

GadMget

Jeremy Friesner

COLLABORATORS

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

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	GadMget	1
1.1	Contents	1
1.2	credits	1
1.3	How to reach me	1
1.4	disclaimer	2
1.5	distribution	2
1.6	requirements	3
1.7	GadMget Output format	3
1.8	introduction	4
1.9	Using GadMget	5
1.10	keys	5
1.11	fileloading	5
1.12	The GadMget GUI	6
1.13	Serving Suggestion	6
1.14	Starting GadMget	6
1.15	closeactionarg	8
1.16	loadbuffersarg	8
1.17	omitcommentarg	9
1.18	commentchararg	9
1.19	filepatarg	9
1.20	eolarg	9
1.21	aminetpatharg	10
1.22	keepfilesortarg	10
1.23	keepgetsortarg	10
1.24	filesortarg	10
1.25	getsortarg	10
1.26	keepblanklines	11
1.27	fontarg	11
1.28	fontsizearg	11
1.29	headarg	11

1.30	footarg	11
1.31	nocdarg	12
1.32	commandarg	12
1.33	noappwindowarg	12
1.34	appwindow	12
1.35	outputarg	13
1.36	The Lower Gadget - Selected Files	13
1.37	The Upper Gadget - Selectable Files	13
1.38	Disable Command-Splitting	13
1.39	Suppress mget printing	13
1.40	Case Sensitivity	14
1.41	Case Sensitivity	14
1.42	Case Sensitivity	14
1.43	Aminet Style Parsing	14
1.44	Grabbing .readme files	14
1.45	Grabbing archive files	14
1.46	Default Public Screen	14
1.47	Workbench Screen	15
1.48	Specify a Public Screen	15
1.49	Specify window height	15
1.50	Specify Window width	15
1.51	Specify window X-coordinate	15
1.52	Specify window Y-coordinate	16
1.53	Line length	16
1.54	FILE/A	16
1.55	GadMget menus	16
1.56	Sort Menu	17
1.57	Project Menu	18
1.58	Find Menu	19
1.59	Specify Line Length	20
1.60	Output Menu	20
1.61	outputnow	20
1.62	outputto	20
1.63	Use Command Splitting	21
1.64	Use Aminet-Style Parsing	21
1.65	Grab archive files	21
1.66	What the plus symbol means	22
1.67	Grab .readme files	22
1.68	Thanks	22

1.69	autoftp	22
1.70	autoftpuse	23
1.71	autoftpsetup	23
1.72	Long long ago..	24
1.73	Known Bugs and Other Problems	29
1.74	What's Next?	29

Chapter 1

GadMget

1.1 Contents

GadMget V2.05 by [Jeremy Friesner](#)

A nice little program to allow you to download more goodies from Aminet, faster than before!!!

Disclaimer Boring legal nothings

Distribution Who can have this program?

Requirements What do I need to run this program?

Introduction What does GadMget do?

Using GadMget How to run GadMget

AutoFTP What's this funny script?

Credits Where it's due

Acknowledgments Thanks to...

History Earlier versions/modifications

Future What next?

Known Problems Bugs! Aack!

1.2 credits

GadMget V2.05

Created by [Jeremy Friesner](#)

GUI originally created with GadToolsBox V2.0c

Compiled with DICE C

1.3 How to reach me

Here are some ways to get in touch with me:

by EMAIL: jfriesne@ucsd.edu

by SMail: Jeremy Friesner,
7581 Knollwood Place
Dublin, CA 94568

1.4 disclaimer

This software comes with no warranty, either expressed or implied.

The **author** is in no way responsible for any damage or loss that may occur due to direct or indirect usage of this software. Use this software entirely at your own risk.

1.5 distribution

GadMget is freeware. However, I have put a lot of time into it, so if you want to send **me** cash, flowers, Porsche 911's, etc... don't let me stop you! :)

Permission is given to include this program in a public archive (such as a BBS, FTP site or PD library) providing that all parts of the original distribution are kept intact. These are as follows:

Listing of archive 'GadMget2.05.lha':

Original Packed Ratio Date Time Name

```
-----
2273 562 75.2% 05-Aug-95 20:00:32 GadMget.info
10286 3185 69.0% 05-Aug-95 20:00:32 +AutoFTP.rexx
1922 933 51.4% 05-Aug-95 20:00:32 +AutoFTP.rexx.info
48840 24541 49.7% 05-Aug-95 20:00:30 +GadMget
59008 20812 64.7% 05-Aug-95 20:00:30 +GadMget.guide
504 286 43.2% 05-Aug-95 20:00:30 +GadMget.guide.info
2477 1198 51.6% 05-Aug-95 20:00:30 +GadMget.info
42760 42760 0.0% 05-Aug-95 20:00:40 +GadMGet.source.lha
554 334 39.7% 05-Aug-95 20:00:30 +GadMGet.source.lha.info
-----
```

168624 94611 43.8% 05-Aug-95 20:00:50 9 files

The contents of GadMget.source.lha are as follows:

Listing of archive 'GadMGet.source.lha':

Original Packed Ratio Date Time Name

```
-----
1261 400 68.2% 29-Jul-95 20:46:24 dmakefile
103469 28622 72.3% 05-Aug-95 19:54:04 gadmget.c
4106 1350 67.1% 29-Jul-95 15:56:32 GadMget.h
-----
```

```
2055 992 51.7% 12-Jun-95 14:40:16 mempools.h
21639 6187 71.4% 30-Jul-95 10:55:08 mget_listview2.c
3537 977 72.3% 28-Jul-95 18:29:32 mget_listview2.h
11462 3039 73.4% 25-Jul-95 00:38:28 tree.c
253 167 33.9% 14-Jul-95 21:55:38 tree.h
1618 721 55.4% 14-Jul-95 21:55:38 vixie.h
```

```
-----
149400 42455 71.5% 05-Aug-95 20:00:40 9 files
```

Operation successful.

No charge may be made for this program, other than a reasonable copying fee, and/or the price of the media.

1.6 requirements

GadMget requires an Amiga running Kickstart V37 (WorkBench 2.04) or higher to operate.

GadMget opens the following libraries:

Library Minimum Version #

```
-----
graphics.library 37
```

```
utility.library 37
```

```
gadtools.library 37
```

```
asl.library 37
```

```
intuition.library 37
```

```
workbench.library 37
```

```
icon.library 33 (37 to use iconify)
```

1.7 GadMget Output format

GadMget's output is meant to be pasted into the command line of your favorite ftp client.

The output can be of two types:

The default variety ("Simple path output") is the one acceptable by most ftp clients. It looks something like this:

```
----BEGIN SAMPLE-----
```

```
cd biz/dbase
```

```
mget dfa-21.lha
```

```
cd .././comm/term
```

```
mget term-4.0-DVI.lha term-4.0-030.lha term-4.0-Doc.lha
```



```
cd ../../dev/asm
mget BarflyDemo.lha memleak.lha
cd ../../text/tex
mget mma2ltx.lha
cd ../../util/app
mget AppMenu2_1.lha
-----END SAMPLE-----
```

If you are using ncftp, however, a more efficient form of output is available. To generate this type of output, specify the NOSIMPLEPATHS parameter on the command line or in the ToolTypes, or uncheck the "Use Simple Paths" menu item. This type of consists of nothing but "mget" commands, with the pathname relative to Aminet's root directory prepended to each.

```
-----BEGIN SAMPLE-----
mget util/misc/ReminderV11.lha docs/hyper/animeguide.lha util/cli/unxshell.lzh
game/shoot/XenoDemo.lha util/cli/PushD37_1.lha game/think/Dozer.lha
game/think/LittleGames.lha mus/edit/RGS-Util.lha mods/pro/Meaningless.lha
mods/pro/Cannon_Fodder.lha
mget mods/misc/Memorydust.lha pix/trace/helmet.lha
-----END SAMPLE-----
```

The output will go to the file specified in the **Output** parameter, or to stdout if there is no such parameter. This alone won't do you much good, you need a **way to get enter it into your FTP program** .

1.8 introduction

Please read the **History** section for information on changes and bug-fixes.

Being the incorrigible file-leeching weasel that I am, I like to download the RECENT file off of Aminet and browse through it, "mget"ing anything and everything that catches my eye. The only problem with this is that I have to type the name and directory of every file I want to download.

The extra thirty seconds a day this was costing me was simply intolerable, so I spent a few hundred hours writing GadMget instead. dB^)

GadMget will parse an Aminet-style file listing (such as the RECENT or INDEX files) and allow you to select which files you want by clicking on them in a ListView gadget. After you are done, it will **output** the proper mget line(s) needed to download your chosen files.

1.9 Using GadMget

The GUI Where do I click?

Startup options Command line arguments & ToolTypes

File loading Ways of selecting a file to browse

Menu options Explanation of cryptic menus

Keyboard Shortcuts For people too lazy to use the mouse

Serving Suggestions Hints

1.10 keys

There are many keyboard equivalents available in GadMget; those that correspond to menu items are listed in the menus. Other keyboard commands are as follows:

Up-Arrow : moves one line up the Available Files List

Down-Arrow : moves one line down the Available Files List

Left-Arrow : moves one line up the Selected Files List

Right-Arrow : moves one line down the Selected Files List

You can modify these keystrokes with the following command keys:

Shift : move one ListView's height up or down the list

Control : move ten items up or down the list

Alt : move to the top or the bottom of the list

Also, the backspace key is equivalent to Shift-Up-Arrow, and the Spacebar is equivalent to Shift-Down-Arrow.

1.11 fileloading

There are four ways to load a file into GadMget.

The first is to specify the **FILE** on the command line or ToolType.

The second is to extend-select the file--that is, hold down the shift key, click on the text file's icon once, and then double-click the GadMget icon.

The third method is applicable only when GadMget is open on the **Workbench** screen. Here, GadMget is an AppWindow--you may drag the text file's icon into the GadMget window, and this will cause GadMget to try to load that file.

The final method is the obvious one--use the bleeding file requester. ;-)

1.12 The GadMget GUI

Once the program is started, you will be presented with a screen containing two ListView gadgets. The one on the top will contain the contents of the text file, and the one on bottom will be empty. Clicking on an item in the **top ListView** will "select" it, moving it to the bottom ListView. Clicking on an item in the **bottom ListView** will send it back. When you are done selecting, click the close box or select "Output & Quit", and GadMget will spit out the **output** .

In the titlebar of the window are several figures about the selected files. They should be self-explanatory.

1.13 Serving Suggestion

Of course, having GadMget put text to your command line isn't very helpful, what you want is to have the **output** go into your ftp program. One way to work it is to use Jukka Marin's CLIP: device (available on Aminet as util/misc/cliphandler.lha) to store the **output** .

To do this, type:

```
GadMget work:RECENT OUTPUT CLIP:
```

And then after you've quit GadMget, a simple right-amiga-V spits the mget commands into your ftp window/Terminus session/whatever.

If you regularly use ncftp to connect to one Aminet ftp site, then an easier method is available to you. Simply rx the included AutoFTP **script** that is included in this archive, and it will do all the work for you. Note that you will probably have to modify this script a bit to reflect your directory setup and ftp site of choice, but it's nothing too difficult if you have a little skill with AmigaDos.

1.14 Starting GadMget

GadMget can be started from either the Shell or the WorkBench. It supports quite a few command line arguments. Each of these arguments also has a ToolType equivalent. Look in the GadMget icon for examples of these ToolTypes--they are there, just commented out with parentheses.

```
Template: GadMget FILE/A, OUTPUT/K, LINELENGTH/K, TOP/K, LEFT/K, WIDTH/K,  
HEIGHT/K, PUBSCREEN/K, WBScreen/S, DEFPUBSCREEN/S,  
NOFILES/S, GETREADMES/S, NOPARSE/S, CASESENSITIVE/S,
```

SORTCASESENSITIVE/S, SEARCHCASESENSITIVE/S, NOCOMMAND/S,
NOSPLIT/S, NOAPPWINDOW/S, COMMAND/K, NOSIMPLEPATHS/S,
NOCD/S, HEADER/K, FOOTER/K, FONT/K, FONTSIZE/K,
KEEPBLANKLINES/S, FILESORT/K, GETSORT/K, KEEPFILESORT/S,
KEEPGETSORT/S, AMINETPATH/K, EOL/K, COMMENTCHAR/K,
OMITCOMMENTS/S, LOADBUFFERS/K, CLOSEACTION/K

AMINETPATH/K Specify user-defined substitute for ../..

CASESENSITIVE/S Makes sorts AND searches case sensitive

CLOSEACTION/K Specify what to do when window is closed

COMMAND/K Change output's "mget" to something else

COMMENTCHAR/K Specify the character that denotes a comment

DEFPUBSCREEN/S Open on the Default Public Screen

EOL/K Specify string to use for End-Of-Line output

FILE/A File to browse

FILEPAT/K Specify default file pattern for ASL requesters

FILESORT/K Specify default File ListView sort

FONT/K Specify font to use

FONTSIZE/K Specify font size to use

FOOTER/S Textfile to postpend to output

GETREADMES/S Generate .readme files

GETSORT/K Specify default Get ListView sort

HEADER/S Textfile to prepend to output

HEIGHT/K Window height

KEEPBLANKLINES/S Display empty lines (day separators)

KEEPFILESORT/S Default to keeping File ListView sorted

KEEPGETSORT/S Default to keeping Get ListView sorted

LEFT/K Window x co-ordinate

LINELENGTH/K Maximum output line length

LOADBUFFERS/K Specify size of load buffer

NOAPPWINDOW/S Disable AppWindow

NOCD/S Don't output CD commands

NOCOMMAND/S Don't generate "mget "

NOFILES/S Don't generate archive names

NOPARSE/S Don't parse output, just print raw data

NOSIMPLEPATHS/S Don't use pathless output style

NOSPLIT/S Don't generate multiple mget commands

OMITCOMMENTS/S Don't load comment lines

OUTPUT/K Where to send the output

PUBSCREEN/K Open on a Public Screen

SEARCHCASESENSITIVE/S Makes searches case sensitive

SORTCASESENSITIVE/S Makes sorts case sensitive

TOP/K Window y co-ordinate

WBSCREEN/S Open on the Workbench Screen

WIDTH/K Window width

1.15 closeactionarg

In the past, GadMget has always automatically spit out output for the selected lines, whenever the GadMget window was closed. Now, however, this behavior can be modified. If you wish GadMget to save the state of the current file, instead, you can specify it as follows:

Usage Example: GadMget CLOSEACTION save

Or if you prefer that GadMget do nothing when the window is closed:

Usage Example: GadMget CLOSEACTION quit

or, a synonym:

Usage Example: GadMget CLOSEACTION none

If you wish GadMget to both save _and_ output, try this:

Usage Example: GadMget CLOSEACTION saveoutput

And, for completeness (although it's the default anyway)

Usage Example: GadMget CLOSEACTION output

1.16 loadbuffersarg

By specifying this keyword you can control the number of kilobytes of memory used by the file loading buffering algorithm. The higher the number you specify is, the faster things will load, but the more memory will be used (temporarily) while the file is loading.

If GadMget cannot allocate enough memory for the buffer, it will try again using a buffer that is ten times smaller, until the allocation succeeds, or the buffer becomes less than 1k.

Here are some rough timings I made for loading ftp.netnet.net's INDEX file on my A3000/25's Maxtor:

Using FastCache Without FastCache

No buffers (GadMget v1.97) 20 sec. 25 sec.

1K buffer 17 sec. 21 sec.

30K buffer 14 sec. 15 sec.

300K buffer 13 sec. 13 sec.

As you can see, there is a point at which a bigger buffer does not buy you any more speed--I tried a 3000K (!) buffer, and got results that were the same or slower than the 300K results.

The default value is 30K, and you can lower it as low as 1K or raise it as high as you can afford memory for. (Make sure that you have memory to hold both the buffer and the list you're loading in at once, otherwise GadMget will abort the load!)

Usage Example: GadMget work:recent LOADBUFFERS 30

1.17 omitcommentarg

Specifying this switch will cause GadMget to suppress any lines beginning with the **Comment Character** .

Usage Example: GadMget work:recent OMITCOMMENTS

1.18 commentchararg

With this argument you can change the character that GadMget uses to recognize a comment. Any string that starts with the Comment character will not be sorted, and if the **OMITCOMMENTS** argument is specified, they will not be displayed either.

The default comment character is "|". The comment character must be the first character in a line to be recognized.

Usage Example: GadMget work:recent COMMENTCHAR %

1.19 filepatarg

With this argument you can specify a string to appear by default as the pattern matching string in the ASL file requesters.

Usage Example: GadMget work:recent FILEPAT REC#?.TXT

1.20 eolarg

This argument allows you to specify a different string besides "\r\n" (carriage return, newline) for GadMget to print at the end of each line of output. This can be any string less than 50 characters long, and \r will be interpreted as a carriage return, \n as a newline, and \ as a single backslash. Other characters and backslash combinations will be output verbatim.

Usage Example: GadMget work:recent EOL "\n"

1.21 aminetpatharg

This argument allows you to specify another string besides "../.." for GadMget to use when CD'ing to a new directory.

Usage Example: GadMget work:recent AMINETPATH "/pub/aminet/"

Will cause GadMget to use absolute directory names.

1.22 keepfilesortarg

This argument causes GadMget to default to keeping the **Files ListView** sorted.

NOTE: Specifying this function can make Selects and Deselects rather slow when working with long lists (e.g. INDEX). Use with care.

Usage Example: GadMget work:recent KEEPFILSORT

1.23 keepgetsortarg

This argument causes GadMget to default to keeping the **Get ListView** sorted.

NOTE: Specifying this function can make Selects and Deselects rather slow when working with long lists (e.g. INDEX). Use with care.

Usage Example: GadMget work:recent KEEPGETSORT

1.24 filesortarg

This argument allows you to specify one of the four sort methods to be used by default in the **Files (top) ListView** . The keyword for the filetype you prefer should be written after this keyword (or after the = if you are setting this option in ToolTypes)

The four sort type keywords are: Size, Description, FileName, and Directory.

Usage Example: GadMget work:recent FILESORT FileName

1.25 getsortarg

This argument allows you to specify one of the four sort methods to be used by default in the **Get (bottom) ListView** . The keyword for the filetype you prefer should be written after this keyword (or after the = if you are setting this option in ToolTypes)

The four sort type keywords are: Size, Description, Filename, and Directory.

Usage Example: GadMget work:recent GETSORT Size

1.26 keepblanklines

If specified, this switch will prevent GadMget from suppressing the display (and printout, if Aminet parse is disabled) of blank lines in the input file.

1.27 fontarg

This argument lets you specify your favorite font to use in GadMget. The argument is case insensitive, and the .font extension is optional. If this argument is not specified, or the font is not found, GadMget will use the System Default Font (if run from WorkBench) or Topaz 8 (if run from the CLI).

Note: Using a proportional font is going to make things look messy!

1.28 fontsizearg

This argument allows you to specify the size of the font you want to use. If a Font is specified and this argument isn't, a size of 12 will be used. If the font size specified does not exist, the font will be algorithmically scaled to the specified size (warning: ugly!).

1.29 headarg

This argument allows you to specify a textfile for GadMget to output just before it outputs the directory data.

Shell Usage Example: GadMget work:recent HEADER s:GadMget.header

Why is this useful? Well, I use it with the following file:

```
---(snip)---
```

```
cd pub/aminet
```

```
binary
```

```
hash
```

```
---(snip)---
```

to automatically do the (tedious) setup commands before I download.

1.30 footarg

This argument allows you to specify a textfile for GadMget to output just after it outputs the directory data.

Shell Usage Example: GadMget work:recent FOOTER s:GadMget.footer

This argument is implemented mostly just to be symmetric with the HEADER argument--although you might want to put in a "cd ../.." or a "quit" command in the text file.

1.31 nocdarg

This argument allows you to suppress the cd commands output by GadMget when **Simple Path Mode** is selected. It has no effect if Simple Path Mode is not selected.

Shell Usage Example: GadMget work:recent NOCD

1.32 commandarg

This argument allows you to change the command GadMget prepends to the filelist.

Note that GadMget does not automatically put a space between the command you specify and the filepath, you should do this yourself if you want. (see the example, below)

Shell Usage Example: GadMget work:recent COMMAND "sz "

This would cause GadMget to create output lines like this:

```
cd biz/dbase
sz dfa-21.lha
cd ../comm/term
sz term-4.0-DVI.lha term-4.0-030.lha
cd ../dev/asm
sz BarflyDemo.lha GEDindent.lha memleak.lha
cd ../docs/misc
sz C_Tutor.lha
```

1.33 noappwindowarg

Specifying this argument will turn off GadMget's ability to load new files via the **AppWindow** mechanism. This argument only takes effect if GadMget is running on the WorkBench screen.

1.34 appwindow

When run on the WorkBench screen, GadMget's window is an AppWindow. What this means is, you may load a new file into GadMget's **top ListView** gadget by dragging their icons into the window.

The effect is exactly the same as if you had selected "Open" from the Project Menu and selected that file.

1.35 outputarg

This argument tells GadMget where to send its output string. If this argument is not specified, and GadMget is started from the shell, the output will be sent to stdout. If the argument is not specified and GadMget is started from the Workbench, output will be sent to the file "RAM:GadMget.output".

Shell Usage Example: GadMget work:recent OUTPUT ram:MyTempFile.txt

1.36 The Lower Gadget - Selected Files

The lower of the two ListView gadgets displays a list of all of the files you have selected to output. If you click on a line of text in this gadget, it will be transported back up to the [Selectable Files](#) gadget and put at the end of the Selectable Files list.

1.37 The Upper Gadget - Selectable Files

The upper of the two ListView gadgets displays a list of all the files in the file listing you have selected to browse. Clicking on any of these files will put it in the [Selected Files](#) gadget near the bottom of the window, thus indicating that you want to output the name of that file in the mget line.

There are certain lines that GadMget will not allow you to transport to the lower ListView. Lines that GadMget does not think are archive listing lines will not move, and if [Get Files](#) is not selected, you will not be able to selected lines that do not contain a [plus](#) symbol after the file size.

1.38 Disable Command-Splitting

This argument deactivates the [Split Commands](#) gadget in the [Output menu](#) .

Usage Example: GadMget work:recent NOSPLIT

1.39 Suppress mget printing

This argument causes GadMget to never generate the command "mget" at the beginning of its [output](#) lines. With this keyword, you can generate just a pure list of files, should you need one.

Usage Example: GadMget work:recent NOCOMMAND

1.40 Case Sensitivity

Specifying this argument has the same effect as specifying both **SEARCHCASESENSITIVE** and **SORTCASESENSITIVE** . It is included for convenience.

Usage Example: GadMget work:recent CASESENSITIVE

1.41 Case Sensitivity

This argument activates the "Case Sensitive" menu item in the **Find menu** .

Usage Example: GadMget work:recent SEARCHCASESENSITIVE

1.42 Case Sensitivity

This argument activates the "Case Sensitive" menu item in the **Sort menu** .

Usage Example: GadMget work:recent SORTCASESENSITIVE

1.43 Aminet Style Parsing

This argument deactivates the **Aminet Parse** menu item in the **Output menu** .

Usage example: GadMget work:recent NOPARSE

1.44 Grabbing .readme files

This argument activates the **Get Readmes** menu item in the **Output menu** .

Usage Example: GadMget work:recent GETREADMES

1.45 Grabbing archive files

This argument deactivates the **Get Files** menu item in the **Output menu** .

Usage Example: GadMget work:recent NOFILES

1.46 Default Public Screen

This keyword causes GadMget to appear on the default Public Screen.

Note that this may cause GadMget to appear on the Workbench screen, if the Workbench screen is the default Public Screen.

Usage Example: GadMget work:recent DEFPUBSCREEN

1.47 Workbench Screen

This keyword causes GadMget to appear on the Workbench Screen.

Usage Example: GadMget work:recent WBSCREEN

1.48 Specify a Public Screen

This allows you to specify the name of a Public Screen upon which you want the GadMget window to appear. If a Public Screen with the given name does not exist, GadMget will fail with a "Couldn't lock public screen!" error.

Remember, with Public Screen names, alphabetical case counts!

Usage Example:

GadMget work:recent PUBSCREEN MyPubScreen24

1.49 Specify window height

This allows you to set the height of the GadMget window.

Usage Example:

GadMget work:recent HEIGHT 250

will make the GadMget window 250 pixels tall.

The height of the window cannot be less than 167 pixels.

1.50 Specify Window width

This allows you to set the width of the GadMget window.

Usage Example:

GadMget work:recent WIDTH 350

will make the GadMget window 350 pixels wide.

The width of the window cannot be less than 155 pixels.

1.51 Specify window X-coordinate

This allows you to set the horizontal position of the GadMget window.

Usage Example:

GadMget work:recent LEFT 60

will make the GadMget window appear 60 pixels from the left of the screen.

1.52 Specify window Y-coordinate

This allows you to set the vertical position of the GadMget window.

Usage Example:

```
GadMget work:recent TOP 60
```

will make the GadMget window appear 60 pixels from the top of the screen.

1.53 Line length

The Amiga shell and possibly other programs will truncate commands that are over a certain number of characters long. The Amiga shell's maximum command length size is 255 characters, for instance. With this in mind, GadMget will generate multiple commands, none of which are longer than this value. If you wish to change this value, you may do so via the command line keyword `LINELENGTH` or via the "Line Length..." item in the **Output Menu** . Setting this value to 1 will cause GadMget to never put more than one file on a single command line.

Shell Usage Example:

```
GadMget work:recent LINELENGTH 150
```

1.54 FILE/A

GadMget will look for this keyword in your command string, and try to load the file specified directly after it. If it cannot find this keyword, it will try use the first argument on the command line as its file to load.

(Even if the first word is a parameter as well!)

Shell Usage Example: `GadMget work:recent`

Shell Usage Example: `GadMget WBScreen WIDTH 50 FILE work:recent`

1.55 GadMget menus

GadMget has a few menus for you to play with...

Project - The regular stuff

Find - Searching for text in the list

Sort - Sort either of the ListViews

Output - Output format control

1.56 Sort Menu

This menu is disabled if Aminet Parse is disabled, since most operations will not make sense if GadMget cannot interpret the lists as Aminet lists. The first two submenus, "Files List" and "Get List" pertain to the Lists in the Upper and Lower ListViews, respectively. They both have the same form, which is:

By Directory - Sorts the files by their directory pathnames. This is a good way to keep your output sorted if you want to generate as few "cd" commands as possible in the output. (Note: if do not you have **Simple Paths** selected, sorting your output with this will not lessen the size of the output.

By Size - Sorts the files by file size, smallest to largest. This is a good order to select if you like to unpack/view archives while downloading in the background. This way you can examine lots of little archives first while downloading the big ones.

By FileName - Sorts the files alphabetically by filename.

By Description - Sorts the files alphabetically by description.

By Age - Sorts the files by age as indicated in the "weeks old" column. You must, of course, be using RECENT/INDEX files with this column in them for this to work properly.

Keep Sorted - If this is checked, GadMget will try to keep the appropriate list sorted to whichever sort format you last specified. If you never selected a "sort", GadMget will default to "sort by directory". Note that using this option can make Selecting and Deselecting items rather slow when working with large lists (e.g. INDEX)

The Sort Menu contains two other items as well:

Sort Algorithm - This SubMenu allows you to choose between the two available methods of sorting the Lists. The default, "Tree Sort", is very quick, but will need to allocate additional memory in order to perform the sort. The amount of memory the Tree Sort will use depends on how long the List being sorted is-- For the entire Aminet INDEX list, it needs about 324k.

This sort will take roughly the same amount of time regardless of how close to being sorted the list was to begin with.

The other method, "Bubble Sort", is very primitive--it simply goes through the list and switches every out-of-order pair of items it finds, until there it can find no more

out of order pairs. This requires no additional memory, but it can take a LONG LONG time if the list wasn't already close to being sorted. Use with care! (Or better yet, don't use it at all!)

Case Sensitive - This toggle allows you to indicate whether or not you want GadMget to take Upper/Lower case into account when sorting. For example, Case Sensitive is not selected, then "Amiga.lha" would be listed next to "amiga.lha".

1.57 Project Menu

Open - Brings up the lovely and talented ASL filerequester to allow you to select a new file to load in. Only the **Selectable Files** gadget will be affected.

Append - Same as Open, only it puts the new file at the end of the old one, without first clearing the old stuff.

Save - Allows you to save the current state of the program back to the file currently being edited. The lists will be re-written out to the input file in the currently displayed order, and a special "bookmark" line will be inserted so that next time, GadMget will know which lines you had selected.

Save As - Same as Save, but you are prompted for a file to save to.

Clear - This is a submenu with two items. The first will clear the Files ListView (the upper one), and the second will clear the Get ListView (the lower one).

About - Opens a leetle info-window about **me** and the program.

Iconify - Makes GadMget reduce itself to an icon on the Workbench. You can then double-click on this icon to resume using GadMget, or drag icons onto it to load them into GadMget. (Note: GadMget gets its AppIcon from the icon that goes with the GadMget executable. This means you can replace it with your own by replacing GadMGet's program icon, if you wish. It also means that it's a good idea to keep GadMget's icon with the executable.)

Output & Quit - Makes GadMget spit out its **output** and go away. Selecting this is equivalent to clicking the close gadget.

Save & Quit - Causes GadMget to save the current contents of the two ListViews out to the current input file, and then quit without generating any output. Useful for when you want to remember your selections, but don't want to download them right now.

Quit - Makes GadMget quit without taking any other action.

1.58 Find Menu

Find Forward - Opens a window to allow searches of the **Selectable Files** gadget for any text string. All searches initiated with this menu item will start from the first currently visible item in the ListView.

Find Backward - Opens a window to allow searches of the **Selectable Files** gadget for any text string. All searches initiated with this menu item will start from the first currently visible item in the ListView and proceed backwards towards the top of the List.

Find from Top - Opens a window to allow searches of the **Selectable Files** gadget for any text string. All searches initiated with this menu item will start from the top of the list.

Find from Bottom - Opens a window to allow searches of the **Selectable Files** gadget for any text string. All searches initiated with this menu item will start from the bottom of the list and go backwards.

Find Next - Use this after having found an item with Find. This will find the next line that contains the same text string.

Find Previous - Works like "Find Next", only searches backwards.

Find Current - Returns to the last item that was highlighted by any Find function. Good for when you "lose your place".

Select Multiple - This item will prompt you to enter a key string, and will then transfer every item in the **Selectable Files** list that contains that key string to the **Selected Files** list.

If you enter an empty string into the gadget, no action will be taken. Note that a string of spaces is NOT an empty string!

Multiple Select is a very powerful command! For example, to download every module (music) file on Aminet, you could open the INDEX to your ftp site and do a Multiple Select on "mods/". Or to download all 18,000+ files (about 2.6 Gigabytes!), you could do a select on "/". Be careful!

Deselect Multiple - This item will prompt you to enter a key string, and will then transfer every item in the **Selected Files** list that contains that key string to the **Selectable Files** list.

If you enter an empty string into the gadget, no action

will be taken. Note that a string of spaces is NOT an empty string!

Case Sensitive - Decides whether or not alphabetical case matters in a search or (de)select. (i.e. is "Amiga" the same as "amiga"?)

1.59 Specify Line Length

This allows you to change the maximum length of the **output** lines that GadMget will produce. The default is 255 characters, the maximum amount that the Amiga shell can handle on one line.

1.60 Output Menu

This menu controls the formatting of the text output that GadMget generates.

Aminet Parse - Toggles Aminet listing -> filepath parsing

Get Files - If set, GadMget will get the archives that are selected

Get Readmes - If set, GadMget will get the .readme files associated with the selected archives.

Split Commands - If set, GadMget will generate multiple "mget" lines.

Use Simple Paths - If set, GadMget will output Simple Path style commands

Line Length - Allows you to set the maximum output line length

Output Now - Outputs without quitting program

Output To... - Change output destination

1.61 outputnow

Selecting this causes GadMget to spit out its output immediately, without having to quit. This is useful if you want the same or similar lists output to several files.

1.62 outputto

This brings up a string requester, into which you can type a filepath/filename or device name that you want GadMget to output to. Note that this does not cause GadMget to actually output anything, it just changes the output destination for when it does output.

1.63 Use Command Splitting

The Amiga shell (and doubtless many other programs) can only handle line lengths up to a certain size (255 characters to be exact). Because of this, when there are more than a few files selected for output, GadMget must split its output into multiple "mget" commands.

If you wish to deactivate this command-splitting mechanism, then deselect "Split Commands". Note that this will deactivate the **Line Length** menu item, since it is meaningless unless you want to split commands.

1.64 Use Aminet-Style Parsing

Normally this program is to be used in conjunction with FTP and the Aminet FTP sites to generate mget command lines. To generate command lines properly, GadMget has to make assumptions such as where in each line of text the file name and path are going to be.

However, it may be that you want to use GadMget without Aminet in mind, perhaps from a script to allow the user to select a sublist of some other list.

If that is the case, you can turn Aminet Parse off and GadMget will simply echo the selected lines to the output file, verbatim.

Note that because most of the other options in the Output menu are related to Aminet-style parsing, de-selecting Aminet Parse will disable them.

1.65 Grab archive files

Usually you will want to download the archives that you select in GadMget, but sometimes (perhaps over a slow modem?) you might want to get just the .readme files. If this is the case, deselect Get Files to tell GadMget that you don't want the archives.

Deselecting Get Files will automatically select Get Readmes, since if neither was selected, you wouldn't be able to get anything.

Note that when Get Files is deselected, you will not be able to select any files that do not have a '+' symbol directly after the file size. The '+' means that there is an .readme to go with that archive, and so if there is no plus there is no .readme, and if there is no .readme, then you can't get it. So there ;)

Any lines without readmes that you have already selected will remain in the **Selected Files** gadget, but they will not be output.

1.66 What the plus symbol means

In an Aminet listing, for any line where there is a + after the file size, there is an accompanying .readme file to go along with the archive. GadMget will automatically get this .readme file for you if you have the [Get Readmes](#) option selected

1.67 Grab .readme files

Select this menu item if you want to get the [.readme files](#) that are included with many of the archives on Aminet.

Deselecting this menu item will cause [Get Files](#) to be automatically selected, since if neither was selected, you wouldn't be able to get anything.

1.68 Thanks

Thanks go to the following people:

Jan van den Baard for GadToolsBox, which was used initially to design GadMget's GUI.

Matt Dillon for DICE

Commodore for the nifty computer and O/S

Brian Mury for the original AutoFTP script

Preben Nielsen for his help with ListViews

Paul Vixie for the AVL Tree routines

Mike Gleason, et al for ncFTP, and AmiTCP

David P. Wiles for the suggestions and support

...my friends, and all the nice people on the Internet who helped me out with Amiga problems.

1.69 autoftp

AutoFTP.rexx is an ARExx script designed to completely automate the process of downloading files from Aminet via GadMget and ncFTP.

AutoFTP will make use of rexxreqtools.library if it is present, but does not require it.

In order to make use of AutoFTP, you need to have AmiTCP running, and ncFTP must be installed on your system.

[Setting up AutoFTP](#)

[Using AutoFTP](#)

1.70 autoftpuse

AutoFTP can be run via its icon or the CLI. (It is a standard ARexx script) My favorite way is to run it from a ToolManager dock ;).

AutoFTP takes one of three possible arguments:

YES : Cause AutoFTP to always retrieve the newest RECENT file before loading GadMget.

NO : Cause AutoFTP go directly to GadMget without downloading the RECENT file or presenting a requester to the user.

DEBUG : Cause AutoFTP to emit information on what commands it is executing.

You can only specify one of these arguments. If neither YES nor NO is specified, a requester will be shown to the user, asking whether or not they wish to download a new RECENT.

1.71 autoftpsetup

In order to successfully run AutoFTP, you will probably need to alter a few parameters that are embedded within the script. Use your text editor to change any values you need to within the parameters section at the beginning of the AutoFTP.rexx script.

The parameters come set up according to my preferences, which may not match yours. A brief rundown of the parameters:

AminetSite : set this to the host name of the Aminet FTP Site you wish to use.

AminetRootDir : set this to the directory path where the Aminet archive is based on the FTP server. The RECENT file should be in this directory on the ftp site.

LocalRecentDir : set this to the directory on YOUR system where you wish to keep the RECENT file.

GadMget : set this to the fully qualified filename of GadMget on your system. I.e., you should be able to run GadMget from anywhere by typing this line into your shell.

ncFTP : set to the fully qualified filename for ncFTP.

GadMgetArgs : If you have any **personal preferences** for GadMget, enter them on this line. Note that AutoFTP overrides certain options, such as AMINETPATH, FILE and OUTPUT, so specifying these will have no effect.

DownloadDir : Enter the name of the directory you want AutoFTP to place the downloaded files (other than RECENT) into.

PreserveSelected : If you set this to YES, AutoFTP will retain any selected filenames from the old RECENT file, putting them in the selected section of the new RECENT file.

DoPruning : If this is set to YES, AutoFTP will check to make sure that any item you had in the old RECENT file's selected filenames list is not duplicated in the new RECENT file's list of filenames. This option can take some time on slow Amigas, or if you had many filenames selected in the old RECENT file.

1.72 Long long ago..

("-" = new feature, "*" = bug fix)

2.05 : (Released 08/05/95)

* Fixed some problems with iconification. GadMGet sometimes got confused about where its icon was, and wouldn't iconify. Now it is smarter about locating itself, and if it can't find its icon it will iconify with a default tool icon.

* GadMget was likely to crash if it couldn't reopen its window after iconifying. Fixed.

- For files of over 10,000 lines, the "Save" menu option will operate like "Save As", bringing up an ASL requester to confirm the save. I got tired of "accidentally" saving INDEX (a 2 minute operation), when I meant to select multiple files.

- **AutoFTP** is now much smarter about how it saves new RECENT files.

- **CLOSEACTION SAVE** , **CLOSEACTION OUTPUT** , **CLOSEACTION SAVEOUTPUT** and **CLOSEACTION NONE** can be specified at once, causing GadMget to save and/or quit when the close box is clicked.

2.0 : (Released 07/31/95)

- Mucked around with the Project menu--Added the **Save** , Save As..., **Save & Quit** , and **Output & Quit** items. The "Quit" item now quits without taking any action, and "Output & Quit" does what "Quit" used to do.

- Changed many of the keyboard shortcuts around. Sorry, guys! :-)

- Changed the order of the sort types in the menus and with respect to the keyboard shortcuts--now they correspond to the column number of the data they sort. (FileName = 1, Size = 2, etc.)

* Fixed a minor misprint in the abort requester.

- Somewhat faster loading via buffer pairs algorithm.
- Added **LOADBUFFERS** option.
- Added **CLOSEACTION** option.
- * Reorganized code structure for file loading and saving.
- Added **Iconify** option to Project Menu, and reworked AppWindow code.
- Added a counter and an abort sequence for list clearing.
- Added compilation date to About box.

1.97 : (Released 07/22/95)

- * Fixed window's title bar during "Sort by Age" sorts
- * Implemented the "Sort by Age" tooltips (oops)
- * The comment lines are now kept at the top of the list during Tree Sorts.
- * Fixed several Enforcer hits in the search engine, and an evil bug wherein a search could cause a subsequent ClearList() to hose the system.
- Rewrote AutoFTP in ARexx, and documented it better.
- Added **COMMENTCHAR** parameter
- Added **OMITCOMMENTS** parameter
- Sorts, Loads, Appends, and Searches can all now be aborted by pressing <Escape> or clicking on the close box.
- Removed 'c' directory from GadMget distribution as it is no longer needed by AutoFTP.

1.95 : (Released 5/28/95)

- Added file pattern matching to the ASL file requesters, and a FILEPAT startup parameter to set its default.
- Tree Sorts of large files have been sped up dramatically. Apparently using malloc() and free() is much less efficient than using AllocMem() and FreeMem(), so now tree.c uses the latter. As an example, sorting netnet's INDEX by Directory used to take 3 minutes, 28 seconds on my A3000/25, now it takes 48 seconds.
- * "Sort by Age" now works correctly, and line parsing is improved in general.
- * Slowed down the progress counters a bit, don't want them to waste CPU. (No, this does NOT mean that things are working slower--use a stopwatch if you want to be sure ;))

1.9 : (Released 02/10/95)

- * Now (sort of) intelligently parses data lines to find data column positions instead of just assuming the data to be in a specific column. This should fix problems that occur whenever the Aminet guys decide to adjust the column spacings.

- Added "Sort by Age" function to the sort menu.
 - A space is no longer automatically added between a custom command string (i.e. user's "mget" replacement) and the filepath. This allows a bit more flexibility, since you can always add a space in yourself.
 - Added a progress indicator to the Tree Sort function.
- 1.8 : (Released 12/18/94)
- NewLookMenus flag for proper menu coloring under 3.x
 - * Multiple Select/Deselect now smarter/faster for large (de)select ops when Keep Sorted is selected. (it was keeping the list sorted at every line transfer, now for large lists it only sorts once, when it's done.)
 - * Better handling of out-of-memory situations during file loading
 - Added **AMINETPATH** keyword to allow changing of "../.." to something else
 - Added **EOL** keyword for end-of-line string specification
 - Added Clear SubMenu to Project menu, allowing you to clear either the files list or the get list.
 - * Handles carriage returns at the end of input lines better
 - * Killed the occasional Enforcer hit that happened during Case-Insensitive Multi[de]Selects.
 - * Window title now always properly reset when cancelling a daughter window.
- 1.7 : (Released 12/02/94)
- * The Search menu is now properly called the Find menu.
 - Added **Multiple Select** and **Multiple Deselect** options to the Find menu. You can now grab lots of files by key substring!
 - Separated **CASESENSITIVE** switch into **SORTCASESENSITIVE** and **SEARCHCASESENSITIVE** -- although the old switch is still there also, and is the same as specifying both of the new ones.
 - Added more keyboard shortcuts for menu items and switched some around around
 - Added Search from beginning/end, backward search, and Previous options to the Find menu.
 - Added {"keyboard shortcuts" link Keys} for moving through the Lists
 - * Fixed the "Resizing window causes move to top of list" bug-- thanks to Preben Nielsen for help with this one.
 - Sorts by filename now are case sensitive or not depending on Case Sensitive flag in Find menu (should this be changed to a separate Case Sensitive flag?)
 - GadMget now defaults to a new sort algorithm based on AVL Trees. This means that sorts actually take an acceptable period of time now! For instance, on my A3000/25, sorting warchive's INDEX now takes about 35
-

seconds.

- * Fixed a hidden bug that caused the incorrect sort type to be used when "Aminet Parse" was reselected with "Keep Sorted" checked.
- Added an option to switch between the AVL Tree Sort and (old) Bubble Sort algorithms, since Tree sorting does take some extra memory and there might be people who can't spare the RAM.

1.6 : (Released 10/29/94)

- * Fixed bug that caused text not to be displayed when you didn't specify a font to use under AmigaDos 3.0
- Added the **Output To...** option to the Output menu.
- Added the **Output Now** option to the Output menu.
- AutoFTP is now runnable from Workbench via Iconx.
- AutoFTP no longer requires the s:GadMget.header file.
- Added the **Sort menu** , with 4 different types of sorts, and the Keep Sorted option.
- Added **KEEPFILESORT** and **KEEPGETSORT** switches
- Added **FILESORT** and **GETSORT** keywords

1.5 : (Released 10/01/94)

- * Fixed bug that would cause nasty crashes if you used a larger font on the WorkBench screen and then sized GadMget too small.
 - * GadMget now allows you to select files with unknown ('?') file size.
 - GadMget now includes a short AmigaDos script that can be used in conjunction with ncftp to completely automate ftp sessions.
 - GadMget is now (more or less) font sensitive.
 - Added **FONT** and **FONTSIZE** startup parameters
 - GadMget distribution now also contains EasyReq, a tiny little requester-displaying program written by me and used by the AutoFTP script.
 - * Window no longer has to close and reopen when it is resized (thanks to Petter Nilsen for the help on this)
 - Added **KEEPBLANKLINES** switch
- 1.4 : (Released 08/04/94)
- * Embarrassingly enough, GadMget's output format would apparently only work on a few ftp clients such as ncFtp. GadMget now also supports a simpler output format that works on all ftp clients.
 - Added **NOSIMPLEPATHS** switch
 - Added **COMMAND** keyword
 - Added **N OCD** switch
 - * The **NOMGET** switch is now called **NOCOMMAND** (since the generated
-

command is no longer necessarily "mget")

- Added HEADER and FOOTER keywords

1.3 : (Released 07/06/94)

- GadMget now can be started from the WorkBench, with ToolTypes!

* Removed a glitch which caused GadMget to highlight the wrong line if you did a word search on very short files.

- Added FILE and OUTPUT parameters

- GadMget is now an AppWindow if run on the workbench (unless you specify NOAPPWINDOW --another added parameter!)

- Most errors now pop up EasyRequesters instead of printing to the standard output.

- GadMget now defaults to opening a screen with the same dimensions as the Workbench screen.

* Removed INTERLACE and OVERSCAN flags since they no longer affect anything (see previous item)

- Funky fresh icon

1.2 : (Released 06/25/94)

- GadMget's window is now resizable

- Added HEIGHT , WIDTH , TOP , LEFT parameters

- GadMget can now open on the Workbench or on a Public Screen

- Added string search window, CASESENSITIVE parameter

- Added Output menu to control formatting from within the program

- Added File Requester

- Added Busy Pointer

* Hopefully finally squashed that bug that was causing the occasional garbage output

- AmigaGuide docs!

- Code is once again a gigantic mess

1.1 : (Released 06/16/94)

- Added NOLACE, NOOVERSCAN, LINELENGTH , GETREADMES , and NOFILES parameters.

* Patched up the source a bit, and spruced up the docs.

1.0 : (Released 06/08/94)

- First release. Simple and kind of buggy.

1.73 Known Bugs and Other Problems

Here are the things that still don't work quite right with GadMget V1.5:

- There are still occasional Enforcer hits--unfortunately, they are occasional enough to be really hard to pin down!
- The NOCD command is probably useless, and doesn't work quite right anyway.
- Files with tabs and such in them will come out looking rather strange. (this isn't a problem if you stick to file listings only, of course)
- After reading very large files such as INDEX, GadMget sometimes leaves memory rather fragmented. This is because each line in the file is AllocMem'd separately.
- Opening GadMget on the Workbench screen when you don't have Workbench loaded causes a 40 byte memory leak.
- Using KEEPBLANKLINES together with NOPARSE will often cause GadMget to output some extra blank lines at the end of the output stream.
- Probably there are some glitches when using variations of EOL together with AminetParse = FALSE, headers, footers?
- If there are different line formats in the listings, sorts will sometimes give incorrect results.
- GadMget doesn't parse CLI startup args of the form PARAM=value. It should.

1.74 What's Next?

- find a fix for the AllocMem/fragmentation problem.
 - other bug fixes as they show up (and they will, oh yes, they will... ;-)
 - Smaller & cleaner code
 - Add an Iconify gadget.
 - Any suggestions?
-