Contents

Introduction

This program provides a common framework for the execution of a number of general UNIX utilities. In particular, it provides a Single Document Interface (SDI), with common menu commands, display facilities and printing services. The program can be invoked directly from Program Manager; the utilities can be invoked individually from the WinXs menu in File Manager.

Utilities

See Also

Csplit

Name

csplit - split files based on context

Description

The csplit utility reads the named input file and separates it into n+1 sections, defined by the arguments arg1...argn in the Arguments list. By default, these sections are written to files xx00...xxn, as follows:

```
xx00  From the start of the input file to the first
      argument in the Argument list (arg1).

xx01  From arg1 to the second argument in the Argument
      list (arg2)
...
xxn   From the last argument in the Argument list (argn)
      to the end of the input file
```

Both the prefix of output files (xx) and the number of decimal digits (00...n) can be changed.

Operands

FIlename

Specifies the name of the input file. Only one filename can be passed to each invocation of this program and it cannot contain DOS wild-card characters. Select a new or different filename by using the associated Browse button.

Arguments

This field provides the context on which file splitting will be based, as a series of space separated arguments. The following argument types are accepted by this version of csplit:

- /regexp/[offset] Create a file using the contents of the lines from the current line up to, but not including, the first line that matches "regexp". The syntax of regular expressions accepted by this utility is the same as that defined for the Grep utility. The optional offset is a positive or negative integer, representing a number of lines after (+) or before (-) the matched line, to be included in or excluded from the output file. If this causes the resulting file to have zero size, the offset is ignored. After an output file is created, the current line is set to the next unwritten line in the input file.
- %regexp%[offset] This is the same as the previous argument, except that no output file will be written for the selected section of the input file.
- line_no Create a file from the current line up to, but not including, the line number "line_no". Lines in the input file are numbered starting at one. After an output file is created using this type of argument, the current line becomes "line_no".
- {num} Repeat the previous argument "num" times. This argument can follow any of the argument types described above. If it follows a "regexp" type argument, that argument will be applied "num" more times. If it follows a "line_no" argument, the input file will be split every "line_no" lines, "num" times, from that point.

Options

- Prefix. Name the output files "prefix00", "prefix01", etc.. The default prefix is "xx".
- Digits in Filenames. Use the specified number of digits when creating output files (the default is 2).

Examples

An argument list containing "/^}/+1 {n}" will split a C source file into a section for each function definition, where "n" is some suitably large number.

An argument list containing "50 {10}" will split a file into 1 file of 49 lines (there is no line number zero) and 10 files each of 50 lines. If the input file contains more than 549 lines, the last file will contain line numbers 550 to end-of-file.

Du

Name

du - estimate file space usage

Description

This program is a Windows-based implementation of the UNIX du(1) command, which displays the amount of disk space occuppied by files and directories. Operation begins at the current directory, or in the directory given as a command argument (if any), and proceeds through all subfiles and subdirectories. Optionally, the command can be directed to list the size of each subfile, or output can be limited to subdirectories only. The final line of output gives the total number of blocks occuppied by this directory and all its descendents.

Output consists of a series of text lines, one per file/directory, each giving the size of a file/directory in 1024 or 512 byte units and the name of the file/directory. Menu options permit output to be copied to the Windows clipboard or sent to a printer.

Operands

Directory

Identifies the directory at which the output listing will begin. This value can be changed by double-clicking a directory in the directory listbox.

List All Files

If checked, this option causes all subfiles and subdirectories to be included in the output. By default, only subdirectories are listed.

Exclude Subdirectories

If checked, this option prevents subdirectories being included in the output, meaning that only a total for all subfiles and subdirectories will be listed.

Show Utilization

If checked, this option causes a space utilization figure to be displayed after the block count, as a percentage of used space versus allocated space.

On FAT file-systems, space is allocated in fixed units which grow with the size of the disk partition holding the file-system. Minimally this unit size is 2kb, but can grow to as large as 32kb on 1.0 gigabyte partitions. The upshot of this is that large amounts of disk space may be wasted as a result of large allocation units. Use this option to check how much wasted space is accumulating on your file-systems (the lower the percentage, the more space is being wasted).

There are various ways to improve the utilization figure. One way is to reformat your disk with a number of smaller partitions; another way is to use DriveSpace (or equivalent) to create compressed partitions within a large non-compressed partition (the allocation unit size for DriveSpace is typically 8kb). This second solution also has the advantage of using disk space more effectively.

Units

This controls the units in which file/directory sizes will be displayed; that is, 1024-byte (the default) or 512-byte.

File

Name

file - determine file type

Description

The File utility performs a series of tests on each specified file in an attempt to determine its contents. This version of File checks the following:

1 If the files extension (.ext) is registered in the Windows Registration Database, then the associated registration database description is used to identify the file.

2 The intial segment of the file is examined and its contents are matched against a series of magic numbers built into the File program. If a match is found, this is used to identify the file.

3 If all else fails, a simple test is performed to determine if the file is an ASCII text file or a data file (i.e., of completely unknown type).

The efficacy of this utility is largely dependent on the contents of the Windows registration database; that is, systems with large numbers of registered applications are more likely to produce accurate results than systems with only a few. Note also that when a matching entry is found in this database, the associated text will be used to identify the file. This can sometimes be terse, and often is simply the name of the executable file used to process the file.

Magic numbers provide a second line of attack. These are a series of built-in database entries that identify a file offset and a value which, if present at that offset, will uniquely identify the contents of the file. This database can be expected to grow with time.

Unlike the UNIX version of file(1), no attempt is made to distinguish program text from English language text, partly because these checks often produce the wrong results, and also because the text of other natural languages cannot always be discerned according to the rules of the English language.

Operands

File(s)

Identifies the file or files to be checked. By default, this field will be set to "*.*"; that is, all files in the current working directory will be checked. One or more filenames (possibly containing DOS wild-card characters) can be substituted, separated by space characters.

Current Directory

Identifies the current working directory, which can be changed by double-clicking a directory name in the associated listbox. Drives can be changed by double-clicking a drive name.

Check Magic First

To try and identify the contents of each file, File will normally check the Windows Registration Database for a matching file extension; if this fails, it will then scan the magic number database. Checking this option reverses the order in which these databases are used, so the magic number database is checked first, followed by the Windows Registration Database. This is useful, for example, when looking for UNIX files stored in a specific format (e.g., compressed).

Debug

Because the results produced by this utility can sometimes be puzzling, File provides a mechanism

through which users can determine how those results are being reached. Selecting the button marked None (the default), prompts File to perform normal file checking. Selecting "Registration" causes the contents of the Windows registration database to be displayed, and selecting "Magic numbers" causes the built-in database of magic values and offsets to be displayed.

Output

Debug=None

Each checked file is displayed on a separate line in the form:

<filename>: <description>

where the <description> identifies the contents of the file.

Debug=Registration

File performs a serial scan of the Windows registration database and produces one line for each entry in the database with an extension (.ext) as its key. Output is in the form:

<.ext> <ClassName> <ClassDescription>

where <ClassName> is an application-defined string that identifies the registration class with which <.ext> is associated, and <ClassDescription> is the ASCII text string associated with <ClassName>. Either or both <ClassName> and <ClassDescription> can be blank.

Debug=Magic Number

File lists the contents of the built-in magic number database in the form:

<offset> <length> <magic number> <description>

where <offset> is the file offset, in bytes, at which to check for a <magic number> of <length> bytes. <description> is the text string that will be output by File when a match is found.

Split

Name

split - break a file into separate pieces

Description

This utility reads an input file and writes one or more output files. The default size of each output file is 1000 lines. The size of output files can be modifed by specifying ouput files of different units (lines, bytes, kilobytes or megabytes), and different numbers of units/file.

Each output file is created with a different suffix. The suffix consists of 3-letter sequences, incremented as if they were base-26 numbers. For example, using the default suffix "xaa", the first output file will be named "<file>.xaa", the second will be "<file>.xab", and so on to "<file>.zzz"; where <file> is the same as the input filename.

If the number of files required exceeds the maximum allowed value of the suffix (i.e., ".zzz"), Split will fail when it tries to create the next output file. Files already created by the operation will be left intact.

This version of the program also provides an Unsplit facility, which allows output files generated by a previous Split operation to be formed back into a single large file.

Operands

Filename

Specifes the pathname of the input file (Split), or the pathname of the file to be generated (Unsplit). Output filenames will be generated using the file-prefix from this name and starting at the extension value given below. For example, if the input filename is "bigfile.txt", output filenames will start at "bigfile.xaa", increasing by 1 up to the last required filename ("bigfile.zzz" being the absolute limit).

Extension

The 3-character extension to use when generating output filenames, or the starting extension for Unsplit operations. By default, this value is set to "xaa".

Type

Identifies whether this is a Split or an Unsplit operation (Split by default).

Units (Split Only)

Indicates the unit type (for splitting operations) and the maximum number of units to be written to each output file. Units may be either Lines, Bytes, Kilobytes or Megabytes. The units/file value will be scaled according to which of these unit types is selected.

Strings

Name

strings - find printable strings in a file or files

Description

Ths Strings utility looks for printable strings in any type of file and displays those strings in the output window. A printable string is defined as a sequence of four (by default) or more printable characters terminated by a newline or NUL character. This command is often used on UNIX systems to locate copyright statements in object files, or to extract error messages.

By default, Strings will search for sequences of 4 or more printable characters. This value can be changed by the user, and the displayed lines can optionally be preceded by their offset in the input file, displayed in decimal, octal, or hexadecimal format.

Operands

File(s):

One or more filenames (possibly containing DOS wild-card characters) identifying the file or files to be scanned. Multiple filenames appearing in this field should be separated by space characters.

Current Directory:

The drop-down listbox can be used to select a file or directory. Double-clicking a file in the listbox causes that filename to be added to the "File(s):" field; double-clicking a directory name causes that directory to become the current working directory, and its name to be displayed in the "Current Directory:" field.

String Length:

By default, Strings will look for printable strings of 4 or more characters. This value can be changed by typing a different number in this field.

Offsets In:

Allows the file offset of located strings to be displayed in decimal, octal, or hexadecimal format. By default, file offsets are not displayed.

tail

Name

head - Display the first part of a file
tail - Display the last part of a file

Description

These utilities display the first part or the last part (respectively) of one or more text files. In the case of Head, it is also possible to skip the first N lines in a file, thus allowing the mid sections of a file or files to be displayed.

Typically, Tail is used to display such things as the latest entries in log files; Head can be used to display selected pieces of text from very large files such as dictionaries.

Operands

File(s):

One or more filenames (possibly containing DOS wild-card characters) identifying the file or files to be processed. Multiple filenames appearing in this field should be separated by space characters.

Current Directory:

The drop-down listbox can be used to select a file or directory. Double-clicking a file in the listbox causes that filename to be added to the "File(s):" field; double-clicking a directory name causes that directory to become the current working directory, and its name to be displayed in the "Current Directory:" field.

No of Lines:

Specifies the number of lines to be displayed in each input file (default 10).

Skip Lines:

If Head is selected, specifies the number of lines to be skipped in each input file before displaying the number of lines indicated by "No of Lines:". This is useful for displaying the mid sections of a file. The contents of this field are ignored if Tail is selected.

Program:

- Head. Display the first "No of Lines:" in each input file, optionally skipping over "Skip Lines:".
- Tail. Display the last "No of Lines:" in each input file.

Wc

Name

wc - count words, lines and bytes in a file or files

Description

This program is a Windows version of the UNIX wc(1) command, which counts the lines, words and bytes in one or more text files. A separate set of counts is produced for each file scanned, which are totalled at the end of the output display. Output is in the form:

<lines> <words> <bytes> <filename>

Optionally any or all of <lines>, <words> and <bytes> can be omitted.

Operands

Filenames

One or more filenames (possibly containing DOS wild-card characters) identifying the file or files to be scanned. Multiple filenames appearing in this field should be separated by space characters.

Directory

Specifies the current working directory. This can be changed by double-clicking a directory name in the directory window.

Count Lines

If checked, causes the number of lines in each file to be output; or, more accurately, the number of newline characters encountered.

Count Words

If checked, causes the number of words in each file to be output. Words are considered to be delimited by white-space, tab and newline characters.

Count Bytes

If checked, causes the number of bytes in each file to be output.

Menus

File Menu

New

Restart the current utility with a new set of options.

Save As...

Save results to a file.

Switch to...

Switch to another utility in this package.

View

View selected file. Select a filename anywhere in the output window (e.g., by double-clicking the left mouse button over the appropriate name) and then select this item to pass the filename to more for viewing.

Print

Send the contents of the current output display, or the current selection, to a printer.

Print Preview

Preview the output display for printing.

Print Setup

Invoke the standard Windows printer setup dialog.

Exit

Terminate the program and return to Windows.

Edit Menu

Copy

Copy the current selection to the WIndows clipboard.

Help Menu

Contents

Go to the contents topic associated with this help file.

About

Display program name, copyright statements and program version number.