1997

- Option to lock pens to give correct colours.
- Option to format the title bar to your own preferences added.
- Semaphore added to menus.
- Startup bug fixed - if there already exists an arexx port called MBench, MBench will now detect this and load properly

28-May-1997

- Bug fixed that could try and unload a program twice if it failed to load properly.
- Bug fixed - if you clicked on a project icon and the default tool was invalid then MBench would crash on exit trying to free a lock twice.
- Bug fixed - added a semaphore to the commodities which should avoid some crashes.

27-May-1997

- Serious bug fixed now uses SYS: as the current dir always like Workbench this stops some programs crashing on exit.

26-May-1997

- Copy bug fixed.
- You can now enter up to 300 chars in input windows - the limit before was way too low.

-

25-May-1997

- Major fixes to the Information requester - now works OK when icons with no file are used and also enforcer hit produced by information on drawers fixed.
- Enforcer hit when copying just an icon fixed.
- Bug fixed that would not update an icon when there is no file for it.

24-May-1997

- Enforcer hit fixed when selecting an AppMenu item with nothing selected.

22-May-1997

- Fixed a major bug that would leave a lock on any drawer that you double clicked upon using up memory and not allowing you to delete it.

21-May-1997

- Bug fixed where MBench would leave a lock on a newly created icon.
- When deleting files etc. MBench will now correctly try and close/free any locks that it holds on an item.
- Enforcer hit with App Routines when passing a Disk WBArg fixed.
- Bug fixed in DICE startup code that would cause an enforcer hit when exiting MBPrefs. NOTE TO ANYONE USING DICE - The main startup code contains a major bug - the directory is changed to the first lock in the WBArg list and not changed back.
- Pop up menus now correctly make the window that they are clicked over the activate one (ie. the one that operations will take place in).
- Fixed enforcer hit when copying directories.
- Fixed enforcer hit when deleting items from a buffer with no associated window open.
- Fixed bug that failed to report any errors if delete failed.

20-May-1997

- Copy of files majorly sped up.
- Snapshot of windows now records the type (icon/list and show all flag.)
- Bug fixed that would cause an enforcer hit if you drag a disk item over an app window.

19-May-1997

- Copy routine when icons copied sped up - No longer has to reread the icons.
- Snapshot files bug fixed that could cause crashing.
- NewDrawer memory loss bug fixed.
- Snapshot etc. of NewIcons no longer loses the imagery until updated.

18-May-1997

- Bug fixed that could cause MBench to hang when a disk was inserted.
- Popup menus added for windows (registered users only).

14-May-1997

- Prefs program finished - needs testing.
- Notification of prefs change added - currently can't change some settings though.
- File and open location windows are now app windows.

11-May-1997

- Not really been keeping a note - had a lot of work etc on.
 - Select by pattern added.
 - Bug fixing display problems with the main window fixed eg. when you make
-

a new drawer in the main window.

- Changing the main windows backdrop mode now correctly redisplay the icons when it is done.
- started to add pop up menus for windows - at the moment they are very basic.

27-Apr-1997

- minor bug fixed that sometimes didn't allow you to load any programs in the demo version - was due to the async loading of programs.

23-Apr-1997

- Serious bug that was causing problems with KingCON and a few other programs fixed.
- fixed bug causing a crash when you flush libraries while MBench is running.

20-Apr-1997

- Icons routines changed, bug that could cause enforcer hits and possible crashes fixed.
- Now only changes its name to Workbench if Workbench isn't already running - it has to do this as some programs eg. DICE assume that if they get a Workbench Message at start up then they were started by Workbench - WRONG!!!
- Added flush buffers to the MBench menu.

18-Apr-1997

- Clicking in an empty list window no longer produces an enforcer hit.
- You can drag icons to leave out into the main window and they can now be renamed, deleted etc. just like normal icons can.
- left out icons now have a symbol in the bottom corner so that they can be distinguished from icons in the Workspace directory.

17-Apr-1997

- Clickin the right mouse button while the left is pressed over an item can now be used to cancel selection of that item.

16-Apr-1997

- Info requesters now have working shortcuts.
- shows busy etc. disks OK.
- left out items don't have to have a disk inserted for them to be used now.

15-Apr-1997

- Now correctly uses directory buffers - the maximum number of buffers to cache can be user defined in the settings.
- Fixed problem when you duplicate disk icons - now correctly brings up the disk copy program.
- Icon text is now displayed according to the font prefs settings.

13-Apr-1997

- fixed bug where left out drawers can't be opened.

12-Apr-1997

- correctly updates scroll bars when items are dragged in the same window.

11-Apr-1997

- fixed bug causing hang after Open is selected.
 - started rewrite of internal representation of directory lists.
-

10-Apr-1997

- Finished the help file.

9-Apr-1997

- updated scrolling of icon routines. They now work slightly better.

8-Apr-1997

- a memory loss bug removed.
- information requesters now correctly show information on disks, not always showing SYS:
- started to move some routines into mbmisc.library so that a number of internal features of MBench can be accessible to external programs even when MBench isn't loaded eg. progress windows, line input windows.
- you can now drag icons into the main widows although not as yet leave them out:-(. You can however drag them over disks, appicons and have an option to copy them to the Workspace: assign, in effect leaving them out To leave out items, you have to revert back to Workbench.
- the 'O' key was used twice in the menus. Open now uses '0' as its shortcut (O, P, E, N were all already defined.)
- clicking with various qualifiers held down doesn't register as a mouse click anymore so things such as powersnap can still use there default hotkeys.

6-Apr-1997

- Bug causing possible crash in rename routines fixed.
- Possible deadlock bug fixed in Close file detection routines.
- Added GetSelectedList command so arexx programs can obtain a list of currently selected files.
- You can now drag icons on to the main window although not as yet leave them out.
- Moved the E key back to execute file like on workbench. Even I kept getting confused by this.

5-Apr-1997

- Loading up a windows contents is now quite a bit faster.
- cleanup now takes into account the icons width.
- copying an icon to list mode where we show only files no longer displays the icon in that mode.
- clicking a drawer in list mode now correctly uses the drawers size and position.
- In the root directory icons are no longer saved as <name>disk.info if there doesn't exist a file along with the icon.

4-Apr-1997

- Added a KeyFile and a need to register to receive all full functionality. Sorry but I need the money being a poor student and all that ;-)
- Works on my A600 now aswell.

3-Apr-1997

- fixed bug where you are unable to select list header items when the list header is scrolled past the first item.
 - fixed alert when executing an internal command.
 - fixed bug where you sometimes loose memory when you switch display mode or close a window while the items are being loaded up into it. Also results in slightly faster item loading.
 - added the routine to detect renaming of disks.
-

- updated the routine to detect the sizes of files after they are changed now works slightly quicker and better.

2-Apr-1997

- List headers added and working.
- sped up display of assign window.
- added layout windows by Stacked, Horizontal, Vertical commands.
- added activate next, previous window commands.
- added send window to back, bring to front commands.

30-Mar-1997

- bug fixed. You can now select all items in the assign window. Even the top ones!
- Bug causing a possible hang when refreshing windows fixed.
- Window refreshing sped up. This is at the expense of having the windows not refreshed sometimes for a while but c'est la vie or whatever.
- Started to add the list headers for the list view mode. Still not working, but shouldn't be long now.

29-Mar-1997

- bug causing hang when main window updated fixed.
- bug causing alert in MBCCommand fixed. Was due to wrong value passed into ReadArgs.

28-Mar-1997

- when changing display (list->icon etc.) the border is no longer slightly overwritten.
- when dragging list items they now start off in the correct places.

27-Mar-1997 V0.9

- Loads a changes since the last release.

1.12 Using MBench

Using MBench

Screen Layout How it all looks.

New Features

Using the new features.

The Menus

The menus and what they do.

ARexx

Using MBenchs ARexx port.

Commands

Commands available to ARexx.

1.13 Layout

Layout

When you load up MBench you are presented with the main window which shows all inserted disks, left out icons and all icons in the Workspace assign. At the top of this window is a menu bar with available commands and options.

1.14 New Features

New Features

There are several different ways in which MBench has enhanced existing Workbench features or created new, exclusive features that only appear on MBench.

Menus

- Extended keyboard shortcuts
- A customisable Quick-start menu showing a list of MBench commodities and user defined items. The menu items are read from MBench:StartMenu and take their image from the icon and executes the file specified in the icons default tool where selected.

Layout

- MBench provides additional commands to enable you to arrange the windows in MBench. These are all located in the windows menu.
- The list mode has headings so you can sort the lists by clicking on the heading

Over All Operation

- All operations are now executed in a multitasking fashion so you can perform various tasks at once.
- The 'Assign List Window' shows a list of all current assigns so you can quickly open a window for a specified location.
- Online Help.
- Open a window directly for any location.
- Copy and delete operations have progress requesters.
- WBStartup window. Hold down left amiga at startup to skip loading of WBStartup programs or left amiga and left shift to bring up a window so you can select which items to run.
- Workspace: assign icons displayed in the main MBench window.
- Use middle mouse to extended select items.
- Press shift CloseWindow to close all windows in one go.
- Select files by a pattern.

ARexx and Commands

- MBench features an
ARexx
port.
- Execute
commands
via a window in MBench.

1.15 MB Extra Commands

Extra Commands

The following are additional programs provided with MBench that can be used to give additional functionality. These programs are:

MBCommand

To execute MBench commands.

MBInfo

To show an MBench information requester for a file.

1.16 MBCommand

MBCommand

The MBCommand command allows you to get MBench to execute one of its internal commands from the CLI

MBCommand can be run with the following format

Format:

MBCommand [COMMAND]

Template:

COMMAND/F/A

Specification:

The COMMAND is required and specifies one of MBench's internal commands to execute.

Example:

MBCommand OpenLocation System:Prefs

1.17 MBInfo

The MBInfo command allows you to access MBench's built in information requester from the CLI.

MBInfo can be run with the following format

Format:

MBInfo [NAME] [PUBSCREEN]

Template:

NAME/A,PUBSCREEN/K

Specification:

The name of a file is required and the public screen to show it on is

optional.

NAME=<Name> where name is the name of a disk or file to get information off. PUBSCREEN=<Screen> where screen is the name of the public screen to open on. If this screen can't be found, MBench will open on the default screen.

Example:

```
MBInfo System3.0:Prefs PUBSCREEN=Cinema.screen
```

1.18 Open Location Window

The Open Location Window

The Open Location Window is where you can type the name of a directory that you want a window opened for by MBench.

The window is a standard line input window. You enter the name of the directory you want a window opened for into the string gadget and then press either 'Icon' or 'List' to open a window for the directory, in either list or icon mode, or cancel to close the window without doing anything.

To the right of the string gadget is a small icon that can be clicked upon to bring up a file requester from where you can select a directory.

Full keyboard shortcuts are available and when the string gadget is selected, pressing the up and down arrow keys will scroll through any commands you have previously typed.

1.19 Execute Command Window

The Execute Command Window

The Execute Command Window is where you can type commands to be directly executed by MBench.

The window is a standard line input window. You enter the name of the command you want to execute into the string gadget and then press either 'OK' to execute the command or cancel to close the window without doing anything.

Full keyboard shortcuts are available and when the string gadget is selected, pressing the up and down arrow keys will scroll through any commands you have previously typed.

1.20 Execute File Window

The Execute File Window

The Execute File Window is where you can type the name of a file that you want to run.

The window is a standard line input window. You enter the name of the file you want to run into the string gadget and then press either 'OK' to run the program or cancel to close the window without doing anything.

To the right of the string gadget is a small icon that can be clicked upon to bring up a file requester from where you can select a file name.

Full keyboard shortcuts are available and when the string gadget is selected, pressing the up and down arrow keys will scroll through any commands you have previously typed.

1.21 Select Pattern Window

The Select Pattern Window

The Select Pattern Window is where you can type the a standard Amiga wildcard pattern for files that you would like selected.

The window is a standard line input window. You enter the pattern to match by into the string gadget and then press either 'OK' to select files in the current window or cancel to close the window without doing anything.

Full keyboard shortcuts are available and when the string gadget is selected, pressing the up and down arrow keys will scroll through any commands you have previously typed.

1.22 Information Window

Information Requesters

Information requesters have a number of additions.

These include:

- Keyboard shortcuts to gadgets.
- They are app windows so you can drag icon over various parts of the window to use features of existing icons. The parts that you can drag icons over are:
 - The image to copy the image from a new icon.
 - The tool types array to copy the tool types from an existing icon.
 - The default tool to cope an icons default tool.
- The windows also feature a file select tool next to the default tool gadgets which you can select to bring up a file requester to select the default tool from.

1.23 Rename Window

The Rename Window

The Rename Window is where you can type the name of a file that you want to be renamed.

The window is a standard line input window. You enter the new name of the specified file into the string gadget and then press either 'OK' to rename the file, 'skip' to skip to the next item to be renamed (if there is one) or cancel to close the window without doing anything.

To the right of the string gadget is a small icon that can be clicked upon to bring up a file requester from where you can rename another file.

Full keyboard shortcuts are available and when the string gadget is selected, pressing the up and down arrow keys will scroll through any commands you have previously typed.

1.24 Menu

```
V
  MBench
  Windows
  Icons
  Tools
V
```

```
-----
Commodities >> <list of running commodities>
~~~~~
<User Definable>
```

MBench

```
-----
      Backdrop          B
      Execute Command... E
      Execute File...   F
      Open Location...  O
      ~~~~~
      Open Assign Window W
      ~~~~~
      Flush Buffers
      ~~~~~
      Redraw All
      Update All
```

Last Message
~~~~~

Help

About  
~~~~~

Quit Q
Window

New Drawer N

OpenParent P
Close C

Update U
~~~~~

Select

>>  
Select Contents A

By Pattern

Clean Up L  
Snapshot >>  
Window

All  
~~~~~

Layout

>>
Stacked

Horizontal

Vertical
Select >>
Select Next ,

Select Previous .
Order >>
Window To Back <

Window To Front >
View By >> Icon 1
List 2

Show All Icons H
Icons

Open 0

Duplicate D

Rename R

```

Information  I
Snapshot    S
UnSnapshot
~~~~~

Delete

Format Disk
Empty Trash

```

Tools

1.25 ARexx

Using Arexx and the Commands.

One main benefit that MBench has over the standard Workbench is the addition of an Arexx port. This allows commands to be executed externally from the application, but MBench also provides a window so that you can type these directly into the program.

Via Arexx

To execute commands via Arexx you should send commands to the Address 'MBench.1' which will always be the Arexx port name for MBench. Follow this link for a list of examples.

Internally

The other method is to select the Execute Command menu item from the MBench menu. This will bring up another window into which you can directly type commands for MBench to execute. Not all commands are available when executed via this method, most noticeably those which are used to return a value or result.

eg. Normally if you wanted to close the window relating to Work:MBench/Arexx/Examples you would click on the close gadget, In MBench however you could open the command window and enter:

```
CloseWBWindow Work:MBench/Arexx/Examples
```

Not really a very usefull application of the commands but it goes to show what I mean.

See Also.

List Of Commands

Examples

1.26 Examples

Arexx Examples

Below follow a couple of examples that show how you can use MBenchs Arexx port.

- About.rexx simply pops up MBenchs about requester.
- OpenSYS.rexx simply opens the SYS: window.
- ListSelected.rexx shows a list of all selected files in MBench. This could be extended to then perform some operation on these files.

```
/*
 * File: About.rexx
 *
 * Shows how to pop up the about window...
 *
 * You need to run MBench first...
 */ Options FailAt 100
```

Options Results

Address MBENCH.1

```
say "Welcome to this demo of the MBench Arexx Port."
say "This example shows you how to pop up the about"
say "window from an arexx program"
```

About

```
/*
 * File: OpenSYS.rexx
 *
 * Shows how to open the SYS window...
 *
 * You need to run MBench first...
 */ Options FailAt 100
```

Options Results

Address MBENCH.1

```
say "Welcome to this demo of the MBench Arexx Port."
say "This example shows you how to open a window for"
```



```
say "a location from an arexx program"
```

```
OpenLocation SYS:
```

```
/*
 * File: ListSelected.rexx
 *
 * Shows how to retrieve a list of selected files from MBench...
 *
 * You need to run MBench first...
 */ Options FailAt 100
```

```
Options Results
```

```
Address MBENCH.1
```

```
say "Welcome to this demo of the MBench Arexx Port."
say "This example shows you how to read a list of"
say "selected files from an arexx program."
```

```
/* show a list of selected files. */
GetSelectedList "RAM:RexxTemp"
OPEN("selected", "RAM:RexxTemp", "R")
DO UNTIL EOF("selected")
  nextname = READLN("selected")
```

```
/* now print the list out or whatever you want to do with them.
 * One idea is to add an item to the start menu eg. LHA archive selected
 * and have that run an arexx script the could get a list of selected
 * files and then create an archive from them. The list is endless. */
ECHO "selected item : " nextname
END
```

1.27 Command List

MBench Commands

The following commands are available either via MBench's ARexx port, as an internal command or both. The available commands are:

General MBench

About

Help

LastMessage

Quit

RedrawAll

SetBackdropMode

UpdateAll

Window Commands

CleanUp

CloseWBWindow

OpenParent

SortList

SelectContents

ShowAllFiles

SnapshotWindow

SnapshotAll

ScrollWindowTop

ScrollWindowBottom

ScrollWindowPageUp

ScrollWindowPageDown

ScrollWindowUp

ScrollWindowDown

ScrollWindowFarLeft

ScrollWindowFarRight

ScrollWindowPageLeft

ScrollWindowPageRight

ScrollWindowLeft

ScrollWindowRight

SelectNextWindow

SelectPreviousWindow

Update

WindowsStacked

WindowToBack

WindowToFront
WindowsHorizontal
WindowsVertical
Icon Commands

Delete
Duplicate
EmptyTrash
Information
LeaveOut
NewDrawer
Open
PutAway
Rename
Snapshot
UnSnapshot
Disk Commands

DiskCopy
FormatDisk
Running Files

Run
WBRun
Other

ExecuteCommandWindow
FlushBuffers
OpenAssignWindow
OpenOpenLocationWindow
OpenLocation
 GetSelectedList

1.28 About

About

The About command is used to pop up MBenches 'About' requester.

Format:

About

Callable Via:

ARexx, Internally

Example:

About

1.29 Clean Up

CleanUp

The CleanUp command is used to arrange all of the icons in the specified window.

Format:

CleanUp [WINDOW]

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the window to arrange the icons in. If not specified this will arrange the icons in the current window.

Callable via:

ARexx, internally

Example:

CleanUp

CleanUp SYS:Prefs

1.30 Close Window

CloseWBWindow

The CloseWBWindow command is used to close the specified window.

Format:

CloseWBWindow [WINDOW]

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows to close. If not specified this will close the current window

Callable via:

ARexx, internally

Example:

CloseWBWindow
CloseWBWindow SYS:Prefs

1.31 Disk Copy

DiskCopy

The DiskCopy command creates a exact replica of a disk onto another disk.

Format:

DiskCopy [FROMDEVICE] [TODEVICE]

Template:

D=FROMDEVICE, T=TODEVICE

Specification:

FROMDEVICE=<Fromdevice> The Fromdevice command is the name of the device to copy from.

TODEVICE=<Todevice> The Todevice command is the name of the device to copy to. If this is not specified then it will use Fromdevice as a destination, prompting you when you need to switch disks.

Callable via:

ARexx, internally

Example

DiskCopy DF0:
DiskCopy DF0: DF1:

1.32 Duplicate

Duplicate

The Duplicate command will produce an exact copy of the selected item. It will give the new file a name of the form: eg. 'Daves' becomes 'Copy_of Daves'. If 'Copy_of Daves' already exists, then the file will be called 'Copy_2_of Daves' etc.

Format:
Duplicate

Callable via:
ARexx, internally

Example:
Duplicate

1.33 Delete

Delete

The Delete command will cause any selected icons to be deleted from existence. You will be prompted first!

Format:
Delete

Callable via:
ARexx, internally

Example:
Delete

1.34 Execute Command Window

The Execute Command Window

The Execute Command Window is where you can type commands to be directly executed by MBench.

The window is a standard line input window. You enter the name of the command you want to execute into the string gadget and then press either 'OK' to execute the command or cancel to close the window without doing anything.

Full keyboard shortcuts are available and when the string gadget is selected, pressing the up and down arrow keys will scroll through any commands you have previously typed.

1.35 Execute File Window

ExecuteFileWindow

The ExecuteFileWindow command is used to open an ExecuteFileWindow into which you can enter the name of a command to run.

Format:
ExecuteFileWindow [NAME]

Template:
NAME/F

Specification:
The command is required.

NAME=<Name> Where Name is the name to enter into the windows string
 gadget.

Callable via:
ARexx, internally

Example:
ExecuteFileWindow dir SYS:
ExecuteFileWindow SYS:Prefs/Font

1.36 Empty Trash

EmptyTrash

The EmptyTrash command will delete any files in Trashcans.

Format:
EmptyTrash

Callable via:
ARexx, internally

Example:
EmptyTrash

1.37 Flush Buffers

FlushBuffers

The FlushBuffers command is used to get MBench to flush any unused durectory buffers from memory and hence free up that memory.
other windows.

Format:
FlushBuffers

Callable via:
ARexx, internally

Example:
FlushBuffers

1.38 Format Disk

FormatDisk

The FormatDisk command will completely erase all the contents on the disk and will prepare it for the Amiga Operating System.

Format:

```
FormatDisk [DEVICE]
```

Template:

```
D=DEVICE
```

Specification:

DEVICE=<Device> The Device command is the name of the device to format.

Callable via:

ARexx, internally

Example

```
FormatDisk DF0:
```

1.39 GetSelectedList

GetSelectedList

The GetSelectedList command is used for obtaining a list of currently selected files and outputting that list to a file.

Format:

```
GetSelectedList [FILENAME]
```

Template:

```
F=FILENAME/F/A
```

Specification:

The filename is required.

FILENAME=<Filename> Where Filename is the name of the file to output the list to.

Callable via:

ARexx, internally

Example:

```
GetSelectedList RAM:List
```

1.40 Help

Help

The Help command is used to show help on a specific topic. It will cause MBench to bring up an amiga guide requester and display the page for the specified name or the default MBench page if there is no name.

Format:

Help [COMMAND]

Template:

COMMAND/F

Specification:

The command is optional.

COMMAND=<Command> Where Command is the topic that you want information on or NULL for the default "Main" page.

Callable Via:

ARexx, Internally

Example:

Help
Help Commands

1.41 Information

Information

The Information command will bring up a window showing information on the selected file.

Format:

Information [NAME]

Template:

NAME/M

Specification:

NAME=<Name> The Name command is used to specify a list of filenames to bring up a list of.

Callable via:

ARexx, internally

Example:

Information
Information DF0:
Information SYS:Prefs/Font

1.42 Last Message

LastMessage

The LastMessage command is used to get MBench to display the last message that it showed in its title bar (or the default one if there were no messages).

Format:

LastMessage

Callable Via:

ARexx, Internally

Example:

LastMessage

1.43 Leave Out

LeaveOut

The LeaveOut command will cause selected icons to be left out in the main MBench window.

Format:

LeaveOut

Callable via:

ARexx, internally

Example:

LeaveOut

1.44 Open Assign Window

OpenAssignWindow

The OpenAssignWindow brings up a window showing a list of all current assigns.

Format:

OpenAssignWindow

Callable via:

ARexx, internally

Example:

OpenAssignWindow

1.45 New Drawer

NewDrawer

The NewDrawer creates a new empty drawer to store files in along with an icon.

Format:

```
NewDrawer [NAME] [ADDEXTENSION]
```

Template:

```
N=NAME, A=ADDEXTENSION/S
```

Specification:

NAME=<Name> Where Name is the name to give to the newly created draw. If not specified then 'unnamed' will be used.

ADDEXTENSION=<Addextension> If this switch is specified then if the name already exists will start adding an extension to the file name
eg. 'unnamed' already exists so it will use 'unnamed1' instead.

Callable via:

```
ARexx, internally
```

Example:

```
NewDrawer SYS:Temp
NewDrawer SYS:Temp ADDEXTENSION
NewDrawer ADDEXTENSION
```

1.46 Open Open Location Window

OpenOpenLocationWindow

The OpenOpenLocationWindow brings up a window where you can enter the name of a directory to open a window for.

Format:

```
OpenOpenLocationWindow [NAME]
```

Template:

```
NAME/F
```

Specification:

NAME=<Name> Where Name is the default name to enter the string gadget or nothing for no name.

Callable via:
ARexx, internally

Example:
OpenOpenLocationWindow
OpenOpenLocationWindow SYS:Prefs

1.47 Open Location

Open Location

The OpenLocation brings up a window where you can enter the name of a directory to open a window for.

Format:
OpenLocation [MODE] [LOCATION]

Template:
MODE/K/N, LOCATION/F

Specification:

MODE=<Mode> Where Mode is the type of display mode to open the window in. Mode can be one of the following values:
0 - Autodetect, takes mode from icon.
1 - Icon mode.
2 - List mode.

LOCATION=<Location> Where Location is the directory to open the window for.

Callable via:
ARexx, internally

Example:
OpenLocation MODE=0 SYS:Prefs
OpenLocation MODE=2 C:
OpenLocation SYS:Prefs

1.48 Open

Open

The Open command will open the currently selected files. For drawers and disks this will open a window showing the contents. For programs this will run them.

Format:
Open

Callable via:

ARexx, internally

Example:

Open

1.49 Open Parent

OpenParent

The OpenParent command is used to open the window for the parent directory of the specified window.

Format:

OpenParent [WINDOW]

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows to use. If not specified this will use the current window

Callable via:

ARexx, internally

Example:

OpenParent

OpenParent SYS:Prefs

1.50 Put Away

PutAway

The PutAway command will cause selected icons left out in the main MBench window to be put back into their appropriate drawers.

Format:

PutAway

Callable via:

ARexx, internally

Example:

PutAway

1.51 Quit

Quit

The Quit command can be used to try and get MBench to quit. MBench will be unable to quit if there are running processes or MBench loaded programs.

Format:

Quit

Callable Via:

ARexx, Internally

Example:

Quit

1.52 Rename

Rename

The Rename command will bring up a window allowing you to change the specified file names to others of your choice.

Format:

Rename [NAMES]

Template:

NAMES/M

Specification:

NAMES=<Names> The Names command is used to pass a list of files to have renamed. If not specified then it will just use the currently selected files in MBench ←

Callable via:

ARexx, internally

Example

Rename DF0:

1.53 Redraw All

RedrawAll

The RedrawAll command is used to redraw all of MBenchs windows.

Format:

RedrawAll

Callable via:

ARexx, internally

Example:

RedrawAll

1.54 Run

Run

The Run command is used for executing programs as though started from the CLI. It allows you to specify arguments that should be executed as part of the command.

Format:

Run [COMMAND] [ARGLIST]

Template:

COMMAND/A,ARGLIST/F

Specification:

The command is required.

COMMAND=<Command> Where Command is the name of the program to run.

ARGLIST=<ArgList> Where ArgList are the arguments to pass to the program.

Callable via:

ARexx, internally

Example:

Run dir

Run list SYS:Prefs

1.55 Snapshot All

SnapshotAll

The SnapshotWindow command is used to save the current position and size of the current window aswell as the positioning of all of the icons.

Format:

SnapshotAll [WINDOW]

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the window to save its size and contents. If not specified this will save the current windows size and contents.

Callable via:
ARexx, internally

Example:
SnapshotAll
SnapshotAll SYS:Prefs

1.56 Show All Files

ShowAllFiles

The ShowAllFiles command is used to toggle between files that have icons only and showing all files.

Format:
ShowAllFiles [FALSE] [WINDOW]

Template:
FALSE/S,W=WINDOW/F

Specification:
FALSE If this switch is specified then it will show only icons else it will show all files.
WINDOW=<Window> Where Window is a file name specifying the name of the window to change the mode of. If not specified this will change the current windows.

Callable via:
ARexx, internally

Example:
ShowAllFiles
ShowAllFiles SYS:Prefs

1.57 Set Backdrop Mode

SetBackdropMode

The SetBackdropMode command is used to change the main window between backdrop and normal window mode in MBench.

Format:
SetBackdropMode [BACKDROP]

Template:
BACKDROP/S

Specification:
BACKDROP If specified sets the main window to backdrop mode else sets it to normal window mode.

Callable via:
ARexx, internally

Example:
SetBackdropMode BACKDROP
SetBackdropMode

1.58 Select Contents

SelectContents

The SelectContents command is used to select all of the files in the specified window.

Format:
SelectContents [WINDOW]

Template:
W=WINDOW/F

Specification:
WINDOW=<Window> Where Window is a file name specifying the name of the window to select the contents in. If not specified this will select the current windows files.

Callable via:
ARexx, internally

Example:
SelectContents
SelectContents SYS:Prefs

1.59 Snapshot

Snapshot

The Snapshot command will record the position of all selected icons.

Format:
Snapshot

Callable via:
ARexx, internally

Example:
Snapshot

1.60 Sort List

SortList

The SortList command is used to move the windows display up by one line.

Format:

```
SortList [WINDOW] [NAME] [SIZE] [DATE] [TIME] [TYPE]
```

Template:

```
W=WINDOW,N=NAME/S,S=SIZE/S,D=DATE/S,T=TIME/S,Y=TYPE/S
```

Specification:

WINDOW=<Window>	Where Window is a file name specifying the name of the window to sort the contents of. If not specified it will use the default window
NAME	If specified sorts the contents by name. This is the default.
SIZE	If specified sorts the contents by Size.
DATE	If specified sorts the contents by Date.
TIME	If specified sorts the contents by Time.
TYPE	If specified sorts the contents by Type.

Callable via:

```
ARexx, internally
```

Example:

```
SortList
SortList SYS:Prefs SIZE
```

1.61 Select Next Window

SelectNextWindow

The SelectNextWindow command will make the next MBench window the active window.

Format:

```
SelectNextWindow
```

Callable via:

```
ARexx, internally
```

Example:

```
SelectNextWindow
```

1.62 Select Previous Window

SelectPreviousWindow

The SelectPreviousWindow command will make the previous MBench window the active window.

Format:

```
SelectPreviousWindow
```

Callable via:

```
ARexx, internally
```

Example:

```
SelectPreviousWindow
```

1.63 Scroll Window Bottom

ScrollWindowBottom

The `ScrollWindowBottom` command is used to move the windows display down to the bottom of the display.

Format:

```
ScrollWindowBottom  
[WINDOW]
```

Template:

```
W=WINDOW/F
```

Specification:

`WINDOW=<Window>` Where `Window` is a file name specifying the name of the windows display to move to the bottom of. If not specified this will move the current windows down to the bottom of the display.

Callable via:

```
ARexx, internally
```

Example:

```
ScrollWindowBottom  
ScrollWindowBottom SYS:Prefs
```

1.64 Snapshot Window

SnapshotWindow

The `SnapshotWindow` command is used to save the current position and size of the current window.

Format:

```
SnapshotWindow [WINDOW]
```

Template:

```
W=WINDOW/F
```

Specification:

`WINDOW=<Window>` Where `Window` is a file name specifying the name of the

window to snapshot. If not specified this will snapshot the current window.

Callable via:

ARexx, internally

Example:

SnapshotWindow
SnapshotWindow SYS:Prefs

1.65 Scroll Window Down

ScrollWindowDown

The ScrollWindowDown command is used to move the windows display down by one line.

Format:

ScrollWindowDown
[WINDOW]

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows display to move down by one line. If not specified this will move the current windows display down by one line.

Callable via:

ARexx, internally

Example:

ScrollWindowDown
ScrollWindowDown SYS:Prefs

1.66 Scroll Window Far Left

ScrollWindowFarLeft

The ScrollWindowFarLeft command is used to move the windows display as far left as possible.

Format:

ScrollWindowFarLeft
[WINDOW]

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows display to move as far left as possible. If not specified this will move the current windows display as far left as possible.

Callable via:
 ARexx, internally

Example:
 ScrollWindowFarLeft
 ScrollWindowFarLeft SYS:Prefs

1.67 Scroll Window Far Right

ScrollWindowFarRight

The ScrollWindowFarLeft command is used to move the windows display as far right as possible.

Format:
 ScrollWindowFarLeft
 [WINDOW]

Template:
 W=WINDOW/F

Specification:
WINDOW=<Window> Where Window is a file name specifying the name of the windows display to move as far right as possible. If not specified this will move the current windows display as far right as possible.

Callable via:
 ARexx, internally

Example:
 ScrollWindowFarRight
 ScrollWindowFarRight SYS:Prefs

1.68 Scroll Window Left

ScrollWindowLeft

The ScrollWindowPageLeft command is used to move the windows display left by one unit.

Format:
 ScrollWindowLeft
 [WINDOW]

Template:
 W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows display to move left by one unit. If not specified this will move the current windows display left by one unit.

Callable via:

ARexx, internally

Example:

```
ScrollWindowLeft
ScrollWindowLeft SYS:Prefs
```

1.69 Scroll Window Page Up

ScrollWindowPageUp

The ScrollWindowPageUp command is used to move the windows display up one page.

Format:

```
ScrollWindowPageUp [WINDOW]
```

Template:

```
W=WINDOW/F
```

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows display to move up one page. If not specified this will move the current windows display up one page.

Callable via:

ARexx, internally

Example:

```
ScrollWindowPageUp
ScrollWindowPageUp SYS:Prefs
```

1.70 Scroll Window Page Down

ScrollWindowPageDown

The ScrollWindowPageDown command is used to move the windows display down by one page.

Format:

```
ScrollWindowPageDown
[WINDOW]
```

Template:

```
W=WINDOW/F
```

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows display to move down by one page. If not specified this will move the current windows display down by one page.

Callable via:

ARexx, internally

Example:

```
ScrollWindowPageDown
ScrollWindowPageDown SYS:Prefs
```

1.71 Scroll Window Page Left

ScrollWindowPageLeft

The ScrollWindowPageLeft command is used to move the windows display left by the visible width.

Format:

```
ScrollWindowPageLeft
[WINDOW]
```

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows display to move left by the visible width. If not specified this will move the current windows display left by the visible width.

Callable via:

ARexx, internally

Example:

```
ScrollWindowPageLeft
ScrollWindowPageLeft SYS:Prefs
```

1.72 Scroll Window Page Right

ScrollWindowPageRight

The ScrollWindowPageRight command is used to move the windows display right by the visible width.

Format:

```
ScrollWindowPageRight
[WINDOW]
```

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows display to move right by the visible width. If not specified this will move the current windows display right by the visible width.

Callable via:

ARexx, internally

Example:

```
ScrollWindowPageRight
ScrollWindowPageRight SYS:Prefs
```

1.73 Scroll Window Right

ScrollWindowRight

The ScrollWindowPageRight command is used to move the windows display right by one unit.

Format:

```
ScrollWindowRight
[WINDOW]
```

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows display to move Right by one unit. If not specified this will move the current windows display right by one unit.

Callable via:

ARexx, internally

Example:

```
ScrollWindowRight
ScrollWindowRight SYS:Prefs
```

1.74 Scroll Window Top

ScrollWindowTop

The ScrollWindowTop command is used to move to the top of the window.

Format:

```
ScrollWindowTop [WINDOW]
```

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the window to move to the top. If not specified this will move the current window to the top.

Callable via:

ARexx, internally

Example:

ScrollWindowTop

ScrollWindowTop SYS:Prefs

1.75 Scroll Window Up

ScrollWindowUp

The ScrollWindowUp command is used to move the windows display up by one line.

Format:

ScrollWindowUp [WINDOW]

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows display to move up by one line. If not specified this will move the current windows display up by one line.

Callable via:

ARexx, internally

Example:

ScrollWindowUp

ScrollWindowUp SYS:Prefs

1.76 Update All

UpdateAll

The UpdateAll command is used to update all of MBenchs windows to reflect any changes to files that have occurred but not been detected by MBench

Format:

UpdateAll

Callable via:

ARexx, internally

Example:

```
RedrawAll
```

1.77 Update

Update

The Update command is used to update the files in the specified window if MBench has not automatically done so.

Format:

```
Update [WINDOW]
```

Template:

```
W=WINDOW/F
```

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the windows to update. If not specified this will update the current window

Callable via:

```
ARexx, internally
```

Example:

```
Update  
Update SYS:Prefs
```

1.78 UnSnapshot

UnSnapshot

The UnSnapshot command will make the icon be placed by MBench rather than a specified co-ordinate.

Format:

```
UnSnapshot
```

Callable via:

```
ARexx, internally
```

Example:

```
UnSnapshot
```

1.79 WBRun

WBRun

The WBRun command is for running programs as though they were started by MBench.

Format:

```
WBRun [ADDARGS] [COMMAND] [ARGLIST]
```

Template:

```
ADDARGS/S,COMMAND/A,ARGLIST/F
```

Specification:

The command is required. ArgList is not currently implemented.

ADDARGS If specified adds all selected files in MBench as arguments passed to the program when it is run.
COMMAND=<Command> Where Command is the name of the program to run.
ARGLIST=<ArgList> A list of files to pass to the program as arguments.

Callable via:

ARexx, internally

Example:

```
WBRun SYS:Prefs  
WBRun ADDARGS SYS:Prefs
```

1.80 Windows Horizontal

WindowsHorizontal

The WindowsHorizontal command will cause all of MBenchs none backdrop windows to fill the screen in equal size horizontally so that none are overlapping, making it easier to see each window.

Format:

```
WindowsHorizontal
```

Callable via:

ARexx, internally

Example:

```
WindowsHorizontal
```

1.81 Windows Stacked

WindowsStacked

The WindowsStacked command will cause all MBench none backdrop windows to be placed into an orderly fashion where each window is placed under the title bar of another one.

Format:

WindowsStacked

Callable via:

ARexx, internally

Example:

WindowsStacked

1.82 Window To Back

WindowToBack

The WindowToBack command is used to move the selected window behind all other windows.

Format:

WindowToBack [WINDOW]

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the window to send to the back. If not specified, it will use the current window

Callable via:

ARexx, internally

Example:

WindowToBack

WindowToBack SYS:Prefs

1.83 Window To Front

WindowToFront

The WindowToFront command is used to move the selected window in front of all other windows.

Format:

WindowToFront [WINDOW]

Template:

W=WINDOW/F

Specification:

WINDOW=<Window> Where Window is a file name specifying the name of the window to send to the front. If not specified, it will

use the current window

Callable via:

ARexx, internally

Example:

WindowToFront

WindowToFront SYS:Prefs

1.84 Windows Vertical

WindowsVertical

The WindowsVertical command will cause all of MBenchs none backdrop windows to fill the screen in equal size vertically so that none are overlapping, making it easier to see each window.

Format:

WindowsVertical

Callable via:

ARexx, internally

Example:

WindowsVertical
