WHAT?format

A file format recognition utility for
IBM PCs and compatibles

Version 30.0

Boots & Pepper

WHAT?format is copyright © Boots & Pepper 1988-91.

It may be distributed freely provided no changes are made to WHAT.EXE or the WHATDOC files. The program may *not* be bundled or distributed with commercial software without the author's permission.

WHAT is shareware. If you find it useful, please register your copy by sending $15 (NOK 100) to:

> Boots & Pepper, Pilestredet 97, N-0358 Oslo, Norway.
> Giro: 1730.18.96921

Registration entitles you to a free copy of the next public release of the program. Other feedback – bug reports, information about formats not currently supported, whatever – may be sent to the address above or to:

> CompuServe: 76057,246 (Steve Pepper)
> email: pepper@falch.no (from medio May 1991)

Boots & Pepper is Steve Pepper and Dag (Boots) Hasvold. We run Computertext BBS – a bulletin board specialising in aspects of computing in the printing industry (DTP, PostScript, SGML, text conversion etc.) – on +47-2-162650 (2400-8N1).

WHAT?format 30.0

A file format recognition utility for
IBM PCs and compatibles

Introduction

WHAT?format started life as a simple utility whose purpose was to distinguish between text files created by WordStar and WordPerfect. It was originally written for people working in a typesetting house which received a lot of raw text on floppy disks – all too often without any information regarding the system that had created the files.

As time has gone by, the program's capabilities have been extended so that the present version can distinguish between the native formats of thirty of the most common word processors. In many cases it will also distinguish between files created by different versions of the same program, e.g. WordPerfect 4.x (4.2 and earlier), 5.0, 5.1, etc.

In addition to native word processor formats, version 30.0 also supports a number of text interchange formats (e.g. Document Content Architecture and Rich Text Format), various text-only formats (ASCII, DOS, EBCDIC, etc.), and a few print-formatted/page description formats (PostScript, PCL, etc.).

Although it is primarily concerned with text files, **WHAT** will also recognise an assortment of other common data formats stemming from database, spreadsheet, graphics and other applications. (A complete list of supported formats is given in Appendix A.) Other features include the ability to create straight hex dumps and character set maps. These are described in detail below under *Options*.

Usage

To use **WHAT** in its basic mode as a format recognition utility, simply type **WHAT** at the DOS prompt, followed by the name of the file to be analysed. The file name may include optional drive and path specifications, as well as standard DOS wildcards:

C:\>WHAT myfile.doc
(analyse myfile.doc in current directory)

C:\>WHAT a:*.*
(analyse all files on disk in drive A:)

```
W H A T ? f o r m a t  3 0 . 0
                                        (c) Boots & Pepper 1988-1991

123-WK1.WHT    1327 Lotus 123 2.0        PCWRITE.WHT    2312 PC-Write
ABILITY.WHT    3421 Ability WP 1.0       PCX.WHT       39324 PCX 3.0 640x720
ACTO-NDS.WHT   1010 Notis M (S-format)   PLANPERF.WHT   6363 PlanPerfect 3.0
ASCII.WHT      1102 ASCII text file      POSTSCPT.WHT  13204 PostScript DSC 1.0
CICERO.WHT    10156 Cicero               REFLEX.WHT    16402 Reflex 1.14
DBASE2.WHT     6144 dBASE II             RICH-T-F.WHT   1913 Rich Text Format
DBASE3.WHT     1901 dBASE III            SAMNA-IV.WHT   2048 Samna Word 1.0
DCA-FFT.WHT    2215 EBCDIC file          SPRINT.WHT     6974 Sprint
DIF.WHT        4522 DIF spreadsheet      SYLK.WHT      11929 SYLK spreadsheet
DISPWR-4.WHT   4608 DisplayWrite 4       VENTURA.WHT    2314 Ventura Publisher
DOS.WHT        3103 DOS text file        VOLKS-DL.WHT   4256 Volkswriter
ENABLE.WHT     1244 Enable WPF 1.7       WDST2000.WHT   2485 WordStar 2000 1.00
MANUSCPT.WHT  12154 Manuscript 1         WINWORD.WHT    4530 Word for Windows
MASS-11.WHT    1626 MASS-11              WORDPF42.WHT   3113 WordPerfect 4.x
MICROWRD.WHT   4235 MicroWord            WORDPF51.WHT   1766 WordPerfect 5.1
MS-WORD.WHT    2768 Microsoft Word       WORDST34.WHT   3256 WordStar 4.0
MULTIMAT.WHT   7680 Multimate            WORDST50.WHT   3768 WordStar 5.0
NOTIS-S.WHT    4892 Notis N (S-format)   WRITE.WHT      3712 Windows Write
ORD20.WHT      7378 Ord 2.0              XPRESS30.WHT  15034 XPress tagged ASCII
PALANTIR.WHT   1128 Palantir 3.0         XYWRITE.WHT    2862 XyWrite

d:\what>_
```

WHAT takes each file matching the file specification and writes its name and size on the screen, analyses it and reports the result. Unrecognized files are reported as being of *UNKNOWN FORMAT*. The file name and size are written to DOS' CON device; the result is written to the standard output device (normally the console), making it possible to send the result to a file using the usual DOS redirection techniques.

The DOS Errorlevel

Each major format supported by **WHAT** has its own *format code* that distinguishes it from all other major formats. For example, WordPerfect has the format

code 32. Upon exiting to DOS, **WHAT** sets the DOS errorlevel variable to the format code corresponding to the result it arrived at for the *last* of the files that were analysed. Thus, if the last file analysed by **WHAT** turns out to be in WordPerfect format, the DOS errorlevel will be set to 32.

This feature can be used in batch files, both to automate various kinds of file processing (conversion, cataloging, etc.) and as a way of ensuring that files of the wrong type do not get sent through a particular process. An example of such a batch file, WPCONV.BAT, is given in Appendix B. (See the description of the **/E** and **/F** switches below for more information on using the errorlevel feature.)

NOTE: **WHAT***'s format codes change from version to version, as new formats are added to the program, so be sure to update your batch files when you receive a new version of* **WHAT**.

The %WHAT% variable

Where possible, **WHAT** attempts to distinguish between different *versions* of the same file format. Thus a WordPerfect file will be identified as version 4.x (meaning 4.2 or older), 5.0, 5.1 or whatever. In the case of bitmap files, **WHAT** will often report the size of the image, and possibly also the number of colours. With a file in MacBinary format, **WHAT** reports the file's TYPE and CREATOR. It is possible to test for all these kinds of information using the %WHAT% environment variable.

Before exiting to DOS, **WHAT** looks to see if the environment variable %WHAT% exists. If it does, **WHAT** sets it to the exact result shown on the screen for the last file analysed, e.g. "WordPerfect 5.1" or

"PCX IV 640x480x256". (Note that this string can be up to 19 characters in length. An error is reported if there is not enough room in the environment.) Appendix B gives an example (SORTPICS.BAT) of a batch file that uses the %WHAT% variable to sort graphics files by format.

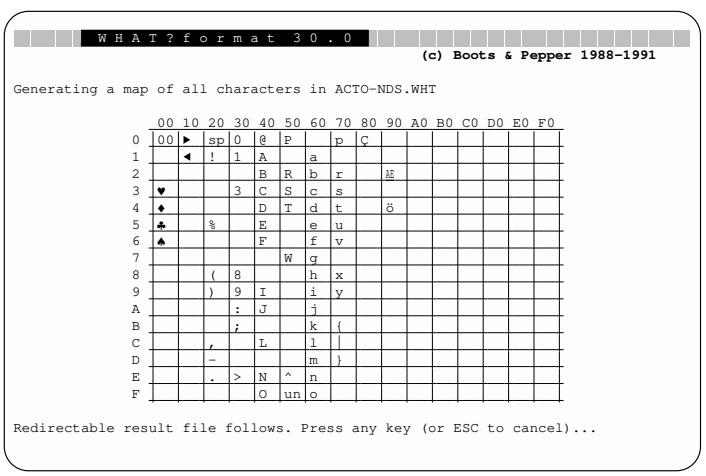
Options

NOTE: All options can be used in either upper or lower case, and may be preceded by either a slash or a hyphen. Those used in conjunction with file names may appear either before or after the file specification.

Character set

Usage: WHAT <filespec> /C [>filename]

Creates an on-screen map of all characters appearing in the first file that matches <filespec>. The user is then given the option of writing more detailed information (including the offset and context of the first occurrence of each character) to the standard output. If redirection has been specified on the command line, the result will be a text file suitable for viewing with Vern Buerg's LIST.



The character set option is particularly useful with plain text files that do not use one of the standard character sets. Note that /C uses the underline attribute in order to create a 16x16 character set matrix on the screen. This gives better results on monochrome than on colour monitors.

Errorlevel

Usage: WHAT /E [>filename]

Generates a list of format codes in a form which can easily be modified to create batch files like those shown in Appendix B. The output can be sent to a file using DOS redirection.

Format code

Usage: WHAT /F<format>

Presents a list of all supported major formats which contain the substring <format>, together with the corresponding format code. The operation is case insensitive. For example:

```
WHAT /fperfect
```

will give the following result:

```
32 WordPerfect
46 DataPerfect
56 PlanPerfect
62 DrawPerfect
```

Hex dump

Usage: WHAT <filespec> /X [>filename]

Creates a hex dump of the first file that matches <filespec>. The output contains only hex values – no file offsets or character equivalents. The main purpose of this switch is to simplify the analysis of long and complicated formatting instructions contained within a text file. (The resulting file is easy to edit

since it only contains hex values.) If you merely want to *view* the contents of a file in hex format, you will be better off using a file browsing utility like LIST, PC-Tools or Norton Utilities that also displays file offsets and ASCII equivalents.

Help

Usage: WHAT /H

Shows **WHAT**'s help screen. The help screen is also shown when **WHAT** is invoked without any command line parameters.

List formats

Usage: WHAT /L

Presents an on-screen list of all file formats supported by the current version of **WHAT**.

Quiet mode

Usage: WHAT /Q <filespec>

Suppresses screen output (for use in batch files).

Redirection

Usage: WHAT /R <filespec>

Enables redirection of all three elements of **WHAT**'s screen output (i.e. the file's name, size and format). Normally only the format is written to DOS' standard output.

Commentary

WHAT is not foolproof, nor is it meant to be. It belongs to the venerable family of Q&D-utilities, and its basic philosophy is to be right as often as possible but without spending all day about it. It is not as Quick as it could be, and it is no doubt a good deal Dirtier than it would have been if I'd been a real programmer. That said, it *has* been tested fairly thoroughly on a number of systems and performs as described in this documentation. No problems have

been reported that would consitute a threat to your computer or data, but as always, no responsibility is taken for damage resulting from incorrect or careless use of the program.

How it works

WHAT works by scanning the beginning of a file and looking for specific formatting features that can identify its format. The precise features looked for vary. Some applications – especially newer ones – create files with headers containing an ID-tag, a kind of "thumbprint" consisting of a special sequence of bytes that the application itself uses to determine whether or not the file is in its native format. For example, all files created by WordPerfect 5.0 or later (and other WP Corp products) begin with the byte sequence FF 57 50 43 (-1,"WPC"). These kind of files are an easy match, and **WHAT** will handle them quickly and flawlessly.

Problems

Other programs present greater problems, especially those with a native format closely akin to pure ASCII. PC-Write, for example, produces ASCII files if the document doesn't contain guide line font commands or text with attributes such as bold, underline etc. Such a file will be reported as being ASCII by **WHAT**.

If on the other hand, the PC-Write document contains a few words that are underlined, the file will resemble an ASCII file interspersed with the odd 17h – a "non-ASCII" character. This will probably be enough for **WHAT** to reach a verdict of PC-Write, but it is not difficult to imagine that the file *could* have been produced by another program and that the 17h means something quite different. In such borderline cases a programming decision has been made based upon the

assumed popularity of particular applications. (If you disagree with the decision, don't hesitate to let me know!) When **WHAT** makes a mistake, it is often in this kind of situation.

More problems

Another example will further illustrate the problems involved in differentiating between word processing systems that use similar formats. I recently downloaded an ARChive file containing a number of text files from a bulletin board system. These files looked like ASCII when I viewed them with LIST, but **WHAT** said they were WordPerfect 4.x. In actual fact they turned out to be UNIX-type ASCII files with line endings marked by a single LF instead of the CR/LF pair used under DOS. (The archive file seems to have been put together on an Amiga.) LF (0Ah) is the code used by WordPerfect to represent a hard return (hence **WHAT**'s diagnosis), so the files could equally well have been prepared using WordPerfect (except that they also had hard returns where there should have been soft returns).

The question here is whether the result reached by **WHAT** was acceptable. My answer – based mainly upon pragmatic considerations – is yes: Wherever the file might have come from, it is now on a PC (otherwise I wouldn't be using **WHAT**!), and if it is to be edited on a PC, the best program to use is WordPerfect. Most ASCII editors would complain bitterly about the missing CR at the end of each line; but WordPerfect is over the moon, and it will even allow me to regenerate most of the soft returns (by reading in the file, saving it as DOS text, and reading it in again, this time as DOS text, using the option of converting hard returns in the hyphenation zone to

soft returns). So in this case, WordPerfect is the best answer – even though strictly speaking it is the wrong one.

Dirty tricks

If there is one thing that really slows **WHAT** down it is a lot of files in unsupported formats. A couple of dirty tricks are used to minimise this problem. Firstly, **WHAT** never reads more than the first 5 Kb of a file, reasoning that if it hasn't made up its mind by then, it probably never will. This could in theory lead to problems. For example, a PC-Write document consisting of 2–3 pages of straight ASCII followed by a few pages of heavily formatted text will be judged to be ASCII – but you'll be in trouble if you try to import it to, say, WordPerfect as "DOS Text". Such situations occur so rarely in practice, however, that the speed advantages of just looking at the beginning of a document outweigh the potential disadvantages.

Secondly, **WHAT** doesn't bother to try to ascertain whether a COM-file really is executable: The present version quite simply ignores files with the extension .COM (except when the only files that match the file specification have this extension, in which case **WHAT** will attempt to analyse the *last* one – hopefully unsuccessfully).

ASCII files and DOS files

The criterion for differentiating between what **WHAT** calls "ASCII text files" and "DOS text files" is whether or not characters from the Extended ASCII set appear in the file. An ASCII file can only contain 7-bit characters. This is an important distinction in certain European countries where accented characters may be represented by national versions of the (7-bit) ISO 646 character set, so English-speaking users will

just have to live with it! In neither format does **WHAT** expect to encounter any control characters other than TAB (09h), CR (0Dh), LF (0Ah), FF (0Ch) or a single Control-Z end-of-file marker (1Ah).

Feedback

The biggest problem with a program like **WHAT** is keeping it up to date. New word processing programs are appearing all the time, and most of them use their own native format. Occasionally the format is described in the documentation that follows the application, but usually that is not the case. Some software publishers are willing to make the details of the format available to developers; others (like Microsoft and IBM) keep them a closely guarded secret.

Upgrades of existing programs also present problems. As new formatting features are added to the application, the native format changes in order to accommodate them. Sometimes these changes amount to no more than the addition of new codes to the old format (as when WordPerfect was upgraded from 4.1 to 4.2). More major revisions, on the other hand, can lead to a complete revamping of the native format (as was the case with WordPerfect 5.0). **WHAT** has been designed as far as possible to be able to handle new versions of formats that are already supported, but no guarantees are made. (I am fairly certain that **WHAT** will recognise documents created by version 6.5 of WordPerfect, but what happens with 9.0 documents is anybody's guess!)

Keeping abreast of all these changes and additions is no easy matter (I have yet to find a company that runs a mailing list for people interested in this kind of information!). What that means is that **WHAT** can

only be improved and kept up to date with the assistance of its users. So if you find that **WHAT** makes a mistake when analysing a supported format, experience trouble with the latest version of a particular program, or can provide information on file formats not currently supported by **WHAT**, please do not hesitate to get in touch. The more example files and technical information you can provide for a particular format the better. Your efforts will be rewarded with an acknowledgement in the next version of WHATDOC and a typeset copy of this one. (The "wish list" for the next version of **WHAT** includes support for CGM, CUT, DXF, GEM and PIC graphics, Quattro, PFS, Q&A, PageMaker, and the latest versions of DisplayWrite, Framework and Lotus 1-2-3; detailed information on Word for Windows and Excel – and whatever else you and I can get our hands on!)

Thanks to... Dag Hasvold, Aron Gurski, Gisle Hannemyr, Truls Meland, Tor Nordahl, Mike Robertson, Mats Tande and Chris Wolf for suggestions and help.

Send comments, files and format documentation to *Steve Pepper, Pilestredet 97, N-0358 Oslo, Norway* (email: pepper@falch.no), or log on to Computertext BBS (2400 8-N-1) +47-2-420825.

One final thing: *Don't* bother suggesting that the next version of **WHAT** ought to be able to recognise non-DOS disk formats unless you are prepared to tell me how to implement such a feature. I know it would be enormously useful, but I am a typographer, not a programmer!

Steve Pepper
Oslo, 19 April 1991

Appendix A

Text and data formats supported by WHAT?format v. 30.0

Here is a complete list of all formats supported by version 30.0 of **WHAT**. Those formats for which extra information is given (other than version number) are shown in bold type. Please support **WHAT** by helping to make this list more comprehensive!

Word processors

Ability WP
Acto WP
Amí Professional
ASCII text file (09,0A,0C,0D,1A and 20..7E)
ASCII even parity
Cicero
DisplayWrite
DOS text file (as ASCII, plus 80..FE)
DSI Tekst
EBCDIC file
Enable WPF
Framework
Manuscript
MASS-11
Microsoft Word
MicroWord
Multimate
Notis WP
OfficeWriter
Ordbehandling
Palantir
PC-Write
Samna Word
Sprint

	Super WP Symphony Ventura Publisher Volkswriter Windows Write Word for Windows WordPerfect WordStar WordStar 2000 XPress tagged ASCII XyWrite
Formatted text	**PostScript** – Structuring Conventions version DCA/RFT (DCA Revisable Form Text) DEC DX HP LaserJet (PCL) IBM DCF-GML (Generalised Markup Language) RTF (Microsoft Rich Text Format)
Data bases	Ability DataPerfect dBASE Enable Reflex
Spreadsheets	Ability DIF Enable Excel Lotus 1-2-3 PlanPerfect SuperCalc SYLK (Microsoft Symbolic Link)
Graphics	Ability Amí Metafile DrawPerfect

EPSF (Encapsulated PostScript)
GIF – resolution and number of colours
IFF – resolution for ILBM files
IMG – width and height
Lotus PIC
MacPaint
MSPaint – width and height
PCX – version, size and number of colours
TIFF – version and type (Motorola or Intel)
WPG – version and type (bitmap/drawing)

Various

Ability comms
ARC archive
DOS Code Page font
EXE file
LZH archive
MacBinary – TYPE and CREATOR
PostScript outline font
StuffIt! archive
Windows EXE file
Windows loadable font
ZIP archive
Miscellaneous file types from WordPerfect Corp.

Appendix B

Example batch files using the DOS errorlevel and %WHAT% variable

SORTPICS.BAT:

(using the DOS errorlevel)

```
@echo off
rem
rem SORTPICS.BAT
rem
rem Sort your pics using WHAT?format!
rem
rem Change to a directory containing an
rem assortment of graphics files and give
rem the command:
rem
rem    for %f in (*.*) do sortpics %f
rem
rem The files are copied to different
rem directories depending on their format
rem
if not exist %1 goto :end
what %1
if errorlevel 72 goto :end
if errorlevel 71 goto :TIFF
if errorlevel 70 goto :PCX
if errorlevel 67 goto :end
if errorlevel 66 goto :IMG
if errorlevel 65 goto :end
if errorlevel 64 goto :GIF
goto :end
:TIFF
copy %1 c:\graphics\tiff
del %1
goto :end
:PCX
copy %1 c:\graphics\pcx
del %1
goto :end
:IMG
copy %1 c:\graphics\img
del %1
goto :end
:GIF
copy %1 c:\graphics\gif
del %1
goto :end
:end
```

WPCONV.BAT:

(using the %WHAT% variable)

```
@echo off
rem
rem WPCONV.BAT
rem
rem Automate conversion using WHAT?format!
rem
rem Change to a directory containing
rem assorted WordPerfect files and give
rem the command:
rem
rem    for %f in (*.*) do wpconv %f
rem
rem The files are converted to DCA format
rem using the correct version of WP's
rem CONVERT.EXE.
rem
if not exist %1 goto :end
set what=what
what %1
if errorlevel 33 goto :notwp
if errorlevel 32 goto :wp
if errorlevel 1 goto :notwp
goto :end
:wp
if "%what%"=="WordPerfect 5.1" goto :wp51
if "%what%"=="WordPerfect 5.0" goto :wp50
if "%what%"=="WordPerfect 4.x" goto :wp4x
echo New version: %what%
goto :end
:wp51
convwp51 %1 d:\DCAstuff\%1 1 1 std.crs
goto :end
:wp50
convwp50 %1 d:\DCAstuff\%1 1 1
goto :end
:wp4x
convwp42 %1 d:\DCAstuff\%1 1 1
goto :end
:notwp
echo File is not WordPerfect format!
:end
set what=
```