

CLIARGSLIB

Conversion program

COLLABORATORS

	<i>TITLE :</i> CLIARGSLIB	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY	Conversion program	October 9, 2022
<i>SIGNATURE</i>		

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	CLIARGSLIB	1
1.1	Overview of CLIARGSLIB	1
1.2	CLIARGSLIB	1
1.3	CLIARGSLIB	2
1.4	CLIARGSLIB	2
1.5	CLIARGSLIB	2
1.6	CLIARGSLIB	3

Chapter 1

CLIARGSLIB

1.1 Overview of CLIARGSLIB

Overview

An Acid Software Library

Converted to AmigaGuide by

Red When Excited Ltd

Used with the permission of Acid Software

Edited, fixed and cleaned by Toby Zuijdveld 26/02/1999.
mailto:hotcakes@abacus.net.au

1.2 CLIARGSLIB

Statement: NumPars

Modes :

Syntax : NumPars pars.w=NumPars Number of paramters passed to program.

The NumPars function allows an executable file to determine how many parameters were passed to it by either Workbench or the CLI. Parameters passed from the CLI are typed following the program name and separated by spaces.

For example, let's say you have created an executable program called myprog, and run it from the CLI in the following way:

```
myprog file1 file2
```

In this case, NumPars would return the value '2' - 'file1' and 'file2' beng the 2 parameters.

Programs run from Workbench are only capable of picking up 1 parameter through the use of either the parameter file's 'Default Tool' entry in it's '.info' file, or by use of multiple selection through the 'Shift' key.

If no parameters are supplied to an executable file, NumPars will return 0.

During program development, the 'CLI Argument' menu item in the 'COMPILER' menu allows you to test out CLI parameters.

1.3 CLIARGSLIB

Statement: Par\$

Modes :

Syntax : Par\$ parameter\$=Par\$(Parameter#) Returns the string value of a ↔ parameter.

Par\$ return a string equivalent to a parameter passed to an executable file through either the CLI or Workbench. Please refer to NumPars for more information on parameter passing.

NOTE: If the parameter asked for is a directory/device/volume etc (IE NOT A FILE) then Par\$(#) will return an empty string. This is one way you can check to see if a file was passed or not.

1.4 CLIARGSLIB

Statement: FromCLI

Modes :

Syntax : FromCLI result=FromCLI 0=Run from WorkBench, -1=Run from CLI

Returns TRUE (-1) if your program was run from the CLI, or FALSE (0) if run from the WorkBench.

1.5 CLIARGSLIB

Statement: ParPath\$

Modes :

Syntax : ParPath\$ path\$=ParPath\$(Par#, #type) Full/Dir path of WB arg. (v36+ ↔ only!)

This returns the path that this parameter resides in.

'type' specifies how you want the path returned.

0 You want only the directory of the parameter returned.

1 You want the directory along with the parameter name returned.

EG:

If you passed the parameter "FRED" to your program from WorkBench, and FRED resides in the directory "work:mystuff/myprograms" then ParPath\$(0,0) will return "work:mystuff/myprograms", but ParPath\$(0,1) will return "work:mystuff/myprograms/FRED".

CAVEAT

The way WB handles argument passing of directories is different to that of files. When a directory is passed as an argument, ArgsLib gets an empty string for the name, and the directory string holds the path to the passed directory AND the directory name itself. EG

Passing the blitz2 directory to a program will result in:

Par\$(x) Being an empty string.

ParPath\$(x,0) Being something like work:Basic/blitz2.

ParPath\$(x,1) Being work:Basic/blitz2/

YES! The / is appended! This is because to keep things simpler, and more uniform ParPath\$(x,1) Is the concatenation of

1) The directory string passed by Workbench

AND

2) A / followed by the name given by WorkBench.

So you can see why the / followed by the empty string occurs.

The easy way around this is simply to check Par\$(x), if it is empty, then use ParPath\$(x,0), if it isn't (IE a file was passed) use ParPath\$(x,1) and you will have the entire pathname of the file OR directory.

See the demo program, which handles both cases.

NOTE 2: Is only useable from WorkBench, you will get an error if your program was run from the CLI and you try to call ParPath\$.

1.6 CLIARGSLIB

CLIARGSLIB

Overview

Command Index

FromCLI

NumPars

Par\$

ParPath\$
