

ComAll.doc

COLLABORATORS

	<i>TITLE :</i> ComAll.doc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ComAll.doc	1
1.1	ComAll V1.0	1
1.2	Disclamier	2
1.3	System Requirements	2
1.4	Introduction to ComAll	2
1.5	How to use ComAll	2
1.6	Options	3
1.7	Command to Execute	4
1.8	File Patterns	4
1.9	Destination	5
1.10	Examples	5
1.11	History of ComAll	7
1.12	Comments/Bugs/Questions/Suggestions	7
1.13	Known Bugs	7
1.14	Credits	8

Chapter 1

ComAll.doc

1.1 ComAll V1.0

ComAll V1.0

Copyright © 1995
by
Michael Jaecks

FREWARE

Disclaimer
Don't blame me!

Requirements
System requirements.

Introduction
What do I do?

Usage
How to use me.

Examples
Ways to use me.

History
When? Where? What? How?

Comments?
Where you can find the author.

Known Bugs
Oh no!

Credits
I'm not worthy! I'm not worthy!

1.2 Disclamier

I will not be held responsible for any damage that ComAll may cause to you or your hardware.

USE AT YOUR OWN RISK!

Please, ComAll may freely redistributed. However, no charge (other than the cost of the disk/mailler) may be made upon another user of this program.

1.3 System Requirements

ComAll was written in C with registered Version #2.06.39 of DICE.

My system consists of:

- Amiga 500
- 1 meg CHIP RAM
- SupraRAM (4 megs FAST RAM)
- SupraTurbo 28 accelerator
- 2.04 ROM
- 2.1 WB/OS

You will need to have at least OS 2.0+ to run this program. I have no access to OS 3.1+, so I cannot claim compatibility.

1.4 Introduction to ComAll

Once in a while, I come across a good program that does not handle ↔ the input of multiple files the way I would like it to. An example would be a program that compresses text files, but only one at a time. Thus, if you had five text files to compress, you would have to execute the program five times. Or the program overwrites the input file automatically, but you wanted the output file to have a different filename (One with an extra extension perhaps.). This is where ComAll comes in.

ComAll executes another program on a specified range of input files. Then outputting the files with either duplicate filenames or user-specified filenames.

Check out the

Examples
section for a better understanding.

1.5 How to use ComAll

ComAll is a CLI-based utility, thus it may not be started from ↔ Workbench.

Invoking ComAll by itself will list the appropriate command line options.

You may invoke ComAll like so:

```
>ComAll
    -[ipsx]
    <Command>
    <File Pattern>
    [Destination]
        ^         ^
    :             :
```

These MUST follow the ComAll issued command in this order. Thus, don't put something like "-ip" after the "<Command>".

NOTE: [] indicate an OPTIONAL entry
<> indicate a REQUIRED entry

When using archiving programs (i.e. LhA), you may need to switch the "<File Pattern>" and "[Destination]" entries. See Examples

.

1.6 Options

NOTE: In order to use the following options, you must make sure to use ONLY ONE "-" in front of them. ←

Option:

- i Enables the use of extended information. When finished, ComAll will print out a list of how many files were successfully processed. If any of the files were not processed correctly, a list of these files will be displayed. This is useful in going back and seeing why they failed.

NOTE: The error report of this option really depends on how well written the program you are executing is. ComAll checks for a return value of zero. If no zero is returned from the executed command, then it is presumed an error occurred on the input file. Some authors write programs that return a zero, even if an error occurred.

- p Enables the removal of prefixes from input filenames. Check out the

Examples section for ways to use this feature.

- s Enables quotation marks to be put around ALL input/output filenames. This feature is best used when SPACES occur in the filenames.

You may be asking why I didn't make this a default. Well, I do not like using SPACES in my filenames and use the UNDERSCORE character instead. Plus, I've come across a few programs that didn't like the use of quotation marks around the filenames.

If you know the program you are going to execute through ComAll does not care about quotation marks, you might be better off using this option.

- x Enables the removal of extensions from input filenames. Opposite of the "p" option.

NOTE: You MAY NOT use both the "p" and "x" options together. If someone requests, I may add the ability to use both in the future. Right now, I don't need it.

1.7 Command to Execute

The command you wish to execute through ComAll MUST BE VALID and ↵
in the
search PATH of executable files. Otherwise, a fatal error might occur.
See

Known Bugs

.

A single command may be issued, or one with command line options. In the case of command line options you MUST enclose the entire command in quotation marks. See

Examples

.

I've found that ALIASES may even be used by ComAll. For example, you might have an alias "l" defined as "LhA a". In this case, no quotation marks are needed around "l".

1.8 File Patterns

Valid file patterns (wildcards) are "*", "#", "?" or any ↵
combination of the
three.

SPECIAL CASE:

There is a time when the "@" character may be used in the file pattern. When "@" is used, the output filename will replace it. This is most useful when using LhA, etc. See

Examples

.

NOTE: When the "@" is used in the file pattern, you MUST use a wildcard in the destination entry. See

Destination

1.9 Destination

The destination may be any VALID device, such as "RAM:", "DF1:", ↵
etc. See

Examples

. If you enter NO destination, then the output filenames mirror
the input filenames.

SPECIAL CASE:

There is a time when a wildcard may be used in the destination. This
case only exists if the user uses the "@" character in the file pattern.
See

File Patterns

. Again, this comes in handy when using LhA.

1.10 Examples

Here are some examples to get you started:

```
>ComAll
```

Lists the appropriate command line options used by ComAll.

```
>ComAll compress *
```

Compresses all files in the current directory. All files are over-
written.

```
>ComAll compress #?
```

SAME AS ABOVE!

```
>ComAll compress ?
```

Compresses all files with ONLY ONE character in their filename. All
these files are overwritten.

```
>ComAll compress *.txt
```

Compresses all files that end with ".txt". All these files are over-
written.

```
>ComAll compress RAM:* DF1:
```

Compresses all files in the "RAM:" directory. All compressed output
files are saved to the "DF1:" directory.

```
>ComAll compress RAM:* DF1:@
```

SAME AS ABOVE!

>ComAll compress RAM:*.txt DF1:@.cmp

Compresses all files in the "RAM:" directory ending with ".txt". All compressed output files are saved to the "DF1:" directory. An extension of ".cmp" is added to each output file.

>ComAll -i compress *

Compresses all files in current directory. Extended information will be displayed after all files are processed.

>ComAll -p compress *

Compresses all files in current directory. Any prefixes are removed from the output filenames.

>ComAll -s compress DF1:* RAM:cmp.@.bak

Compresses all files in the "DF1:" directory. All compressed output files are saved to the "RAM:" directory. A prefix of "cmp." and an extension of ".bak" are added to each output filename. Quotation marks are used around ALL input/output filenames. Useful if SPACES exist.

>ComAll -x compress * DF1:@.bak

Compresses all files in current directory. Any extensions are removed from the output filenames. All compressed files are saved to the "DF1:" directory. An extension of ".bak" is added to each output filename.

>ComAll -ips compress *

Compresses all files in current directory. All filenames are enclosed in quotation marks and any prefixes are removed from output filenames. Extended information is displayed after all files are processed.

>ComAll -ix compress RAM:*.txt DF1:

Compresses all files in the "RAM:" directory ending with ".txt". All compressed files are saved to the "DF1:" directory without extensions. Thus, all files would just be missing the ".txt".

>ComAll copy RAM:* RAM:@.bak

Copies all files in the "RAM:" directory to the "RAM:" directory with an extension of ".bak" added to each output filename. Useful for making backups of multiple files.

>ComAll rename * @.txt

Renames all files in current directory to ones with a ".txt" extension.

>ComAll "LhA a" RAM:@.LHA DF1:*

Creates separate compressed LhA files from all files in the "DF1:"

directory. All output files are saved to the "RAM:" directory with an extension of ".LHA" added to each output filename. NOTICE the quotation marks around the LhA command. VERY IMPORTANT!

```
>ComAll "LhA a" @.LHA DF1:*.txt
```

Creates separate compressed LhA files from all files in the "DF1:" directory ending with a ".txt" extension. All output files are saved to the current directory with the ".LHA" extension added.

```
>ComAll "LhA a" RAM:@.LHA DF1:mod.*
```

Creates separate compressed LhA files from all files in the "DF1:" directory that have a prefix of "mod.". All output files are saved to the "RAM:" directory with an extension of ".LHA". This is useful if you have a disk full of ProTracker modules that you need to archive.

```
>ComAll l RAM:@.LHA DF1:mod.*
```

The "l" is an alias defined as "LhA a". Otherwise, this does the SAME AS ABOVE.

1.11 History of ComAll

Version:

1.0 (5-May-1995) First release to the public.

1.12 Comments/Bugs/Questions/Suggestions

If you have any comments, bugs (Some that I haven't already told you about.), or questions, please feel free to contact me.

I may be reached at the following locations:

E-Mail: mjaecks@cup.portal.com

Snail Mail: Michael Jaecks
1415 Park Avenue
Alamogordo, NM 88310
USA

IRC: MotoXR (I join #amiga occasionally!)

1.13 Known Bugs

This is the part I hate. Maybe some suggestions would help in a ↵
future
release.

ComAll does not have the best error detection around. In fact, only parsing the command line is about it.

- 1) Hitting CTRL-C during ComAll's operation may result in a system crash. Just say NO!
- 2) Some programs overwrite existing input files nicely, while some don't. ComAll crashes when trying something like:

```
>ComAll copy *
```

or

```
>ComAll copy * @
```

I have no clue as to how to fix this problem right now. The "copy" command reports "Can't open <filename> for output - object is in use". Same thing goes for the "move" command. This only happens if both the input filename and output filename point to the SAME destination.

- 3) ComAll does NOT check to see if the
Command
exists. So if you try
to execute something that isn't in your path, you may get some
unwanted results. I thought of checking to see if it exists, but
then the user would have to enter the FULL pathname, such as
"SYS:Utilities/LhA a". This also means you couldn't use aliases.
Just make sure to check the spelling before hitting RETURN. SORRY!

1.14 Credits

I would like to thank the following:

Me

You - For using this program. Or at least for reading this guide.

Matt Dillon - For creating a great C compiler known as DICE.

ESCOM - (In Advance) For supporting the American AMIGA users. If not, I might have to put out a special version of ComAll with just this part of the guide deleted or ESCOM flamed. :)