

NAME

typerules – *HylaFAX* file type identification and conversion rules

DESCRIPTION

Only three types of files are accepted by the *HylaFAX* server for transmission as facsimile: POSTSCRIPT® files, PDF files, and TIFF Class F (bilevel Group 3-encoded) files. All other types of files must be converted to one of these three formats. The facsimile submission program applies a set of rules against the contents of each input file to identify the file’s type and to figure out how to convert the file to a format that is suitable for transmission. These rules are stored in the file `/usr/local/lib/fax/typerules`, an ASCII file that is patterned after the `/etc/magic` file used by the System V `file(1)` program. However, there are significant differences, noted below.

Type rules work by matching data patterns in a file; typically patterns that appear in the first few bytes of the file (i.e. magic numbers). There are two types of rules, *primary rules* and *secondary rules*. Secondary rules specify additional rules to apply *after* a primary rule has been matched. When secondary rules are used, rule scanning continues up to the next primary type rule in the file.

Each rule consists of a set of whitespace-separated fields:

offset datatype match result command

If an line is terminated with a backslash character, the entry is continued on the next line with any leading whitespace characters compressed to a single space. Comments are marked with the “#” character; everything from to the end of the line is discarded. Secondary rules have a “>” character in the first column of the line; primary rules do not.

The fields in each rule entry are:

offset The byte offset in the file at which data should be extracted and compared to a matching string or value.

datatype The type of data value to extract at the specified offset for comparison purposes; one of: “byte” (8 bit unsigned number), “short” (16 bit unsigned number), “long” (32 bit unsigned number), “string” (an array of bytes), “istring” (a case-insensitive array of bytes), or “ascii” (an array of ASCII-only bytes).

match The value and operation to use in matching; the value used is based on the datatype field. If value is “x”, then it is interpreted to mean *match anything*; otherwise the following operators are supported (where *data* is the value extracted from the file and *value* is specified in the match field) except for types “string”, “istring”, and “ascii”:

=	data == value	!=	data != value
>	data > value	<	data < value
<=	data <= value	>=	data >= value
&	(data & value) == value	!	(data & value) != value
^	(data ^ value) != 0		

If no operation is specified then “=” is used.

For “string”, “istring”, and “ascii” no operator is allowed; the implicit operation is always “=”. In these cases, the field is terminated by a tab or end of line, not by “#”. Characters in the field have their literal value; there are no C-style character escapes.

result One of “ps”, “tiff”, or “error” (case insensitive). The first two results specify whether the rule generates a POSTSCRIPT file or a TIFF/F file (Group 3-encoded bilevel data), respectively. The “error” result indicates that a file is unsuitable for transmission and, if supplied for transmission, should cause the job to be aborted with the *command* field used in an error message.

command A command description that is expanded and passed to the shell to convert the input file to the result format (suitable for sending as facsimile). Before the string is passed to the shell, it is scanned and the following “%” escape codes are substituted for:

- %i input file name
- %o output file name
- %r output horizontal resolution in pixels/mm
- %R output horizontal resolution in pixels/inch
- %v output vertical resolution in lines/mm
- %V output vertical resolution in lines/inch
- %f data format, “1” for 1-d encoding or “2” for 2-d encoding
- %w page width in pixels
- %W page width in mm
- %l page length in pixels
- %L page length in mm
- %s page size by name
- %F the directory where *HylaFAX* filter programs reside
- %<x> the <x> character (e.g. “%%” results in “%”)

See below for example uses of these codes.

EXAMPLES

The following rules are used to match the formats that are handled directly by the server:

#offset	datatype	match	result	command
0	string	%!	ps	# POSTSCRIPT
0	string	%PDF	ps	# POSTSCRIPT by Ghostscript
0	short	0x4d4d	tiff	# big-endian TIFF
0	short	0x4949	tiff	# little-endian TIFF

These rules are used to process the ASCII version of IRIS Inventor database files while blocking the transmission of the binary format variant:

#offset	datatype	match	result	command
0	string	#Inventor V	error	IRIS Inventor file
>15	string	binary	error	binary IRIS Inventor file
>15	string	ascii	ps	%F/textfmt -fCourier-Bold -p11bp\ -U -q >%o <%i

This rule is typically the last entry in the file and is used to convert all unmatched ASCII data files to POSTSCRIPT:

#offset	datatype	match	result	command
0	ascii	x	ps	%F/textfmt -fCourier-Bold -p11bp -U -q >%o <%i

NOTES

It is much better to convert data that is to be transmitted to POSTSCRIPT because this data format permits the facsimile server to do the final imaging according to the optimal transfer parameters (resolution, binary encoding, etc.).

It might be better to allow secondary rules to augment a primary rule rather than just replace them. This would allow, for example, command line options to be selected based on file type.

SEE ALSO

sendfax(1), hylafax-client(1)