

ShellScr

Kyzer/CSG

COLLABORATORS

	<i>TITLE :</i> ShellScr		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Kyzer/CSG	January 2, 2023	

REVISION HISTORY

<i>NUMBER</i>	<i>DATE</i>	<i>DESCRIPTION</i>	<i>NAME</i>

Contents

1	ShellScr	1
1.1	RealSize v1.1 documentatino	1
1.2	Program History	2
1.3	Credits	2
1.4	Introduction to RealSize	2
1.5	Usage of RealSize	3
1.6	Workbench 1.3	4
1.7	Known bugs and limitations in v1.1	4
1.8	source	4

Chapter 1

ShellScr

1.1 RealSize v1.1 documentatino

RealSize v1.1

Introduction

Usage

Source

Program history

Credits

WB 1.3
compatible

RealSize is a program to calculate the real filesize of crunched files.
Copyright (C) 1997 Kyzer/CSG

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Contacting the Author

1.2 Program History

1.0: First release

1.1: Added WIDTH/N option

Bugs? But there's no bugs.

1.3 Credits

RealSize was conceived by Kyzer/CSG using the Amiga E language and uses the xfdmaster.library by Georg Hörmann.

Thanks to Mikael Lund for the idea to write this, even if I can't spell 'Mikael' right.

Contact:

Kyzer/CSG,
49 Fairview Road,
AB22 8ZG, Scotland.

or email: kyzer@4u.net

Incitement Works plc, <http://www.abdn.ac.uk/~u13sac/>

1.4 Introduction to RealSize

This is a simple program that prints the size of a file. "So what?" you may say, "I can do that with `list lformat %l`". But the main purpose of RealSize is that it can open the great xfdmaster.library, and if the file is compressed it can decompress it and print the uncompressed size.

At it's most basic, RealSize prints the listed size of the file as stored by the filesystem. But if RealSize can open and read in the file, then it will try and decrunch it. If it succeeds, the decrunched size will be printed. If it fails, the read-in size will be printed.

RealSize also supports the xfdmaster V38 feature of calculating the size without actually decrunching.

1.5 Usage of RealSize

The template is `RealSize FILE/A,NE=NOEXTERN/S,WIDTH/N,NOLINE/S` or more traditionally, `realsize <file> [NOEXTERN] [WIDTH=n] [NOLINE]`

`WIDTH/N` is the number of spaces you want the output size to be formatted into. If `WIDTH` is a negative number, the positive value will be used as the number of spaces, but the output will be left-justified rather than right-justified. If `WIDTH` is 0 or not mentioned, no special formatting will be used. This option does not exist when using

```
WB 1.3
```

.

If you state the `NOEXTERN` switch, then `RealSize` will exclude `xfd`'s external slaves. This is in case they are buggy or broken.

The switch does not exist when using

```
WB 1.3
```

.

The `NOLINE` switch in `RealSize` is the same as the one in `"echo"` - if you set it, `RealSize` will not take a new line, allowing you to concatenate output on the same line. The switch does not exist when using

```
WB 1.3
```

.

You must state the filename, possibly with a path, of the file you want `RealSize` to print the size of.

If everything went OK, the uncrunched size of the file will be written to the shell (`stdout`). This may be redirected, or used in AmigaDOS scripts using backwards quotes. See the example script for some info.

If the arguments you give to `RealSize` are bad, or the file you mention does not exist, `RealSize` will print nothing, only return the code `FAIL (20)`. If `RealSize` can get at least some info about the file, it will print the size it managed to get, but as this may not be perfectly right it will return the code `WARN`.

`RealSize` will set the local variable `$RealSizeError` with a short description of any error or warning that stopped it from completing properly. Note that with

```
WB 1.3
```

```
this requires the 'set' command to be available, usually in C:
```

Fatal errors:

- bad args (you must supply only one filename)
- can't examine file (file doesn't exist or is not a file)

Warnings:

- | | |
|--|------------------------|
| - can't open file (may be read protected) | ondisk size is printed |
| - out of memory (or largest block is too small) | ondisk size is printed |
| - read error (corrupt disk?) | ondisk size is printed |
| - can't open <code>xfdmaster.library</code> (v37+) | read size is printed |
| - can't allocate XFD buffer | read size is printed |

- not crunched/unknown cruncher read size is printed
- file needs password or key read size is printed
- couldn't decrunch file (no mem/data corrupt) read size is printed

1.6 Workbench 1.3

Workbench 1.3 (kickstart v34) is old and poop.
This sucks! Change it!

1.7 Known bugs and limitations in v1.1

Actual problems that may be fixed:

- Doesn't 'recognize' executable files that are crunched, only data files.
- does not work on fifo/pipe files, eg IN: or PIPE:bla or such. Why? Well, to use XFD requires that the whole file is in memory. A pipe file never reveals it's size, you just have to keep on reading data until the end. At the moment RealSize will only read the first 'gulp' of a pipe, then complain that the crunched data is incomplete and can't be decrunched. It may be enough to decide the size of a few crunched formats like StoneCracker data and CrM data, but not PowerPacker.
- Argument parsing on
 WB 1.3
 is poo (but it works - just)
You must put only the filename on the command line, no quotes, no spaces no comments and no switches. I may fix this by using arp.library if someone needs it.

Limitations that cannot be fixed.

- Does not like files over 2Gb. First, you'd have to allocate 2Gb of memory and read it all in (long snooze!), then you would realise that even with virtual memory, the maximum size of address space is 4Gb. Now, 2Gb file compressed + uncompressed is WELL over 4Gb limit. Anyway, if you could manage this physical impossibility, your file size would be printed as a negative number. Sorry.
- Can't decrunch password-protected files. That would require you to give the password, and currently I haven't implemented a way to do this.

1.8 source

Source is supplied, written in Amiga E.

Bugs? But there's no bugs.

Also, as the Emodules provided with xfd are out of date compared ↔
to the

other includes, I have converted the xfd V38 includes to Emodule format.
See the emodules/ directory.
