GadMget

Jeremy Friesner

COLLABORATORS			
	TITLE : GadMget		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Jeremy Friesner	February 12, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

GadMget

1.1   Contents

GadMget V1.95 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

Credits Where it's due

Acknowledgments Thanks to...

History Earlier versions/modifications

Future What next?

Known Problems Bugs! Aack!

1.2   credits

GadMget V1.95

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 'GadMget1.95.lha':

Original Packed Ratio Date Time Name

-------- ------- ----- --------- -------- -------------

2273 559 75.4% 28-May-95 00:00:48 GadMget.info

2100 1032 50.8% 28-May-95 00:00:48 +AutoFTP.script

511 284 44.4% 28-May-95 00:00:48 +AutoFTP.script.info

2228 508 77.1% 28-May-95 00:00:46 +c.info

6804 4437 34.7% 28-May-95 00:00:46 +EasyReq

391 225 42.4% 28-May-95 00:00:46 +README

8332 5277 36.6% 28-May-95 00:00:46 +UniqueFile

44440 21903 50.7% 28-May-95 00:00:46 +GadMget

47423 16745 64.6% 28-May-95 00:00:48 +GadMget.guide

504 289 42.6% 28-May-95 00:00:48 +GadMget.guide.info

2388 1152 51.7% 28-May-95 00:00:46 +GadMget.info

36914 36914 0.0% 28-May-95 00:00:56 +GadMGet.source.lha

554 337 39.1% 28-May-95 00:00:48 +GadMGet.source.lha.info

-------- ------- ----- --------- --------

154862 89662 42.1% 28-May-95 00:01:04 13 files

The contents of GadMget1.95.source.lha are as follows:

Listing of archive 'GadMGet.source.lha':

Original Packed Ratio Date Time Name

-------- ------- ----- --------- -------- -------------

1141 362 68.2% 27-May-95 23:56:38 dmakefile

92056 24574 73.3% 27-May-95 23:51:56 gadmget.c

3527 1127 68.0% 18-May-95 11:54:34 GadMget.h

20521 5703 72.2% 02-Feb-95 00:34:16 mget_listview2.c

3405 950 72.0% 01-Feb-95 15:29:40 mget_listview2.h

11462 3039 73.4% 27-May-95 23:08:16 tree.c

253 167 33.9% 17-Dec-94 22:02:46 tree.h

1618 721 55.4% 17-Dec-94 22:02:46 vixie.h

-------- ------- ----- --------- --------

133983 36643 72.6% 28-May-95 00:00:56 8 files

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

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 in by hand.
The extra thirty seconds a day this was costing me was simply intolerable,
so I spent a few hundred hours writing GadMget instead. dBˆ)
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 to lazy to use the mouse

Serving Suggestion 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, for Backspace 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 bleedin' 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 "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 execute the 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

FILE/A File to browse

OUTPUT/K Where to send the output

LINELENGTH/K Maximum output line length

TOP/K Window y co-ordinate

LEFT/K Window x co-ordinate

WIDTH/K Window width

HEIGHT/K Window height

PUBSCREEN/K Open on a Public Screen

WBSCREEN/S Open on the Workbench Screen

DEFPUBSCREEN/S Open on the Default Public Screen

NOFILES/S Don't generate archive names

GETREADMES/S Generate .readme files

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

CASESENSITIVE/S Makes sorts AND searches case sensitive

SORTCASESENSITIVE/S Makes sorts case sensitive

SEARCHCASESENSITIVE/S Makes searches case sensitive

NOCOMMAND/S Don't generate "mget "

NOSPLIT/S Don't generate multiple mget commands

NOAPPWINDOW/S Disable AppWindow

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

NOSIMPLEPATHS/S Don't use pathless output style

NOCD/S Don't output CD commands

HEADER/S Textfile to prepend to output

FOOTER/S Textfile to postpend to output

FONT/K Specify font to use

FONTSIZE/K Specify font size to use

KEEPBLANKLINES/S Display empty lines (day separators)

FILESORT/K Specify default File ListView sort

GETSORT/K Specify default Get ListView sort

KEEPFILESORT/S Default to keeping File ListView sorted

KEEPGETSORT/S Default to keeping Get ListView sorted

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

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

FILEPAT/K Specify default file pattern for ASL requesters

1.15 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 FILEPAT REC#?.TXT

1.16 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"

1.17   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.

1.18   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 KEEPFILESORT

1.19   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

1.20   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

1.21   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

1.22 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.

1.23 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!

1.24 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!).

1.25 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.

1.26 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.

1.27   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

1.28   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

1.29   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.

1.30   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.

1.31   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 send to the file
"RAM:GadMget.output".
Shell Usage Example: GadMGet work:recent OUTPUT ram:MyTempFile.txt

1.32   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.

1.33   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 selecte lines that do not contain a plus symbol
after the file size.

1.34   Disable Command-Splitting

This argument deactivates the Split Commands gadget
in the Output menu .
Usage Example: GadMget work:recent NOSPLIT

1.35   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

1.36 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

1.37 Case Sensitivity

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

Usage Example: GadMget work:recent SEARCHCASESENSITIVE

1.38 Case Sensitivity

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

Usage Example: GadMget work:recent SORTCASESENSITIVE

1.39 Aminet Style Parsing

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

Usage example: GadMget work:recent NOPARSE

1.40 Grabbing .readme files

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

Usage Example: GadMget work:recent GETREADMES

1.41 Grabbing archive files

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

Usage Example: GadMget work:recent NOFILES

1.42 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

1.43   Workbench Screen

This keyword causes GadMget to appear on the Workbench Screen.

Usage Example: GadMget work:recent WBSCREEN

1.44   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

1.45   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.

1.46   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.

1.47   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.

1.48   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.

1.49   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

1.50   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

1.51   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

1.52   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 rougly 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".

1.53   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.

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.

Quit - Makes GadMget spit out its output and go away.

1.54   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!

Note that 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 13,000+

files (about 2 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"?)

1.55   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.

1.56   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

1.57   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.

1.58   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.

1.59   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.

1.60   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 sucha 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.

1.61   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.

1.62   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

1.63   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.

1.64   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 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.

1.65   autoftp

The supplied AutoFTP script was emailed to me by Brian Mury.

Thanks, Brian!

1.66   Long long ago..

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

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 format.

- 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 Search 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 Search 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 wcarchive'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 NOCD 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 NOFILES and GETREADMES parameters

- 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.

1.67   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. I will try to fix this in a future release.

- 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 sometimes tries to sort the header lines.

1.68   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

- Any suggestions?