

in

COLLABORATORS

	<i>TITLE :</i> in		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 18, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	in	1
1.1	legal	1
1.2	Argue	1
1.3	Argue	2
1.4	special	3
1.5	popup	3
1.6	history	4
1.7	preset	4
1.8	07	5
1.9	probs	6
1.10	Magic User Interface	6
1.11	window	6
1.12	dev	7
1.13	req	7
1.14	future	7
1.15	Author	8
1.16	Remember this is EMAILWARE	8
1.17	-Introduction-	9
1.18	-Usage-	9
1.19	older	10
1.20	newer	10
1.21	nospace	11
1.22	oldest	11
1.23	Continue...	12
1.24	New features	13
1.25	faq	14

Chapter 1

in

1.1 legal

Argue (c) Thorsten Stocksmeier (flavour@aventure.teuto.de). This software, including all submitted parts I made, are given out "as is". The author does not give you any guarantee and isn't responsible for any damage caused by the submitted tool. (pew. enough of that. :-)

1.2 Argue

Here's the way to Argue's documentation!

ARGUE IS SUBMITTED UNDER THE RULES OF EMAILWARE. THIS IS NOT A JOKE! IF YOU USE THIS TOOL AND HAVE AN EMAIL ACCOUNT, THEN YOU *MUST* SEND ME AN EMAIL. THIS EMAIL WILL ACT LIKE A SHAREWARE REGISTRATION!!!

I DO NOT WANT GIFTS! I DO NOT WANT MONEY OR POSTCARDS!

I JUST WANT AN

```

|_____|_/_|_____| | | | | | |
| |_____| \ / |_____|
| ____| | \ / | / _' | | |
| |_____| | | | ( | | | |
|_____|_| | | \_,_|_|_| feedback!!!

```

WRITE TO: f l a v o u r @ a v e n t u r e . t e u t o . d e

1.3 Argue

Argue v0.9 (18.7.96)

by Flavour

Remember this is EMAILWARE

* -> new since 0.9

Legal
some legal information

Introduction
introducing Argue! ;-)

Requirements
what you need...

Usage
How to use

The GUI window
Argue in action!

*
Problems
Some minor problems :)

*
Future
Let's look into the future.

*
Special gadgets
ReadArgs() extensions!

Preset gadgets
Brand new features!

*
New popups

*
Add. tooltypes

History
Former times...

*
Argue FAQ
Frequently Asked Questions

For developers
READ THIS!

Author
Who wrote it?

Argue's interface is made with
MUI
(c) Stefan Stuntz!

BTW: If you like to write a german guide about all that (I'm just too lazy :), ←
feel
free to ask me. I would be very happy about that ;) (Jo mei, des waer a Freud ←
)

1.4 special

In Argue 0.9 there is the first built in extension. The cycle gadget handles two opposing switches (HEAD/S,NOHEAD/S) that can be put together this way:

```
...,Description/C(first,second),...
```

"first" and "second" represent the raw switch titles.

Note:

- the /C cycle switch does not work in Unix mode!
- do not use ReadArgs() things in /C brackets. blah/C(tic/N,toc/K) will *NOT* work!
- Cycle gadgets are ALWAYS active. They can't be switched off. So place the option you prefer as "first", because the first title will be set as active at initialization time.

1.5 popup

Argue 0.9 has several new popups. Argue itself decides about which one it will use by looking for argument fragments/keywords that occur.

Fragment	Popup
FONT	Add an ASL font requester popup. Don't care about the font size in the string gadget, this will be cut off when "GO!" is pressed.
SCREENM	Now at all it works ;-)) Thanks to Matthias Scheler. This screenmode popup returns the name of the selected screenmode ← for picture viewers and so on-

PUBSC	Argue builds up a list of public screens at initialization time. Choose out one here.
PASSW	A gadget without popup. Any character is represented by a dot.
FILE	The usual file popup. (default)
DEVICE	A very useful device list will pop up. This is a real nice thing ;-)
DRIVE	Nice drive list popup, for backup tools etc.

1.6 history

... to be continued ...

- 0.9 - ASL multiselect for the /M multiple gadget causing two new internal hooks :), screenmode popup works now, failed New()s and String()s raise an IMEM (insufficient memory) exception. (for safety), added the new cycle switch, output string size now 10kb., new drive list popup, help file support added for the example scripts, string gadgets advance on carriage return (MUIA_String_AdvanceOnCR)
- 0.8 - major improvements: pre-settable switches and string gadgets, windowID no more "MAIN" but "ARG!", five new popup buttons for several arguments containing keywords. (PubScreen etc.)
- 0.7 - template parsing has been widely extended to allow min/max/actual extensions for integer arguments. fixed this and that bug. (as always :)
- 0.6 - new release version, added multi argument list, help bubbles, unix/nospaces routines rewritten, lets the user decide whether he likes the arguments in a register group or not
- 0.5 - all argue development was switched to MUI, which is really predestinated for those dynamic gadget things ;)
- 0.4 - interim release of 0.3 :)))
- 0.3 - first release version using the nicegui system. not a very good gui layout system, but it did the job
- 0.2 - again some alpha things. they DID work, but how...
- 0.1 - never released alpha version (didn't work ;-)

1.7 preset

(If you like, read the older page about preset numeric gadgets here !)

Since Argue 0.7 I built in predefined gadgets. Have a look at the example interfaces for that. This can save huge amounts of time!

How to preset a gadget?

Integers

Replace Number/N by Number/N(2,0,4) and the new numeric gadget will range from null to four and will have the "two" activated after initialization.

Syntax: argument(actual,minimal,maximal)

All those values may go upto 99999! It is also possible to work with negative values. There is no checking for this! MUI DOES accept them, so I would say you may go downto -99999 for integers and -9999 for knobs

WARNING: Please care about ALL the three values. Argue wants to have them, and the minmax-parser is not 100% guru safe when fed with illegal things.

Strings

Replace String/K by String/K(Yeah!) and the new string gadget will show the string "Yeah!" after initialization.

This works with all kinds of string gadgets except /M multiple lists.

Checkmarks

Replace Option/S with Option/S(on) and the new checkbox gadget will be pre-clicked :-)

1.8 07

Bugfix: Argue 0.7 will add the keyword to ANY integer argument. This was not a real bug, but ReadArgs() might get confused with it.

Since Argue 0.6 I added a useful new feature. Any number argument (/N) can have a minimum, maximum and standard value.

This feature is added to the template. Let's say we have a "width" field for a screen definition in our template.

Usually we would have

WIDTH/N

This thing would be at value NULL after Argue is started and will go upto 1000.

Now let's add the new feature to it. Minimum for width will be 320, maximum

1280, and the default value (which is shown without the user's interaction) will be 640.

Simple write

```
WIDTH/N(640,320,1280)
```

Note the order of the values. They must all be supplied, else Argue might fail to open the interface.

1.9 probs

- Argue is not 100% guru safe when fed with shit. (too long arguments, binaries as "templates" or whatever you can think of. The thing YOU as the user have to do is to be a bit careful. That's all.
- preset-parser is not 100% safe
- cycle gadget parser does not support unix mode, and I'm quite sure it will never do ;)

1.10 Magic User Interface

(I don't know if this text is the newest one available)

```
MUI - MagicUserInterface  
=====
```

(C) Copyright 1993/94 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called muiXXusr.lha (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send DM 30.- or US\$ 20.- to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

1.11 window

Since Argue 0.3 there were extreme changes in the interface.

The interface is now built with
MUI
, allowing unbelievable
features.

You'll see the arguments in three types of gadgets. Switches can be clicked to activate/deactivate. Integers can be pushed around with the mouse.

String gadgets may be filled by hand or with the filerequester pop up button.

Key cycling offers a great thing. You don't need your mouse any more to fill out the interface. Just press the TAB key.

There are two additional buttons:

About...	Tell something about Argue
Go!	Start the action.

1.12 dev

If you are a developer and like to write shell-tools (like me :-)) then I would be very happy if you also supply an Argue-Icon for your tool.

In this case, you should add an Argue-drawer to your distribution containing the binary and your icon.

It would be also nice to leave me a note if you do so.

Try to keep Argue alive ;-)))

1.13 req

Argue requirements

- OS 3.x or higher
- some bit of memory
- a bit of adventure ;-)
- MUI (I think it needs 3.3)

1.14 future

Again some new Argue release is out to the public, and I think it is quiet a neat tool ;-))

Argue seems also to be the longest following source code I ever wrote. Actually source size is ~46kb.

Perhaps a bit of shortening would help or putting all those procedures into modules. (BTW: Argue is written in pure AmigaE - Hi Wouter :))

But those things are internal cleanups, you'll await some visible changes.

Hmmmm... There's not much left. Send me your ideas via EMail! We'll certainly find a way of realizing them ;)

Did I already tell you an Argue homepage is in work? You may get the newest version of Argue there when it's finished ;)

But I'll probably tell more about that in Argue 1.0 (hey, isn't that a nice number??? ;-))

1.15 Author

Argue

(c)

Thorsten "Flavour" Stocksmeier

Contacts, Comments, Flames:

flavour@aventure.teuto.de

THIS IS EMAILWARE!

1.16 Remember this is EMAILWARE

The submitted software "Argue" is supplied under the rules of ↔
FREEMWARE.

You don't have to pay a lick for this. If you did, contact me immediately.

You are not allowed to alter the files in this archive. You are (of course) allowed to have a look at everything, use it and spread it :-)

Putting this on a disk or CD is absolutely wanted, if no part of the files is deleted. It would be nice of those magazines to send a free issue to me. :)

Besides Public Domain, this software is EMAILWARE.

If you like/hate/adore/desire/delete :) this tool you have to send an EMail to the author if you can.

See the EMail-address
here

1.17 -Introduction-

You have a shell tool with thousands of arguments? You hate configuring a program with tooltypes or dropping another icon on it?

Here is what you wanted.

Perhaps we'll start with a little background information.

Since OS2.0 there is a standardized way of reading shell arguments. This system function is named ReadArgs(). It parses the arguments with a pattern called the TEMPLATE.

A template looks like this: FILE/A,QUICK/S,BOOM/N

This example parses a filename, a switch (quick) and a number (1234).

Now there's quite a lot of people that type extreeeeemly long argument lines to call a program.

Especially for unix-fashioned tools (-x -y247 -z 472) there are user interfaces. Countless ones for ppp, xanim, lha... But they are (as I said) specific and not usable with other tools.

Now here comes Argue. It is a complete argument interface creation kit. With Argue you, THE USER, are able to create a GUI for ALL programs that parse shell arguments. You don't need to know anything about programming or so.

With development going on it is now possible to avoid any shell arguments for Argue and just work on your workbench!

1.18 -Usage-

There are three, eventually four ways of creating an ARGUE ↔
interface.

If you like to create one for a command with a template (COMMAND/A,QUICK/S appears when called with an "?" as argument) click
here

If you like to create one for a command that has no template but ↔
could

easily have it click
here

If you like to create one for a command without a template and ↔
with a unix

fashioned argument style (-x -h35 -M etc.) click
here

1.19 older

Let's have an example run to create a graphical user interace for ↔
... LHA! :-)

We'll now invent a template as it would be used by ReadArgs() fromout LHA.

Let's think about what we need for LHA

- let the user choose archive/extract/list (just three here, could be more)

We need some switches: a=ARCHIVE/S,x=EXTRACT/S,l=LIST/S

- give a gadget for specifying the archive

We'll use ARCHIVE/A (the /A means it MUST be given)

- give gadgets for specifying the files to process

We'll perhaps use three gadgets for that. FILE,FILE,FILE

Now we put those all together, divided by commas

We get: a=ARCHIVE/S,x=EXTRACT/S,l=LIST/S,ARCHIVE/A,FILE,FILE,FILE

How to go on? Continue
here

1.20 newer

Let's have an example run to create a graphical user interace for ↔
... LIST! ;)

The list command is a system friendly tool and it already has a template.
There's no need to invent a new one ;-)

Enter LIST ? at a shell prompt.

The output will begin with something like

DIR/M,P=PAT/K,KEYS/S,DATES/S,NODATES/S,TO/K,SUB/K,SINCE/K,UPTO/K and so on.

Press RETURN to get out of the ?-mode.

This template (DIR/M...) is already good for use with Argue.

How to go on? Continue
here

1.21 nonspace

Differences between

non-spaced unix arguments (-x174 -r5 ...)

and spaced unix arguments (-x 174 -r 5 ...)

for Argue:

- leave out the "nospaces" in your script file.

All other should work as described.

1.22 oldest

See the note about spaced unix arguments
here

Let's have an example run to create a graphical user interace for ↔
... POVRAY! :-)

(This is a fantastic raytracing program, you don't need to get it for this
example ;)

We'll now invent a template as it would be used by ReadArgs() fromout POVRAY.

Let's think about what we need for POVRAY

- let the user choose the scene we want to render

POVRAY uses non-spaced unix arguments. The scene foo.bar would give
the argument +Ifoo.bar. Argue is even able to handle this complicated
thing. See what we make off that!

```
+I=SCENE/A
```

It's important to know a) two different names for one argument can be
specified by the "=" and b) Argue always takes the first name for
the output and the second name for the user interface. This allows the
great compatibility to four different argument styles.

- let the user choose the destination file

This would usually be +O<name>. As you learnt above, here it is:

```
+O=OUTPUT/A
```

- select the width and the height of the picture

Here comes a new thing: We use /N for a number. Usually it would be +W320 and +H256 and so on. Now look:

+W=WIDTH/N,+H=HEIGHT/N

- some other arguments that don't need further description

+V=VERBOSE/S,+FT=TARGA/S,+Q=QUALITY/N

Specifying /K is no more important!!!

Now we put those all together, divided by commas

We get:

"+I=SCENE/A,+O=OUTPUT/A,+W=WIDTH/N,+H=HEIGHT/N,+V=VERBOSE/S,+FT=TARGA/S,+Q= ↵
QUALITY/N"

How to go on? Continue
here

1.23 Continue...

Following actual Argue 0.6 guidelines interface creation has ↵
changed a bit.

First you need a project icon. These have a "Standardtool" (or so) field in the Workbench Information requester. If you don't want to create an own one search for a dummy icon in the Argue distribution and copy it.

Now select this icon and pop up a workbench Information requester.

- enter "Argue" as standard tool. If the Argue binary is NOT in the command path put it there or add a path to "Argue". If you are a program author remember: You can never be sure your users neither have the Argue binary in their path nor the path itself is the same as yours.
- add the tooltype TEMPLATE= and add the template you want after it.

Example: TEMPLATE=FILE/A,SWITCH/S

If your template is too long to fit into the line:

- delete the TEMPLATE tooltype and store the tooltype in the file that represents your icon. So if you're using an icon called Example.info you have to store the template to the file Example.
- if the template is very long (resulting in lots of gadgets) add the tooltype PAGEGROUP

Now read about the new features of 0.6
here

1.24 New features

Argue 0.6 has made a giant step towards user friendliness and easy usage.

Configuring is done via tooltypes of a project icon that has the Argue binary as the standard tool.

All the following tooltypes have their shell argument equivalent to keep compatibility.

Additional tooltypes:

TEMPLATE	The interface template (FILE/A,SWITCH/S)
UNIX	A switch that must be added if unix style is used. (-x 78 +r)
NOSPACES	When UNIX is selected, there are spaces between keyword and argument. To avoid this (for tools that don't understand these arguments) use NOSPACES. (-x78 +r)
COMMAND	This command is automagically executed with the arguments you entered when the "Go!" button is pressed. So you won't need any scripts!
USEKNOBS	MUI 3.3 offers a new integer button that looks very good but has a big size. If you like it, add this tooltype.
TITLE	Specify the title your interface should have. You might tell about the shell program you wrote the interface for or so.
OUTPUT	Specify the output file. If you use a window, always add /AUTO/WAIT/CLOSE after the window description. (You don't really need this tooltype!)
DIR	Don't use. I didn't have time to realize this!
PAGEGROUP	When using lots of arguments the interface might get unreadable. With this tooltype Argue will sort the three kinds of gadgets and put them to a MUI register.
ADDNULLS	Usually Argue would not add integer values to the result string when they are NULL. With this switch ALL integer values are given, even if they ARE NULL.
NOHELP	Switches off bubble help. DO NOT activate this if you give the interface to the public.

ADDFONTSIZE	Usually Argue will cut off the font size from font gadgets (if there is one). <code>topat.font/8</code> will be cut at the /. If the tool Argue interfaces needs the <code>fontsize</code> , add this thing!
NOLOGO	Bored of that annoying logo? Add this thing here and the logo will not be added. DO NOT activate this if you give the interface to the public.
HELPPFILE	displays the given file in a list box under the interface. This is only meant for educational use, but if you like it so much, feel free to use.
WINDOW_ID	this is a nice new thing. You may now give any Argue interface a unique ID string. It will be used for the application title ("Argue [id]") and for the interface window. Now it is possible to snapshot each interface with its own positions.
REQUIRES	an integer argument forcing a minimal version of Argue. this was added because newer versions might confuse earlier ones. so if you like to restrict a GUI for Argue 1.5 and above write <code>REQUIRES=15</code> (without dot!)

Do not use the PERMANENT keyword. This is still in work as I got major problems with POVray through it.

Now keep a look at the ARGUE FAQ
here

1.25 faq

I put some questions and answers here that I thought might be asked very often ;-))

Q: I choose a lot of files in the /M multiple gadget, but they won't be recognized by the tool that eats Argue's output.

A: Command line size is limited to a specific amount of characters. Problems occur when there are really lots of files specified in the /M list. Please do note: Argue has no direct interface to the tool it works for. This won't change until AT= (or the one who develops the OS then) define such an interface.

Q: I tried to adopt a template from some tool with a shell command like "tool >template ?" and whenever I use this thing with Argue the first/last gadgets just show some shit.

A: Edit the template file and search for some unwanted version/program information at the end/beginning. Often tools (Viewtek etc.) add something before the template.

Q: I tried to adopt a template from some tool with a shell command like "tool >template ?" and whenever I use this thing with Argue the first/last gadgets just show some shit.

A: Edit the template file and search for some unwanted version/program information at the end/beginning. Often tools (Viewtek etc.) add something before the template.

Q: Is there any hidden thing in Argue?

A: :)

Q: Why the hell does Argue use virtual groups when I didn't tell it to?

A: With interfaces getting bigger than screen size MUI will get to the idea of using smaller fonts etc. I don't need to tell you this looks absolutely bad. So Argue itself uses virtual groups and MUI won't get these ideas :)

Q: In shell mode Argue tells me something about "Bad args.", but I use right arguments.

A: Put ""s at the beginning and end of the template string. Without, the ReadArgs() function gets confused.

Q: I know some amazing new MUI features and would like to have them implemented in a new version of Argue. Is it good to tell you?

A: Absolutely good! ;-))

Q: I would really like to have a user interface for program xy, but Argue refuses to do the job. Could you help me?

A: Check out EVERYTHING you might have done wrong. Read through all the Argue documentation. Check out all the supplied example interfaces. If you are sure you did nothing wrong, leave me a mail!

Q: Argue simply refuses to work. It doesn't say anything about it.

A: Check out the "requirements" section of this guide. If you are sure you have everything needed by Argue, run it through with the DEBUG

option. (This just works from shell!!!). Argue will give out a debugging report. Mail it to me and I can say more.

Q: Whenever I try to use Argue it crashes and does wild things on my machine!!! Help me!

A: Try to get a debugging report (as described above) and mail it to me. Argue itself should be very reliable in any case. Although, there might be bad bugs in it.

Q: I supply a COMMAND tootype/argument, but it won't be executed.

A: Supply the whole path. Check if you forgot the UNIX or NOSPACES switch.
