NAME

choptest − *HylaFAX* page chopping test program

SYNOPSIS

/usr/sbin/choptest [*options*] input.tif

DESCRIPTION

choptest is a program for testing the page chopping support in the *HylaFAX* software (specifically, in the *faxq*(8C) program). *choptest* analyzes a TIFF/F (TIFF Class F) file with 1-D MR- or 2-D MR-encoded data and reports what the automatic page chopping logic would do if the file were submitted for transmission. Options are provided for controlling whether *choptest* checks the last page or all pages of the document, and what whitespace threshold to use in deciding if a page should be chopped.

OPTIONS

−a Chop all pages in the document. This is equivalent to setting the **PageChop** configuration parameter to ‘‘all’’; c.f. *hylafax-config*(5F). By default only the last page of the document is considered.

−t *inches* Set the minimum whitespace threshold that must be present on a page for it to be chopped. This is equivalent to the **PageChopThreshold** configuration parameter; c.f. *hylafax-config*(5F). By default *choptest* requires that at least 3 inches of trailing whitespace be present.

EXAMPLES

The following shows a multi-page, high-resolution document. Each page has insufficient whitespace for it to be chopped.

```
hyla% ./choptest -a ~/tiff/pics/faxix.tif
Chop pages with >=3" of white space at the bottom.
Don't chop, found 67 rows, need 588 rows
Don't chop, found 67 rows, need 588 rows
Don't chop, found 67 rows, need 588 rows
Don't chop, found 53 rows, need 588 rows
Don't chop, found 91 rows, need 588 rows
Don't chop, found 99 rows, need 588 rows
Don't chop, found 47 rows, need 588 rows
```

SEE ALSO

faxq(8C), *hylafax-config*(5F)

NAME

cid − Caller-ID access control list

DESCRIPTION

The *HylaFAX* configuration parameter **QualifyCID** specifies whether or not the identity of an inbound call should be checked against an access control list before the telephone is answered. If a modem is attached to a phone line that has Caller-ID service, and **QualifyCID** is non-null, then only the callers identified in the file specified by the string (typically **etc/cid**) will be answered.

Patterns are specified one per line and must conform to the regular expressions syntax specified by POSIX 1003.2; see *re_format*(7). Comments may be included; they are introduced with the "#" character and extend to the end of the line. Any trailing white space on a line is ignored (for convenience when comments are used).

If a line begins with "!", then the regular expression identifies callers that should be *rejected*; otherwise regular expressions identify clients whose calls should be *accepted*. The order of patterns in a CID file is important. When a call is to be answered, the *faxgetty* process will compare the phone number presented by the modem against the patterns in the access control list in the order in which they appear in the file. The first pattern that matches the client's number is used to decide whether to accept or reject the call. If no patterns match the phone number then the call is not answered. Thus if you want to accept all but a restricted set of calls, the last line in the file should be "^.*$".

Note that regular expression patterns should be written to match a phone number exactly. That is, patterns should be of the form:

```
^<pattern>$
```

where the "^" and "$" characters are used to specify the start and end of the matching phone number. Additionally, regular expression patterns should handle white space that may appear in known locations. For example,

```
^([+]1){1}[ .−]*415[ .−]*555[ .−]*1212.*$
```

matches the following phone number strings:

```
+1.415.555.1212
    415  555  1212
1−415−555−1212
```

Finally, note that regular expressions can be used to specify many numbers with one pattern.

NOTES

The caller's telephone number is recognized by using the **CIDNumber** configuration parameter described in *hylafax-config*(5F). This parameter must be present in the per-modem configuration file for Caller-ID screening to function properly. If **QualifyCID** is defined, but no **CIDNumber** is specified, then no phone numbers will be recognized and all incoming calls will be ignored.

Caller-ID data is only sent once by the telco, and the timing of its arrival varies upon the telco and the type of line. If *HylaFAX* is configured to answer via **RingsBeforeAnswer** prior to the arrival of the caller-ID data, then all **CIDNumber** and **CIDName** values will be null. For example, on US analog lines, caller-ID data is usually sent between RINGs 1 and 2. In this situation, **RingsBeforeAnswer** should be set to "2" or greater.

SEE ALSO

faxgetty(8C), *hylafax-config*(5F), *re_format*(7).

NAME

cqtest − *HylaFAX* copy quality checking test program

SYNOPSIS

/usr/sbin/cqtest [*options*] input.tif

DESCRIPTION

cqtest is a program for testing the copy quality checking support in the *HylaFAX* software (specifically, in the *faxgetty*(8C) program). *cqtest* takes a TIFF/F (TIFF Class F) file and generates a new TIFF/F file that is a copy of the input file, but with any erroneous scanlines replaced/regenerated. In addition, *cqtest* prints diagnostic messages describing its actions and indicates whether the input data has acceptable copy quality according to the copy quality checking threshold parameters. Options are provided for specifying copy quality checking threshold parameters

OPTIONS

−m *badlines* Set the maximum consecutive bad lines of data that may appear in each acceptable page of input data. This is equivalent to the **MaxConsecutiveBadLines** configuration parameter; c.f. *hylafax-config*(5F). By default *cqtest* accepts no more than 5 consecutive bad lines in a page.

−o *file* Write output to *file*. By default output is written to the file **cq.tif**.

−p *%goodlines* Set the minimum percentage of ‘‘good lines’’ of data that may appear in acceptable page of input data. A line is good if it decodes without error to a row of pixels that is the expected width. This is equivalent to the **PercentGoodLines** configuration parameter; c.f. *hylafax-config*(5F). By default *cqtest* requires that 95% of the rows of each page be good.

EXAMPLES

The following shows a multi-page, high-resolution document with a single error on each page. Each page has acceptable copy quality using the default threshold parameters.

```
hyla% /usr/sbin/cqtest ˜/tiff/pics/faxix.tif
1728 x 297, 7.7 line/mm, 1-D MR, lsb-to-msb
RECV/CQ: Bad 1D pixel count, row 245, got 1616, expected 1728
RECV: 2234 total lines, 1 bad lines, 1 consecutive bad lines
1728 x 297, 7.7 line/mm, 1-D MR, lsb-to-msb
RECV/CQ: Bad 1D pixel count, row 148, got 3023, expected 1728
RECV: 2234 total lines, 1 bad lines, 1 consecutive bad lines
1728 x 297, 7.7 line/mm, 1-D MR, lsb-to-msb
RECV/CQ: Bad 1D pixel count, row 151, got 1722, expected 1728
RECV: 2234 total lines, 1 bad lines, 1 consecutive bad lines
1728 x 297, 7.7 line/mm, 1-D MR, lsb-to-msb
RECV/CQ: Bad 1D pixel count, row 148, got 1776, expected 1728
RECV: 2234 total lines, 1 bad lines, 1 consecutive bad lines
```

SEE ALSO

faxgetty(8C), *hylafax-config*(5F)

NAME

destctrls − per-destination controls for *HylaFAX* servers

DESCRIPTION

The *HylaFAX* configuration parameter **DestControls** specifies whether or not to apply per-destination controls on job characteristics such as the time-of-day to place a call.

The controls file is a simple ASCII file containing a series of records of the form:

dest param = value ... param = value

where *dest* is a regular expression matched against the canonical form of a destination phone number and parameters are specified by named assignments (though the "=" can also be left out). Whitespace may be used freely though it is used to identify the end of *dest* and the start of the parameter list. Parameter values are parsed exactly as specified in *hylafax-config*(5F); i.e. values with embedded whitespace may be enclosed in quote marks ("""").

Comments are introduced with a "#" character and extend to the end of the line. Parameter lists may be continued across lines by starting subsequent lines with a whitespace character (lines commented out with "#" are also considered to be part of a continued record).

The order of entries in a file is important. When a parameter value is needed *faxq* sequentially scans the file until it finds a record that matches the canonical form of the destination phone number. If the requested parameter is defined in this record, then the value is returned, otherwise a default value is used from the *faxq* configuration file.

The following parameters may be specified on a per-destination basis; consult *hylafax-config*(5F) for a description of each parameter (except **RejectNotice** which is described below).

Tag	**Type**	**Default**	**Description**
MaxConcurrentJobs	integer	1	max concurrent jobs to process for a destination
MaxDials	integer	*unlimited*	max phone calls to make to transmit a job
MaxSendPages	integer	*unlimited*	max pages to permit in a send
MaxTries	integer	3	max attempts to transmit a job
Modem	string	Any	ModemGroup to use for destination
RejectNotice	string	−	rejection notice for transmit job
TimeOfDay	string	Any	default time-of-day restrictions

The **Modem** parameter controls which **ModemGroup** is used in sending faxes to the destination. If the user assigns a **ModemGroup** for a specific job where all modems are outside of this **ModemGroup** matching a destination, then the value of **Modem** is overridden by the user's assignment. Likewise, if the user assigns a **ModemGroup** for a specific job where some, but not all, of the modems are included in this **ModemGroup** matching a destination, then the value of **Modem** is overridden by the inclusive set of modems found in both.

The **RejectNotice** parameter controls whether or not to reject jobs to the destination. Jobs that are rejected are done so without placing a phone call and the associated message is returned to the job submitter. This facility can be used to disallow calling sensitive phone numbers; for example

```
.*911$    RejectNotice = "Calls to emergency numbers are not permitted"
```

In addition to the above parameters, any other parameters that are specified are automatically accumulated and passed to programs invoked by *faxq*, such as *faxsend* and *pagesend*. This is a convenient mechanism for defining configuration parameters for all modems without having to modify each modem-specific configuration file. For example,

```
.*    SessionTracing = 0x4f
```

This mechanism also makes it easy to control transmit-related parameters according to the destination phone number. For example, to disable use of ECM and restrict the transmit speed when placing international phone calls one might use:

```
[+][^1]    DesiredBR=9600 DesiredEC=false
```

NOTES

Beware that canonical phone numbers have a leading “+” which is a special character for regular expressions. An easy way to match “+” is with “[+]”; e.g. “[+]14159657824”.

SEE ALSO

faxq(8C), *hylafax-config*(5F), *re_format*(7).

NAME

dialrules − *HylaFAX* dial string processing rules

DESCRIPTION

A dial string specifies how to dial the telephone in order to reach a destination facsimile machine, or similar device. This string is supplied by a user with each outgoing facsimile job. User-supplied dial strings need to be processed in two ways by the *HylaFAX* server processes: to craft a canonical phone number for use in locating the receiver's capabilities, and to process into a form suitable for sending to a modem. In addition client applications may need to process a dial string to formulate an external form that does not include private information such as a credit card access code. Phone number canonicalization and dial string preparation are done according to *dial string processing rules* that are located in a file specified in the server configuration file; see the **DialStringRules** parameter in *hylafax-config*(5F). The generation of an externalized form for a dial string is done by rules that optionally appear in */usr/share/fax/dialrules* on client machines.

A dial string rules file is an ASCII file containing one or more *rule sets*. A rule set defines a set of *transformation rules* that are sequentially applied to a dial string. Each rule set is associated with an identifier, with certain well-known identifiers being used by the facsimile server or client application. Each transformation rule is a regular expression and a replacement string; the regular expression is repeatedly applied to a dial string and any matching substring is replaced by the replacement string.

The syntax of a dial string rules file is as follows. Comments are introduced with the ''!'' character and continue to the end of the current line. Identifiers are formed from a leading alphabetic and any number of subsequent alpha-numeric characters. A rule set is of the form:

> *Identifier* **:= [**
>   *rule1*
>   *rule2*
>   ...
> **]**

where *rule1*, *rule2*, and so on are transformation rules. Line breaks are significant. The initial rule set definition line and the trailing '']'' must be on separate lines; and each transformation rule must also be on a single line. Transformation rules are of the form:

> *regular-expression* **=** *replacement*

where *regular-expression* is a POSIX 1003.2 extended regular expression and *replacement* is a string that is substituted in place of any portion of the dial string that is matched by the *regular-expression*. White space is significant in parsing transformation rules. If a regular expression or replacement string has embedded white space in it, then the white space needs to be escaped with a ''\'' character or the entire string should be enclosed in quote (''"'') marks. Replacement strings may reference the entire string matched by the regular expression with the ''&'' character. Substrings matched with the ''(...)'' constructs may be referenced by using ''*n*'' where *n* is a single numeric digit between 1 and 9 that refers to the *n*-th matched substring; c.f. *re_format*(7), *sed*(1), etc.

To simplify and parameterize the construction of rule sets, dial string rules files may also include simple text-oriented variable definitions. A line of the form:

> *foo***=***string*

defines a variable named *foo* that has the value *string*. String values with embedded whitespace must use the ''\'' character or be enclosed in quote marks. Variables are interpolated into transformation rules by referencing them as:

> **${***var***}**

Note that variable interpolation is done only once, at the time a transformation rule is defined. This means that forward references are not supported and that circular definitions will not cause loops. The facsimile server automatically defines four variables to have the values defined in its configuration file: **AreaCode**, **CountryCode**, **LongDistancePrefix**, and **InternationalPrefix** These variables are initialized before parsing a dial string rules file; thus if they are defined in the rules file then they will override any definition by the server.

There are three well known rule set names: *CanonicalNumber* to convert a dial string to a canonical format, *DialString* to prepare a dial string before using it to dial the telephone, and *DisplayNumber* to convert a dial string to an external ‘‘displayable’’ form that does not include the private information that might appear in the raw dial string.

EXAMPLES

This is the default set of rules for transforming a dial string into a canonical phone number:

```
Area=${AreaCode}                    ! local area code
Country=${CountryCode}              ! local country code
IDPrefix=${InternationalPrefix}     ! prefix for placing an international call
LDPrefix=${LongDistancePrefix}      ! prefix for placing a long distance call
!
! Convert a phone number to a canonical format:
!
!    +<country><areacode><rest>
!
! by (possibly) stripping off leading dialing prefixes for
! long distance and/or international dialing.
!
CanonicalNumber := [
#.*             =                           ! strip calling card stuff
[abcABC]        = 2                         ! these convert alpha to numbers
[defDEF]        = 3
[ghiGHI]        = 4
[jklJKL]        = 5
[mnoMNO]        = 6
[prsPRS]        = 7
[tuvTUV]        = 8
[wxyWXY]        = 9
[^+0-9]+        =                           ! strip white space etc.
^${IDPrefix}    = +                         ! replace int. dialing code
^${LDPrefix}    = +${Country}               ! replace l.d. dialing code
^[^+]           = +${Country}${Area}&       ! otherwise, insert canon form
]
```

The first rule simply strips anything following a ‘‘#’’; this will remove any calling card-related information. The next eight rules convert upper and lower case alphabetics to the equivalent key numbers (this is convenient for users that use mnemonic phone numbers). The tenth rule removes everything but numbers and plus signs. The eleventh rule translates any explicit international dialing prefix into the ‘‘+’’ symbol used to identify country codes. The twelfth rule replaces a leading long distance dialing prefix with the local country code string. The last rule matches local phone numbers and inserts the local country code and area code.

As an example, assume that

```
AreaCode=415
CountryCode=1
InternationalPrefix=011
LongDistancePrefix=1
```

then if the above set of rules is applied to ‘‘01123965-Tube#2345’’, the transformations would be:

```
01123965-Tube#2345   01123965-Tube   ! strip calling card stuff
01123965-Tube        01123965-8823   ! convert alphabetics
01123965-8823        011239658823    ! strip white space etc.
011239658823         +239658823      ! replace int. dialing code
```

```
+239658823        +239658823      ! replace l.d. dialing code
+239658823        +239658823      ! otherwise, insert canon form
```

for a final result of ‘‘+239658823’’.

SEE ALSO

sendfax(1), *dialtest*(8C), *faxq*(8C), *faxsend*(8C), *faxgetty*(8C), *hylafax-config*(5F)

NAME

dialtest – *HylaFAX* dial string processing rules test program

SYNOPSIS

/usr/sbin/dialtest [*options*] *dialrules*

DESCRIPTION

dialtest is an interactive program for the testing and development of dial string processing rules used by *HylaFAX*. *dialtest* reads the file of rules specified on the command line and then prompts for a dial string. For each string typed in, *dialtest* prints the result of applying the canonicalization and dial-string preparation rule sets. A rule set whose name is *setname* can be specified by typing ‘‘*setname*(*string*)’’; for example, ‘‘*CanonicalName*(+1.415.965.7824)’’.

Note that *dialtest* reads the rules file only when it is initially started up. This means that *dialtest* must be restarted for it to apply changes to a rules file.

OPTIONS

–a *code* Use *code* as the value of **AreaCode** in the rules file. The default area code is ‘‘415’’.

–c *code* Use *code* as the value of **CountryCode** in the rules file. The default country code is ‘‘1’’.

–i *prefix* Use *prefix* as the value of **InternationalPrefix** in the rules file. The default international dialing prefix is ‘‘011’’.

–l *prefix* Use *prefix* as the value of **LongDistancePrefix** in the rules file. The default long distance dialing prefix is ‘‘1’’.

–v Print each input line before displaying the results of applying the dial string rules to the input string. This can be useful, for example, in shell scripts.

SEE ALSO

dialrules(5F)

NAME

doneq/q* − *HylaFAX* outbound job description

DESCRIPTION

Files in the **doneq** directory specify transmission job requests. These files are created by *hfaxd*(8C), when submitting a job on behalf of *sendfax*(1) or *sendpage*(1). Job description files also reside in the **sendq** directory; they are moved from there to the doneq directory by *faxq*(8C) when a job completes.

Job description files are ASCII files with lines of the form

tag: *value*

where a *tag* is one of the identifiers described below and a *value* is either a string or number. Values start at the first non-blank character after the ‘‘:’’ and continue to the end of the line. Lines that end with a backslash character ‘‘\’’ are continued to the next line. Numeric values are decimal integers, except for **chophreshold** which is expressed as a floating point value.

The following table lists the possible tags and the required type of the associated value.

Tag	**Type**	**Description**
chopthreshold	float	page chopping threshold (inches)
commid	string	communication identifier for last call
company	string	receiver’s company
client	string	host that submitted the job
cover	string	continuation coverpage file
data	string	unprocessed document of indeterminate type
!data	string	processed document of indeterminate type
dataformat	string	data format used to transmit a facsimile
desiredbr	integer	desired transmit speed
desireddf	integer	desired page data format
desiredec	integer	desired use of Error Correction Mode (ECM)
desiredst	integer	desired minimum scanline time
desiredtl	integer	whether or not to use **tagline** parameter
donop	string	operation to perform when job is reaped
external	string	external form of dialstring
fax	string	document ready for transmission
groupid	integer	*HylaFAX* job group identifier
jobid	integer	*HylaFAX* job identifier
jobtag	string	user-specified job identifier
jobtype	string	job type identification string
killtime	integer	time to give up trying to send job
location	string	receiver’s location
mailaddr	string	email address of sender
maxdials	integer	max number of times to dial
maxtries	integer	max number of attempts to send job
minsp	integer	minimum required transmit speed
modem	string	outgoing modem to use
ndials	integer	number of consecutive failed attempts to place call
notify	string	email notification specification
npages	integer	number of pages transmitted
ntries	integer	number of attempts to send current page
number	string	dialstring to use in placing call
owner	string	user that owns the job
page	string	PIN in a page operation
pagechop	string	whitespace truncation handling
pagehandling	string	page analysis information
pagelength	integer	facsimile page length (millimeters)
pagewidth	integer	facsimile page width (millimeters)

passwd	string	info to send in HDLC PWD frame
pcl	string	unprocessed PCL document
!pcl	string	processed PCL document
poll	string	polling request
postscript	string	unprocessed POSTSCRIPT document
!postscript	string	processed POSTSCRIPT document
priority	integer	user-specified scheduling priority
receiver	string	receiver's identity
resolution	integer	vertical resolution of facsimile
retrytime	integer	time to use between job retries
schedpri	integer	current scheduling priority
sender	string	identity of sender
signalrate	string	signalling rate at which a facsimile was sent
state	integer	job scheduling state
status	string	job status
subaddr	string	info to send in HDLC SUB frame
tagline	string	tagline format string
tiff	string	unprocessed TIFF/F document
!tiff	string	processed TIFF/F document
totdials	integer	total number of phone calls
totpages	integer	total pages to transmit
tottries	integer	total number of attempts to send job
tts	integer	time to send job
useccover	integer	whether or not to use a continuation cover page

PARAMETERS

Note that all files must be owned by the fax user. Pathnames for document files must be relative to the top of the *HylaFAX* spooling directory and not include ''..''.

chopthreshold The minimum trailing whitespace (in inches) that must appear on a page for it to be considered for ''page chopping''. If this parameter is not specified then the value specified by the **PageChopThreshold** configuration parameter is used; see *hylafax-config*(5F).

commid The communication identifier for the last outbound call placed for this job. This string is of the form ''XXXXXXXX'' where XXXXXXXX is a decimal sequence number in the range 1-99999999.

client The fully qualified hostname for the machine from which the job was submitted. If the hostname is unknown at the time the job is submitted, then the host address in Internet standard ''.'' (dot) notation is supplied instead.

company The receiver's company name as specified with the **−x** option to *sendfax*(1). This value is used only when creating continuation cover pages.

cover The pathname of a *continuation cover page* to send. The file must be TIFF/F (i.e. it must already be prepared for transmission). This document will be transmitted before any other documents listed in the file. See **ContCoverPage** and **ContCoverCmd** in *hylafax-config*(5F).

data The pathname of a document to send. Once a document has been processed, it is marked as **!data** so that subsequent retries do not reprocess the document.

dataformat The format used to transmit facsimile data. This item is recorded by *faxsend*(8C) so that *notify*(8C) can return it in a notification message.

desiredbr The desired signalling rate to use for page data transmissions: 0 for 2400 bps, 1 for 4800 bps, 2 for 7200 bps, 3 for 9600 bps, 4 for 12000 bps, and 5 for 14400 bps (default).

desireddf The desired data format to use for page data transmissions: 0 for 1-D MR, 1 for 2-D MR, 3 for 2-D MR with uncompressed data, and 4 for 2-D MMR (default).

desiredec Whether or not to negotiate the use of Error Correction Mode (ECM): 0 to disable, 1 to enable (default).

desiredst The desired minimum scanline time to use for page data transmissions: 0 for 0 ms (default), 1 for 5 ms, 2 for 10ms/5ms, 3 for 10ms, 4 for 20ms/10ms, 5 for 20ms, 6 for 40ms/20ms, and 7 for 40ms. This value is used in selecting the negotiated session parameters.

desiredtl Whether or not to use the format string specified by the **tagline** job parameter when imaging taglines on document pages. If this value is non-zero then per-job format string is used; otherwise the tagline format string specified in the server configuration file is used; see **TagLineFormat** in *hylafax-config*(5F).

doneop The treatment the job should be given when it is processed by the *faxqclean*(8C) job cleaner program. The may be one of *remove* (to expunge the job description file and all associated documents that are not referenced by another job) or *archive* (to request that the job and associated documents be archived on the server machine). Consult *faxqclean* more details on other information that may be included in this string.

external The external form of the dialstring. This string is used instead of the dialstring in any status and log messages. Clients can protect private information such as access codes that must be placed in dialstrings by supplying a ‘‘cleansed’’ version of the dialstring.

fax The pathname of a TIFF/F document that is ready for transmission. If the pathname string has a leading ‘‘xx:’’, then ‘‘xx’’ is interpreted to be the index of the first directory in the file to transmit (directories are numbered starting at zero).

groupid The job group identifier, an integer number in the range 1 to 32000 assigned to each group of jobs by the *HylaFAX* software. Each job in a group of jobs submitted at the same time is assigned the same value by the *HylaFAX* software.

jobid The job identifier, a unique integer number in the range 1 to 32000 assigned to each job by the *HylaFAX* software.

jobtag A user-specified string that is intended for identification purposes. If this string is defined, it is returned to the user in notification messages instead of the job identifier.

killtime The time to kill this job, expressed in seconds since 00:00:00 GMT, January 1, 1970.

location The receiver’s location as specified with the **−y** option to *sendfax*(1). This value is used only when creating continuation cover pages.

mailaddr The electronic mail address to which notification messages should be directed.

maxdials The maximum number of times the server should dial the phone. If the job cannot be completed in this number of calls, it is terminated.

maxtries The maximum number of times the server should attempt to send a job; where an attempt is defined to be a session that makes it past Phase A of the CCITT T.30 protocol (for facsimile), or similar. If the job cannot be completed in this number of tries, it is terminated.

minsp The minimum signalling rate to use for transmitting page data: 0 for 2400 bps (default), 1 for 4800 bps, 2 for 7200 bps, 3 for 9600 bps, 4 for 12000 bps, and 5 for 14400 bps.

modem The outgoing modem to use to send the job, or ‘‘any’’ if any available modem is acceptable.

owner The identity of the user that owns the job. Typically the owner is the user that created the job, though clients with administrative privileges can change the ownership of jobs. The value is the string specified in the USER command passed to *hfaxd*(8C) when a client logins in to the server.

notify A specification of whether the sender wants notification: whenever the job is requeued after a failed attempt to transmit (‘‘*when requeued*’’), when the job is completed (‘‘*when done*’’), either of the above (‘‘*when done+requeued*’’), or not at all (‘‘*none*’’).

npages The number of pages in the facsimile, including the cover page, that have been successfully transmitted.

ntries The number of unsucessful attempts that have been made to transmit the current page.

ndials The number of consecutive unsucessful attempts to establish carrier with the the remote device.

number The dialstring to use to call the destination device.

page a string with PIN and to use in processing a page transmit request.

pagechop Whether or not to automatically truncate trailing whitespace on pages of outbound facsimile; see **PageChop** in *hylafax-config*(5F). The possible values are: ''*default*'' to use the setting of the **PageChop** configuration parameter, ''*none*'' to disable page chopping, ''*all*'' to enable chopping of all pages, and ''*last*'' to enable chopping of the last page of each document.

pagehandling A string that contains per-page information used during facsimile transmission. Most of the information is used in the negotiation of session capabilities with optional information used in the implementation of page chopping. The string has three characters, say *xxc*, for each page to be transmitted. The *xx* characters are a hex-encoded specification of the session capabilities required to transmit the page. The *c* character is one of ''M'', ''S'', and ''P''; where: ''M'' indicates there is another page to follow and that the page requires different T.30 session parameters be negotiated (an EOM code is to be transmitted). ''S'' indicates there is another page to follow, but the page uses the same session parameters (an MPS. code is to be transmitted). ''P'' indicates this is the last page to be transmitted (an EOP code is to be transmitted). In addition, if page chopping has been enabled and the page has enough trailing whitespace to be a candidate, then each three character string will be followed by a string of the form ''*ZXXXX*'', where ''XXXX'' is a 4 digit hex-encoded count of the number of bytes of data to send for a chopped page.

pagelength The desired page length in millimeters.

pagewidth The desired page width in millimeters.

pcl The pathname of a PCL document to send. Once a PCL document has been processed by *faxsend*, it is marked as **!pcl** so that subsequent retries do not reprocess the document.

poll The value is a Caller Identification String (CIS) to use in making a polling request. If no CIS is specified then *faxsend* will form one from the **FAXNumber** configuration parameter.

postscript The pathname of a POSTSCRIPT® document to send. Once a POSTSCRIPT document has been processed by *faxsend*, it is marked as **!postscript** so that subsequent retries do not reprocess the document.

receiver The receiver's name. This value is used only when creating continuation cover pages.

resolution The desired vertical resolution in lines per inch. This value should be either 98 or 196.

retrytime The time, in seconds, to delay between each attempt to transmit a job that fails because of a communication-oriented problem. Normally jobs are scheduled according to a collection of configuration parameters that are tuned for the specific reason of each failure; e.g. see **JobReqProto** in *hylafax-config*(5F).

sender The ''full name'' of the sender of the facsimile.

signalrate The signalling rate used to transmit facsimile data. This item is recorded by *faxsend*(8C) so that *notify*(8C) can return it in a notification message.

state The job scheduling state. Recognized values are: 1 (*suspended*, not being scheduled), 2 (*pending*, waiting for the time to send), 3 (*sleeping*, waiting for a scheduled timeout), 4 (*blocked*, waiting for concurrent activity to the same destination to complete), 5 (*ready*, ready to be processed except for available resources), 6 (*active*, actively being processed by *HylaFAX*), 7 (*done*, processing completed).

status Status information about the current state of the job. Typically this is the reason why the last attempt to send the facsimile failed.

subaddr The SubAddress information to transmit when sending a facsimile. This string should be used by the receiver to route received facsimile documents.

tagline The format string to use in imaging taglines on outgoing documents. Note that this parameter is only used if the **desiredtl** parameter is non-zero.

tiff The pathname of a TIFF/F document to send. Once a TIFF document has been processed by *faxsend*, it is marked as **!tiff** so that subsequent retries do not reprocess the document.

totdials The total number of times the server has placed a call to the receiving phone number.

totpages The total number of pages to be transmitted.

tottries The total number of attempts to transmit a job; see **maxtries** above.

tts The time to send the job. If this value is non-zero, it is number of seconds since 00:00:00 GMT, January 1, 1970. Otherwise, a value of 0 is interpreted to mean process as soon as possible.

useccover 1 if the job should use a continuation cover page during retransmission, 0 otherwise. See *hylafax-server*(5F) for a description of continuation cover pages.

SEE ALSO

sendfax(1), *faxq*(8C), *faxqclean*(8C), *faxsend*(8C), *pagesend*(8C), *hfaxd*(8C), *hylafax-config*(5F)

NAME

fax2ps − convert a TIFF facsimile to compressed POSTSCRIPT™

SYNOPSIS

fax2ps [*options*] [*file*...]

DESCRIPTION

fax2ps reads one or more TIFF facsimile image files and prints a compressed form of POSTSCRIPT on the standard output that is suitable for printing.

By default, each page is scaled to reflect the image dimensions and resolutions stored in the file. The **−x** and **−y** options can be used to specify the horizontal and vertical image resolutions (lines/inch), respectively. If the **−S** option is specified, each page is scaled to fill an output page. The default output page is 8.5 by 11 inches. Alternate page dimensions can be specified in inches with the **−W** and **−H** options.

By default *fax2ps* generates POSTSCRIPT for all pages in the file. The **−p** option can be used to select one or more pages from a multi-page document.

fax2ps generates a compressed form of POSTSCRIPT that is optimized for sending pages of text to a POSTSCRIPT printer attached to a host through a low-speed link (such as a serial line). Each output page is filled with white and then only the black areas are drawn. The POSTSCRIPT specification of the black drawing operations is optimized by using a special font that encodes the move-draw operations required to fill the black regions on the page. This compression scheme typically results in a substantially reduced POSTSCRIPT description, relative to the straightforward imaging of the page with a POSTSCRIPT *image* operator. This algorithm can, however, be ineffective for continuous-tone and white-on-black images. For these images, it sometimes is more efficient to send the raster bitmap image directly; see *tiff2ps*(1).

OPTIONS

−p *number* Print only the indicated page. Multiple pages may be printed by specifying this option more than once.

−x *resolution*
Use *resolution* as the horizontal resolution, in dots/inch, of the image data. By default this value is taken from the file.

−y *resolution*
Use *resolution* as the vertical resolution, in lines/inch, of the image data. By default this value is taken from the file.

−S Scale each page of image data to fill the output page dimensions. By default images are presented according to the dimension information recorded in the TIFF file.

−W *width* Use *width* as the width, in inches, of the output page. The default page width is 8.5 inches.

−H *height* Use *height* as the height, in inches, of the output page. The default page height is 11 inches.

DIAGNOSTICS

Some messages about malformed TIFF images come from the TIFF library.

Various messages about badly formatted facsimile images may be generated due to transmission errors in received facsimile. *fax2ps* attempts to recover from such data errors by resynchronizing decoding at the end of the current scanline. This can result in long horizontal black lines in the resultant POSTSCRIPT image.

NOTES

If the destination printer supports POSTSCRIPT Level II then it is always faster to just send the encoded bitmap generated by the *tiff2ps*(1) program.

BUGS

fax2ps should probably figure out when it is doing a poor job of compressing the output and just generate POSTSCRIPT to image the bitmap raster instead.

SEE ALSO

tiff2ps(1), *libtiff*(3)

NAME

faxabort − tell a *HylaFAX* server to abort a receive

SYNOPSIS

/usr/sbin/faxabort [**−q** *dir*] *modem*

DESCRIPTION

faxabort sends a message to a *HylaFAX faxgetty*(8C) process servicing *modem* telling it to abort any active receive operation.

The specified *modem* can either be the name (typically the last component of the terminal port the modem is attached to), or the full name of the associated FIFO named pipe file, e.g. ‘‘FIFO.ttym2’’.

Note that to abort a send operation the *faxrm*(1) command must be used.

OPTIONS

−q *dir* Use a spooling area other than */var/spool/hylafax*.

NOTES

On some systems it is not possible to abort a received facsimile because of problems in the operating system; on such systems the message sent by *faxabort* will be (silently) ignored.

FILES

/var/spool/hylafax	default spooling area
/var/spool/hylafax/FIFO	fifo for contacting *faxq*

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

hylafax-server(5F), *faxq*(8C), *faxgetty*(8C)

NAME

faxaddmodem − configure a modem for use with *HylaFAX*

SYNOPSIS

/usr/sbin/faxaddmodem [**−s** *speed*] [*tty*]

DESCRIPTION

faxaddmodem is an interactive shell script that does the necessary setup work to configure a modem for use by the *HylaFAX* software. To enable use of the configured modem one must either use the *faxmodem*(8C) program or configure the *init*(8C) program to startup a *faxgetty*(8C) process for the newly configured modem. In either case the *faxq*(8C) program must also be running if outbound services are to be provided.

If the serial port name is not specified on the command line *faxaddmodem* will prompt for it.

The configuration work done by *faxaddmodem* falls into two main areas: creating the necessary files to run a facsimile server process, and defining the contents of the configuration database file. The *hylafax-config (5F)* manual page has an in-depth description of the contents of the configuration file and should be consulted when installing a modem.

Many aspects of configuring a modem are operating system-specific; *faxaddmodem* uses a collection of support routines and parameters that are generated by the *faxsetup*(8C) program. If *faxsetup* has not been run on a machine prior to running then *faxaddmodem* will abort.

faxaddmodem can be used to install a new modem or re-configure an existing modem. In the latter case it will propagate all the server-related configuration parameters from the existing configuration file to the new configuration file. These parameter include items such as the phone number, area code, and tracing control parameters. Modem-related configuration parameters are, however, not propagated. Instead *faxaddmodem* deduces the type of modem attached to the specified serial port and then sets up the configuration parameters using a set of *prototype* configuration files that have been tested with like modems. If a modem is to be installed for which there is no existing prototype file, *faxaddmodem* will work from a comprehensive template of parameters that can be edited either during or after the installation process.

faxaddmodem deduces the type of modem attached to the serial port by sending commands to the modem at a fixed speed. By default, this speed is deduced by sending commands to the modem at successively decreasing speeds until a valid response is received. Alternatively, a fixed speed can be specified from the command line with the **−s** option.

Because *faxaddmodem* needs to read and write the modem that is to be installed it will not proceed with an installation if the modem is locked for use by another process.

Finally, note that an installation can be safely aborted at any time by hitting the interrupt key. *faxaddmodem* delays making installation-related changes until after everything has been setup.

OPTIONS

−s *speed* Communicate with the modem at a fixed *speed*; this may be any value recognized by the *stty*(1) command.

PROTOTYPE CONFIGURATION FILES

faxaddmodem uses prototype configuration files that have been created for known modems. These prototype files are kept in the **config** subdirectory and, by convention, have names that identify a brand or type of modem and the DTE-DCE flow control scheme the prototype files configures. The *faxaddmodem*(8C) program that is used to configure a modem for use with *HylaFAX* selects a prototype configuration file using information retrieved from the modem and comments embedded in the prototype files. For Class 1 modems the product ID code returned by the command “ATI0” and the response from the comannd “ATI3” are used to select a prototype configuration file, while for Class 2 modems the manufacturer and model as returned by “AT+FMFR?” and “AT+FMDL?”, respectively, are used (or “AT+FMI?” and “AT+FMM?” for Class 2.0 modems).

A Class 1 prototype configuration file is identified for use by *faxaddmodem* by searching for a comment of the form:

```
# CONFIG:CLASS1:144:.*:RTSCTS: Manufacturer='AT&T' Model=Dataport
```

In this example "144" is the product ID code for an AT&T DataPort modem, ".*" is a regular expression matched against the result string returned by the "ATI3" command, and "RTSCTS" indicates the modem is configured to use hardware flow control during fax operation. The remainder of the line is evaluated by the *sh*(1) and used to specify the modem's manufacturer and model (since Class 1 modems do not have standard commands to query this information).

Class 2 and 2.0 prototype configuration files match the string "*manufacturer-model-flowcontrol*" against a *sh*(1) glob pattern specified in the configuration file, where *manufacturer* and *model* are the strings returned by querying the modem and *flowcontrol* is either "RTSCTS" for hardware flow control or "XONXOFF" for software flow control. For example:

```
# CONFIG: CLASS2: ZyXEL*-RTSCTS
# CONFIG: CLASS2.0: USRobotics*-XONXOFF
```

are configuration comments that appear in the prototype file for a ZyXEL 1496E with Class 2 support, and for a US Robotics Courier modem with Class 2.0 firmware.

The *faxaddmodem* program merges server-specific configuration parameters into a prototype configuration according to comments placed in the prototype file. All lines between "BEGIN-SERVER" and "END-SERVER" comments are placed with the appropriate server configuration parameters. Note that this means modem-related configuration parameters must be placed outside this area of the file.

FILES

/var/spool/hylafax/etc/setup.cache	master configuration parameters
/var/spool/hylafax/etc/setup.modem	modem-related configuration support
/var/spool/hylafax/config	prototype configuration files
/usr/sbin/ondelay	program to open device without waiting for carrier
/usr/sbin/lockname	SVR4 UUCP lockfile name generator

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

hylafax-server(5F), *hylafax-config*(5F), *faxsetup*(8C), *faxq*(8C), *faxgetty*(8C), *faxmodem*(8C), *faxconfig*(8C), *faxsend*(8C)

NAME

faxadduser − add a fax user to the permissions file

SYNOPSIS

/usr/sbin/faxadduser [**−a** *admin-password*] [**−f** *permissions file*] [**−p** *password*] [**−u** *uid*] *username*

DESCRIPTION

faxadduser creates a new user for HylaFAX services, updating appropriate permission files(see *hosts.hfaxd*(5F)).

OPTIONS

The following options are available:

−a *admin-password*
Password for the user to gain admin privileges.

−f *permissions-file*
Use a permissions file other than the default.

−p *password*
Password for the user to access fax services.

−u *user-id* User id number for user.

SEE ALSO

faxdeluser(8C), *hosts.hfaxd*(5F),

NAME

faxalter − alter parameters of a *HylaFAX* job queued for transmission

SYNOPSIS

faxalter [*options*] *id...*

DESCRIPTION

faxalter changes the state of one or more *HylaFAX* jobs that are queued for transmission. Numerous parameters can be altered; see the options below.

An *id* is a number that identifies a *job* that has been submitted to *HylaFAX*; these numbers are printed by *sendfax*(1) when jobs are queued for transmission and can also be displayed with *faxstat*(1). Unprivileged users can alter the parameters of jobs that they own; clients with administrative privileges on a server can manipulate any job. By default, *faxalter* contacts the facsimile server on the host specified in the **FAXSERVER** environment variable; but consult the **−h** option for more information.

OPTIONS

−a *time* Specify a new time to send a job. The syntax for specifying a time is the same as used by *sendfax*(1) (and *at*(1)).

−h *host* Force the server to be contacted on a specific *host*. The *host* may be either a symbolic name or a network address. If no **−h** option is supplied, *faxalter* uses the FAXSERVER environment variable to identify the *HylaFAX* server to contact. If this variable is not set, then *faxalter* checks for a setting in the configuration files (first in the per-user file and then in the system-wide file). If all of the above fails, then *faxalter* attempts to contact a server on the machine where it is run.

−k *time* Specify a new time to kill a job; the syntax is the same as used for the **−a** option.

−m *modem*
Assign/reassign the jobs to the specified *modem*. Specifying ‘‘any’’ for *modem* causes the jobs to be scheduled for any available modem.

−p The jobs should be *pushed*; that is, they should be sent immediately (this is equivalent to ‘‘−a now’’.)

−P *pri* Set the scheduling priority for the specified jobs. The priority is an integer value in the range 0-255; lower values give a job higher priority. The default scheduling priority for a job is 127 and priorities are recalculated after each attempt to process a job.

−t *tries* Set the maximum number of times the server should call the remote fax machine.

−n *scheme* Select a notification scheme to use for the specified jobs. Possible schemes are: ‘‘*none*’’ for no notification (default), ‘‘*when done*’’ for notification when the job is completed, ‘‘*when requeued*’’ for notification when the job is requeued after a failed transmission attempt. The **−Q**, **−D**, and **−R** options are shorthand for these notification schemes, respectively. One can also leave off ‘‘*when*’’ when using the **−n** option; e.g., ‘‘*done*’’ instead of ‘‘*when done*’’.

−v Trace the protocol exchanges between *faxalter* and the remote *hfaxd* process. Tracing information is written to the standard output.

CONFIGURATION PARAMETERS

faxalter reads configuration information from the file **/usr/share/fax/hyla.conf** and then from the file **˜/.hylarc**. Configuration files follow the conventions described in *hylafax-client*(1). The following configuration parameters are recognized:

Tag	**Type**	**Default**	**Description**
Host	string	localhost	host to contact for service
Port	integer	4559	port to use in contacting server
Protocol	string	tcp	protocol to use in contacting server
Verbose	boolean	No	whether or not to enable protocol tracing

The configuration parameters are explained below:

Host The host to contact for service. (Equivalent to the **−h** option.)

Port The network port to contact for service. (Eqvuialent to the **−h** option.)

Protocol The name of the communication protocol to use when contacting a server. (Equivalent to the **FAXSERVICE** environment variable.)

Verbose Control protocol tracing. (Equivalent to the **−v** option.)

BUGS

Setting a job's killtime without also setting the time to send may cause the wrong value for the killtime to be sent. This is because the killtime is calculated as the difference in time between the killtime and time to send: and if the time to send is not specified, then *faxalter* uses the current time instead of querying the server for the job's time to send.

faxalter suspends a job before attempting to alter its state. If a job is active when a suspend request is made there may be a delay while waiting for the job to enter a consistent state where it can be suspended. If *faxalter* is interrupted during this time then jobs may be left suspended.

faxalter only understands how to alter a subset of the available job parameters; there are many more client-controllable parameters that can only be modified by communicating directly with the server using the underlying protocol.

The continued use of the *at*(1)-style notation for specifying times is questionable (it was done for backwards compatibility).

SEE ALSO

hylafax-client(1), *sendfax*(1), *faxrm*(1), *faxstat*(1), *hfaxd*(8C)

NAME

faxanswer − tell a *HylaFAX* server to answer the telephone

SYNOPSIS

/usr/sbin/faxanswer [**−q** *queue-dir*] [**−h** *how*] *modem*

DESCRIPTION

faxanswer sends a message to the *HylaFAX faxgetty*(8C) process servicing *modem* telling it to answer the telephone. This is useful, for example, when a modem is used on a shared phone line and the server is configured to not normally answer the phone.

The specified *modem* can either be the name (typically the last component of the terminal port the modem is attached to), or the full name of the associated FIFO named pipe file, e.g. ‘‘FIFO.ttym2’’.

OPTIONS

−q *dir* Use a spooling area other than */var/spool/hylafax*.

−h Specify exactly *how* the phone should be answered. By default, the server process is instructed to answer the phone and accept any kind of call (voice, data, fax). Discrimination of the type of call is dependent on a modem’s *adaptive-answer* support. The **−h** option can be used to force the server to answer the phone for particular type of call: one of *fax*, *data*, *voice*, *any*, or *extern* (to force *faxgetty* to invoke an external application to to handle the call).

FILES

/var/spool/hylafax	default spooling area
/var/spool/hylafax/FIFO	fifo for contacting *faxq*

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

hylafax-server(5F), *faxgetty*(8C)

NAME

faxconfig − dynamically change *HylaFAX* configuration parameters

SYNOPSIS

/usr/sbin/faxconfig [**−q** *queue-dir*] [**−m** *modem*] [*parameter value* ...]

DESCRIPTION

faxconfig sends messages to a *HylaFAX* server process telling it to install new values for one or more configuration parameters. These new values are maintained only for as long as that server process executes; to setup long-lasting configuration parameters the appropriate configuration file(s) must be edited (see *hylafax-config*(5F)).

faxconfig is especially useful for setting certain configuration parameters that control the operation of *HylaFAX* server processes. For example, by manipulating the **ModemPriority** parameter of one or more *faxgetty*(8C) processes it is easy to dynamically reconfigure the order in which *HylaFAX* will assign modems to outbound jobs.

If no *modem* is specified with the **−m** option, then *faxconfig* will send messages to the central scheduler process, *faxq*(8C). If a *faxgetty* process is to be contacted, the specified *modem* must be the last component of the character special device filename the modem is attached to.

Consult the *hylafax-config*(5F) for the list of configuration parameters that can be set with *faxconfig*.

OPTIONS

The following options are available:

−m *devid* Direct messages to the *faxgetty* process servicing *devid* instead of the central scheduler process, *faxq*. The *devid* is an identifier derived from the name of the modem's device file; for example *ttym2* for */dev/ttym2*, *term_10* for */dev/term/10*.

−q *dir* Use a spooling area other than */var/spool/hylafax*.

FILES

/var/spool/hylafax	default spooling area
/var/spool/hylafax/FIFO	fifo for contacting *faxq*

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

hylafax-server(5F), *faxmodem*(8C), *faxstate*(8C), *faxgetty*(8C), *faxq*(8C), *hylafax-config*(5F).

NAME

faxcover – generate a POSTSCRIPT cover page for an outgoing facsimile

SYNOPSIS

faxcover [*options*] **–f** *from-name* **–n** *fax-number*

DESCRIPTION

faxcover generates a POSTSCRIPT® cover page document on the standard output. The cover page fills the entire area of a default page and is created according to the information supplied on the command line and a *cover sheet template* file.

faxcover is normally invoked by the *sendfax*(1) program to generate a cover page for each outgoing facsimile.

faxcover works by creating a POSTSCRIPT dictionary that contains definitions for the variable information that is to be placed on the facsimile cover page. Specifically, a dictionary with the following symbols are defined:

Symbol	**Description**	**Symbol**	**Description**
to	destination person	*from*	*sending* person
to-company	destination company	*from-company*	sender's company
to-location	destination geographic location	*from-location*	sender geographic location
to-voice-number	person's voice phone number	*from-voice-number*	sender's voice phone number
to-fax-number	destination fax phone number	*from-fax-number*	sender's fax phone number
comments	unbroken comment string	*commentsX*	*X*'th line of comments
regarding	Re: subject	*page-count*	# pages other than cover page
pageWidth	page width in millimeters	*todays-date*	current date and time
pageLength	page length in millimeters		

If the *comments* string is longer than 35 characters, it is broken, on word boundaries, into multiple POSTSCRIPT strings *commentsX* where *X* is the line number; e.g. *comments2* for the second line of comments. By default *faxcover* will generate 20 *commentsX* strings, emitting null string values for empty lines. This number can be changed with the **–m** option; see below.

faxcover also emits a POSTSCRIPT procedure for breaking comment strings into multiple lines using the POSTSCRIPT font metric information. This procedure is named *BreakIntoLines* and has the following usage convention:

%<line-width> <line-height> <x> <y> <string> BreakIntoLines -

eg

```
/rule (________________________________________________________) def
/lw rule stringwidth pop def /lh 30 def
/y 191.4 def /x 0 def
y −30 0 { x exch moveto rule show } for
lw lh x y comments BreakIntoLines
```

where the width of the rule controls the length of each line of comments (note that it is made slightly shorter here for the purpose of presentation), *x* and *y* are initialized to coordinate locations on the page where the comment string should be imaged, and the number 30 is the inter-line spacing appropriate for the font used to image the comment lines.

If information is unavailable for an item, the associated POSTSCRIPT symbol is defined to be a null string.

Following the definition prologue, a cover sheet template file is copied. This file is expected to be a POSTSCRIPT program that uses the generated strings to create a cover page. The template must include a **showpage** operator to cause the page to be imaged. The above symbols will be defined in the *current dictionary*. A trailing **end** operator is supplied by *faxcover*, so any manipulation of the dictionary stack should leave the current dictionary on the top of the stack on completion.

OPTIONS

−c *comments* Use *comments* for the comment string. You may use '\n' to break it into several lines on generated the cover page.

−C *file* Use *file* as the template file. The default template file is named *faxcover.ps*. The template file can also be specified by setting the FAXCOVER environment variable. If the template filename is not an absolute pathname, then *faxcover* looks first for this file in the home directory of the sender. If no such file is present, then it looks in the library directory where the *HylaFAX* client application data are installed. If no template file is located, *faxcover* will terminate without generating a cover page.

−D *format* Use *format* as the formatting specification passed to *strftime*(3) when generating the time and date placed on the cover page.

−f *name* Use *name* for the sender's identity.

−l *location* Use *location* for the geographic location of the destination company.

−n *number* Use *number* for the destination fax machine number.

−p *count* Use *count* for the count of pages to follow the cover page.

−r *regarding* Use *regarding* for the ‘‘Re:’’ field.

−s *size* Set the page size to use for the transmitted facsimile. Facsimile are normally imaged with a system-default page size (usually letter-size pages, 8.5" by 11", for sites in North America). Alternate page sizes are specified symbolically using either the name or abbreviation of an entry in the *pagesizes*(5F) database; e.g. *a3* (ISO A3), *a4* (ISO A4), *a5* (ISO A5), *a6* (ISO A6), *b4* (ISO B4), *na-let* (North American Letter), *us-leg* (American Legal), *us-led* (American Ledger), *us-exe* (American Executive), *jp-let* (Japanese Letter), and *jp-leg* (Japanese Legal). Comparisons are case-insensitive and any match of a substring of the full page-size name is sufficient; e.g. ‘‘legal’’ would match ‘‘American Legal’’.

Note that it may not be permissible to image into the full page area; the guaranteed reproducible area for a page is typically inset. Also, note that while arbitrary page sizes can be specified through the page size database, only a limited set of page dimensions are supported by the Group 3 facsimile protocol. Thus if an odd-size facsimile is submitted for transmission it may not be possible to determine if it can be sent until the fax server establishes communication with the remote facsimile machine.

−t *name* Use *name* for the receiver's identity.

−v *number* Use *number* for the receiver's voice phone number.

−x *company* Use *company* for the destination company name.

DIAGNOSTICS

Complaints about not being able to locate a template file.

FILES

˜/.faxdb	personal fax number database
/usr/share/fax/faxcover.ps	default cover page template
/usr/share/fax/pagesizes	page size database

BUGS

Information not supplied on the command line is currently obtained from the *˜/.faxdb* file. *faxcover* should read configuration parameters from the standard *HylaFAX* configuration files.

SEE ALSO

hylafax-client(1), *sendfax*(1)

NAME

faxcron − *HylaFAX* routine maintenance script

SYNOPSIS

/usr/sbin/faxcron [**−n**] [**−l** *lastrun*]

DESCRIPTION

faxcron is a command script that does routine upkeep tasks in a *HylaFAX* spooling directory hierarchy. This script is intended to be invoked from *cron*(8C) on behalf of the fax user once a day, with the standard output sent by mail to the *HylaFAX* administrator. For example, the following crontab might be setup for the fax user:

```
25   23    *    *    *   sh /usr/sbin/faxcron | mail FaxMaster
```

The *faxcron* script does the following work:

- report statistics for facsimile transmitted since the last time *faxcron* was run.
- report statistics for facsimile transmitted in the past week.
- report statistics for facsimile received since the last time *faxcron* was run.
- report statistics for facsimile received in the past week.
- report any recent calls that failed suspiciously and return the associated trace log;
- purge data that is 30 days or older from the remote machine information directory;
- delete information that is 30 days or older from session trace log files;
- remove files from the tmp directory that are older than 1 day;
- remove received facsimile that are 7 days or older;
- report sites that are currently having jobs rejected; and
- force all session log files to be owned by the fax user and kept with protection mode 644.

OPTIONS

−n Forces *faxcron* to run without updating any files.

−l *lastrun* Forces the notion of the last time that *faxcron* was run to be *lastrun*. The value of *lastrun* is a date and time string of the form ‘‘MM/DD/YY HH:MM’’ (the *date*(1) format string ‘‘%D %H:%M’’).

−info *n* Set the expiration time for data in the info database to be *n* days.

−log *n* Set the expiration time for session log data to be *n* days.

−rcv *n* Set the expiration time for files in the received facsimile queue to be *n* days.

−tmp *n* Set the expiration time for files in the tmp directory to be *n* days.

−mode *m* Set the file protection mode for session logs to *m* (a command line argument to *chmod*(1)).

NOTES

This script requires *nawk*(1) or *gawk*(1) and a *date (1)* program that supports the ‘‘+format’’ option.

If session logs are to be kept private this script must be modified to filter out sensitive information such as calling card numbers. (It should also be run with ‘‘−mode 0600’’ so that session log files are not publicly readable.)

FILES

/var/spool/hylafax	spooling area
/var/spool/hylafax/etc/lastrun	timestamp of last run
/usr/sbin/xferfaxstats	for generating transmit statistics
/usr/sbin/recvstats	for generating receive statistics
/var/spool/hylafax/tmp/faxcronXXXX	temporary file for truncating session logs
/var/spool/hylafax/tmp/faxlogXXXX	temporary file for logs of failed calls

SEE ALSO

cron(8C), *faxq*(8C), *faxgetty*(8C), *faxsend*(8C), *xferfaxstats*(8C), *recvstats*(8C), *hylafax-server*(5F)

NAME

faxdeluser − delete a fax user from the permissions file

SYNOPSIS

/usr/sbin/faxdeluser [**−f** *permissions file*] *username*

DESCRIPTION

faxdeluser deletes a user from using HylaFAX services, updating appropriate permission files(see *hosts.hfaxd*(5F)).

OPTIONS

The following options are available:

−f *permissions-file*
Use a permissions file other than the default.

SEE ALSO

faxadduser(8C), *hosts.hfaxd*(5F),

NAME

faxgetty − *HylaFAX* front-door process

SYNOPSIS

/usr/sbin/faxgetty [**−D**] [**−p**] [**−q** *queue-dir*] *device*

DESCRIPTION

faxgetty is the *HylaFAX* server program that listens for incoming phone calls and either handles each call directly or hands it off to an appropriate program. In addition *faxgetty* monitors modem usage and notifies the *HylaFAX* scheduler process when a modem's status changes; e.g. when a modem is busy with an outbound call.

One *faxgetty* should be run for each facsimile modem on a machine. *faxgetty* is typically started by *init*(8C). *faxgetty* can also be used with data-only modems; this may be desirable if some of the non-facsimile features are important, such as the support for screening calls according to Caller-ID information.

faxgetty listens to its facsimile modem for incoming calls and to a FIFO special file on which it accepts command directives from administrative commands such as *faxanswer*(8C) and *faxabort*(8C). *faxgetty* also catches SIGTERM and SIGINT signals; cleaning up any necessary resources before exiting.

Many aspects of *faxgetty* operation are controlled through a configuration file. In particular, for *faxgetty* to automatically invoke the *getty*(8C) program to handle incoming data connections the **GettyArgs** parameter must be setup in the configuration file. The configuration file is located in the spooler hierarchy in the **etc** subdirectory and is named **config.***devid***,** where *devid* is an identifier derived from the name of the modem's device file; for example *ttym2* for */dev/ttym2*, *term_10* for */dev/term/10*. Consult *hylafax-config*(5F) for detailed information on the contents of configuration files.

OPTIONS

−D This option forces *faxgetty* to detach itself from the controlling tty from which it was invoked. In normal operation *faxgetty* will **not** dissassociate itself from the controlling tty.

−p Do not change the process priority when receiving facsimile. Normally *faxgetty* will raise its priority to reduce i/o latency. This option is useful for debugging.

−q *dir* The specified directory is treated as the spooling area. The default spooling area, */var/spool/hylafax*, is defined at the time the software is built.

FILES

/var/spool/hylafax/FIFO.*device*	FIFO special file for submitting commands
/var/spool/hylafax/etc/config.*device*	device-specific configuration file

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

faxabort(8C), *faxanswer*(8C), *faxconfig*(8C), *faxstate*(8C), *faxq*(8C), *getty*(8C), *hylafax-server*(5F), *hylafax-config*(5F)

NAME

faxinfo − print information about a received facsimile

SYNOPSIS

/usr/sbin/faxinfo [**−n**] *file.tif*

DESCRIPTION

faxinfo prints descriptive information on the standard output about a received facsimile file. For example:

```
/var/spool/hylafax/recvq/fax00017.tif:
    Sender: +14159657824
     Pages: 3
   Quality: Normal
      Page: North American Letter
  Received: 1996:01:19 13:51:02
TimeToRecv: 0:39
SignalRate: 14400 bit/s
DataFormat: 2-D MR
```

This information is typically included in the notification mail generated by the *faxrcvd*(8C) script when a facsimile is received by *HylaFAX*.

The **−n** option suppresses the printing of the filename.

NOTES

The information that *faxinfo* prints is obtained from the tags stored in the TIFF image that is written by the *HylaFAX* software. If *faxinfo* is presented with an invalid TIFF image it may print uninteresting information. Similarly if a TIFF image that was not written by *HylaFAX* is supplied as an argument then only partial information may be printed—this is because *HylaFAX* stores certain information in private tags that other TIFF writers may not emit.

SEE ALSO

hylafax-server(5F), *faxrcvd*(8C), *faxgetty*(8C)

NAME

faxmail − *HylaFAX* mail-to-fax gateway application

SYNOPSIS

faxmail [*options*] [*destination* [*from*]]

DESCRIPTION

faxmail takes an electronic mail message on standard input and converts it to POSTSCRIPT® in a form that is suitable for transmission as a facsimile. The converted document is either written to the standard output or, if direct delivery is enabled, it is submitted directly to a *HylaFAX* server for transmission.

faxmail is designed for use in constructing electronic mail to facsimile gateway services. For example, mail aliases may be created to automatically transmit electronic mail; e.g.

```
sam: "|/usr/bin/faxmail -d sam@5551212"
```

or *faxmail* may be used as a ‘‘mail transport agent’’, extracting the necessary delivery information directly from the envelope of the mail message.

If *faxmail* is invoked without enabling direct delivery then it just formats the mail message on the standard input and writes the result to the standard output. To enable direct delivery the **−d** option must be specified on the command line; see below for more information.

FORMATTING

faxmail formats a mail message according to the following rules: First it parses the envelope information interpreting any meta-header information (see below) that is present. Once the entire envelope has been collected it emits a formatted set of header lines. By default all header information except the ‘‘From’’, ‘‘To’’, ‘‘Cc’’, ‘‘Subject’’, and ‘‘Date’’ lines are discarded. Header lines that are kept have the *tag* (the string to the left of the ‘‘:’’) set in a *bold font* and the *value* (the string to the right of the ‘‘:’’) set in an *italic font*. Mail messages that conform to the Multipurpose Internet Mail Extensions (MIME) specification are parsed and handled according to the rules given below. Plain text body parts of a mail message are formatted in a *text font* with any long lines wrapped at word boundaries unless the **−c** option has been specified.

By default, *faxmail* sets all text in 10 point type on an 8.5" by 11" portrait-oriented page with .35" top margin, .25" bottom margin and .25" left and right hand margins. There are command-line options to control the point size, page dimensions, orientation, and multi-column formatting. Text formatting can also be controlled through meta-header directives placed in the envelope of the mail message.

ENVELOPE PROCESSING

faxmail pre-processes the envelope information (i.e. the header lines) before formatting the message contents. Header lines whose names begin with ‘‘X-FAX-’’ (case-insensitive) are handled specially−they are treated as command directives that specify how to generate the resultant POSTSCRIPT or, optionally, how to deliver the resulting document as facsimile. The set of known meta-headers corresponds to the set of configuration parameters defined below. A meta-header is specified as ‘‘X-FAX-*parameter*’’ where *parameter* is the name of a configuration parameter; e.g. ‘‘X-FAX-TabStop’’ to set the number of spaces between tab stops.

Controls for specifying headers to be passed through in the formatted text identify not only which headers but also the order in which the headers should be emitted. *faxmail* initializes the set of headers to retain to: ‘‘To From Subject Cc Date’’. If this order is acceptable then additional headers can simply be added with the X-FAX-Headers directive; e.g. ‘‘X-FAX-Headers: Message-id’’. If however a different order is desired then the header set should be cleared with a ‘‘clear’’ header tag and then completely specified in the desired order; for example,

```
X-FAX-Headers: clear Message-id Date To Subject From Cc
```

will cause headers to be emitted in the order ‘‘Message-Id Date To Subject From Cc’’ (depending on what headers are present). Note in particular that all header lines in the envelope can be suppressed by specifying ‘‘X-FAX-Headers: clear’’; this is useful, for example, when the body of the mail message contains a preformatted document that is to be transmitted.

In addition to the above controls, *faxmail* can also be instructed to substitute an arbitrary string for a header tag when generating the final envelope. This facility can be used to better format ad-hoc header information that is passed through the envelope of the message. The ''X-FAX-MapHeader'' meta-header specifies how to map a header line. For example,

X-FAX-MapHeader: x_FAX_For Deliver FAX To

would cause any header ''x_FAX_For'' that appeared in the envelope to be replaced in the formatted envelope by ''Deliver FAX To''.

MIME PROCESSING

faxmail parses MIME mail messages and does some rudimentary work to:

- strip out unprintable content such as audio, video, or binary data,
- decode encoded parts,
- insert ''digest dividers'' between multipart/digest subparts,
- format message/rfc822 parts as described above for the top-level envelope, and
- optionally convert graphical parts (e.g. images) for display.

MIME processing is fairly simple and (currently) somewhat constrained. *faxmail* has builtin support for the following MIME parts: text/plain, multipart/mixed, multipart/digest, message/rfc822, application/postscript, and application/x-faxmail-prolog. Parts can also be processed through external processing scripts that *faxmail* looks for in a ''MIME converters'' directory hierarchy. External scripts may override builtin processing or supplement the builtin support. For each MIME body part with type *T* and subtype *S* *faxmail* checks first for an executable script named T/S in the converter hierarchy. If a script exists then it is run and the result is appended to the output PostScript document. Otherwise if the part has builtin support then it is processed directly. Any part that does not have external or builtin support is discarded and replaced by a message that indicates the part was removed. This discarded message can be suppressed with the *MarkDiscarded* configuration parameter (also settable with an X-FAX-MarkDiscarded line in the envelope).

The builtin handling support is as follows: text/plain parts are formatted using the default *text font* and point size; multipart/mixed are ''burst'' and interpreted per the specification but are otherwise unformatted; multipart/digest are burst and an optional ''digest divider'' marking may be inserted before each subpart; message/rfc822 are formatted as described above with envelope header lines culled and formatted with bold and italic fonts (in addition, if there is insufficient space in the current output page/column for the message envelope, optional divider, and one line of text, then *faxmail* will insert a ''break'' so the the message starts at the top of the next page/column); application/postscript are copied through untouched to the output; application/x-faxmail-prolog are copied through untouched to the end of the prologue section of the generated PostScript document to permit customization of the formatted output.

faxmail supports the following Content-Transfer-Encoding schemes: 7bit, 8bit, binary, base64, quoted-printable, and x-uuencode. Any character set that is not us-ascii is treated as iso-8859-1.

In general it is recommended that senders either permit *faxmail* to format message contents or completely bypass the formatting facilities and submit data that is to be processed by *sendfax*. Trying to combine the two facilities by, for example, combining PostScript with text that is to be formatted is unlikely to work well because *faxmail* does not track the amount of space on the page that a non-text MIME part uses.

DIRECT DELIVERY

When *faxmail* is invoked with the **−d** option it delivers the formatted document directly to a *HylaFAX* server for transmission as facsimile. Command line arguments may be supplied to specify the delivery destination and sender identity; typically from information extracted by the mail transport facility. A command line *destination* is specified with the same syntax as the argument for the **−d** option to the *sendfax*(1) command. Similarly any *from* identity specified on the command line follows the same rules as the **−f** option to *sendfax*. An explicit dialstring to use in delivery can also be specified with an X-FAX-Dialstring header in the mail message envelope. If no sender identity is provided on the command line then *faxmail* will extract it from the ''From'' line in the envelope. *faxmail* will not submit a message for delivery if either the dialstring or sender identity is missing or null.

When direct delivery is enabled X-FAX- header lines may be included in the mail message envelope to control the submission and delivery process. As above these lines are specified as ‘‘X-FAX-*parameter*’’ where *parameter* is the name of a configuration parameter for the *sendfax* program; e.g. ‘‘X-FAX-VRes’’ to set the vertical resolution of the transmitted facsimile. By default automatic cover page generation is enabled when direct delivery is used; this can be overridden with the **−n** option on the command line or by including an X-FAX-AutoCoverPage header in the message envelope.

OPTIONS

−1 Set text in one column (default).

−2 Set text two columns.

−b *font* Make *font*, a POSTSCRIPT font name, the ‘‘*bold font*’’ used to set header lines. The default bold font is Helvetica-Bold.

−c Clip long text lines instead of wrapping them at page margins.

−d Enable direct delivery of facsimile; the formatted document will be submitted directly to a *HylaFAX* server for transmission as facsimile. This option is similar to piping the output of *faxmail* to the input of *sendfax*(8C) except when direct delivery is enabled *faxmail* interprets ‘‘x-fax-’’ header lines in the envelope of the mail message to look for control information to use in delivering the facsimile and, by default, no automatic cover page generation is done.

−f *font* Make *font*, a POSTSCRIPT font name, the text font used to set the body of mail messages. The default text font is Courier.

−H *height* Use *height* as the height, in inches, of the output page. The default page height is taken from the default page size.

−i *font* Make *font*, a POSTSCRIPT font name, the ‘‘*italic font*’’ used to set header lines. The default italic font is Helvetica-Oblique.

−Ml=*#***,r=***#***,t=***#***,b=***#*
Set the page margins; the default margins are: left and right .25 inch, top .35 inch, bottom .25 inch.

−n Suppress auto cover page generation when doing direct delivery.

−p *size* Set all text using *size* for the font point size.

−r Set pages rotated by 90 degrees (in ‘‘Landscape mode’’).

−R Set pages unrotated (in ‘‘Portrait mode’’).

−s *size* Set the page size to use. Cover pages are normally generated using a system-default page size (usually letter-size pages, 8.5" by 11", for sites in North America). Alternate page sizes are specified symbolically using either the name or abbreviation of an entry in the *pagesizes*(5F) database; e.g. *a3* (ISO A3), *a4* (ISO A4), *a5* (ISO A5), *a6* (ISO A6), *b4* (ISO B4), *na-let* (North American Letter), *us-leg* (American Legal), *us-led* (American Ledger), *us-exe* (American Executive), *jp-let* (Japanese Letter), and *jp-leg* (Japanese Legal). Comparisons are case-insensitive and any match of a substring of the full page-size name is sufficient; e.g. ‘‘legal’’ would match ‘‘American Legal’’.

−u *user* Set the user name to use when logging in to do direct delivery. By default the user is specified by the **MailUser** configuration parameter (see below). If a null user name is specified, then the facsimile will be submitted using the identity of the user that invoked *faxmail*.

−v Enable tracing of envelope, MIME, and job submission processing. Diagnostic information is written to the standard error (envelope and MIME processing) and standard output (job submission).

−W *width* Use *width* as the width, in inches, of the output page. The default page width is taken from the default page size.

CONFIGURATION PARAMETERS

faxmail reads configuration information from the files **/usr/share/fax/hyla.conf**, and **~/.hylarc**; in that order. Configuration files follow the conventions described in *hylafax-client*(1). The following configuration parameters are recognized:

Tag	**Type**	**Default**	**Description**
AutoCoverPage	boolean	Yes	automatically generate cover page
BoldFont	string	Helvetica-Bold	font for setting header tags
Columns	integer	1	number of columns in formatted output
DigestDivider	string	−	multipart/digest divider POSTSCRIPT command
FontPath	string	*see below*	directory for font metric files
GaudyHeaders	boolean	No	enable/disable gaudy page headers
Headers	string	*see below*	headers to retain in envelope
ISO8859	boolean	Yes	use ISO 8859-1 character encoding
ItalicFont	string	Helvetica-Oblique	font for setting header values
LineWrap	boolean	Yes	wrap/truncate text lines
MailUser	string	−	user identity for doing direct delivery
MarkDiscarded	boolean	Yes	mark discarded MIME body parts
MapHeader	string	−	define header mapping
MIMEConverters	string	*see below*	pathname of MIME converter scripts
Orientation	string	portrait	orientation of text on page
OutlineMargin	inches	0	width of outline line
PageCollation	string	forward	collate pages in forward or reverse direction
PageHeaders	boolean	Yes	enable/disable page headers
PageHeight	float	−	output page height
PageMargins	string	*see below*	margins for formatted page
PageSize	string	default	output page size from database
PageWidth	float	−	output page width
Prologfile	string	*see below*	pathname of POSTSCRIPT prologue file
TabStop	integer	8	inter-stop setting in characters
TextFont	string	Courier	name of font for setting text
TextLineHeight	inches	−	text formatting line height control
TextPointSize	inches	*see below*	size to use in setting text
Verbose	boolean	No	trace envelope and MIME processing

Values marked as *inches* are specified using a syntax that identifies one of several possible units:

#.##bp	big point (1in = 72bp)
#.##cc	cicero (1cc = 12dd)
#.##cm	centimeter
#.##dd	didot point (1157dd = 1238pt)
#.##in	inch
#.##mm	millimeter (10mm = 1cm)
#.##pc	pica (1pc = 12pt)
#.##pt	point (72.27pt = 1in)
#.##sp	scaled point (65536sp = 1pt)

Unit names can be upper or lower case but no white space is permitted between the number and the unit. Values specified with no unit are interpreted as points.

The configuration parameters are explained below. Most parameters correspond to a command line option. Parameter values identified above as *inches* are converted according to the conventions described above.

AutoCoverPage Control whether or not a cover page is automatically generated for each job.

BoldFont The name of the font to use to set header tags (i.e. the string to the left of the ":").

Columns The number of columns to set text in. (Equivalent to the **−m** option.)

DigestDivider The string to emit in the output before each subpart of a multipart/digest mail message. This string is typically a POSTSCRIPT procedure that draws a mark of some sort. Dividers are expected to use no more vertical space on the output page than a line of text.

FontPath The path where Adobe Font Metric (AFM) files are located; by default /usr/share/ghostscript/5.50:/usr/share/ghostscript/5.50/vflib:/usr/share/ghostscript/fonts:/usr/share/fonts/defa font/pfb.

GaudyHeaders Control whether or not to use a gaudy-style page header. Enabling gaudy headers implicitly enables page headers.

Headers Define the headers retained from the envelope and specify the order that they should be emitted in the formatted output. The set of headers is initialized to ‘‘To From Subject Cc Date’’. Headers specified are appended to this list except for a ‘‘clear’’ header that causes the current set of headers to be discarded.

ISO8859 Control the use of ISO 8859-1 encoding in the generated POSTSCRIPT

ItalicFont The name of the font to use to set header values (i.e. the string to the right of the ‘‘:’’).

LineWrap Control whether long text lines are wrapper or truncated at the right hand margin. (Equivalent to the **−c** option.)

MailUser The account name to use to login to a fax server when doing direct delivery. Note that this account name is not used for the identity of the sender; this comes from the command line or the ‘‘From’’ line in the mail message. If a null account name is specified, then the facsimile will be submitted using the identity of the user that invoked *faxmail*.

MapHeader Define a substitution for the specified header that should be done each time the header is emitted in the formatted envelope. Header tags are matched in a case-insensitive manner.

MarkDiscarded Control whether discarded MIME parts are replaced by a text message indicating the original content was removed.

MIMEConverters
The pathname of a directory hierarchy that has scripts for external processing of MIME body parts. The default pathname is /usr/sbin/faxmail.

Orientation Control whether pages are oriented horizontally (‘‘landscape’’) or vertically (‘‘portrait’’). (Equivalent to the **−r** and **−R** options.)

OutlineMargin Control whether columns of text have a line drawn around them and the width of the line. Setting this parameter to 0 disables outlines.

PageCollation Control whether the output file has pages collated in the same order as the input file (‘‘forward’’) or in reverse order (‘‘reverse).

PageHeaders Control whether page headers are generated.

PageHeight Set the output page height (in inches).

PageMargins Set the output page margins. Margins are specified as string of the form: ‘‘**l=#**,**r=#**,**t=#**,**b=#**’’ where **l** indicates the left margin, **r** indicates the right margin, **t** indicates the top margin, **b** indicates the bottom margin, and numbers are interpreted as *inches*. (Equvalent to the **−M** option.)

PageSize Set the output page dimensions by name. (Equivalent to the **−s** option.)

PageWidth Set the output page width (in inches).

PrologFile The pathname of a file containing POSTSCRIPT that should be included in the prologue section of the generated POSTSCRIPT. The default prologue file is /usr/share/fax/faxmail.ps.

TabStop	Set the tab stop distance (in characters).
TextFont	Set the name of font to use for setting text. (Equivalent to the **−f** option.)
TextLineHeight	Set the vertical text line height and spacing.
TextPointSize	Set the point size to use in setting plain text. (Equvalent to the **−p** option.)
Verbose	Control tracing envelope and MIME processing.

NOTES

Because a sender's identity in an electronic mail message is inherently untrustworthy, using *faxmail* to build a mail to fax gateway is problematic. Unless mail service is somehow restricted or the sender's identity is verified using a mechanism such as RFC 1847's multipart/signed MIME type there is no reliable way to restrict access to facilities setup with *faxmail*.

BUGS

Only the last instance of a header is kept and written to the output. This means, for example, that only the last of many ''Received'' lines will be included in the formatted output.

FILES

~/.hylarc	per-user configuration file
/usr/share/fax/pagesizes	page size database
/usr/share/fax/faxmail.ps	POSTSCRIPT prologue
/usr/share/fax/hyla.conf	system-wide configuration file
/usr/share/fax/faxmail.conf	system-wide configuration file
/usr/share/fax/sendfax.conf	system-wide configuration file for direct delivery
/usr/sbin/faxmail	hierarchy for external MIME converters
/usr/share/ghostscript/5.50:/usr/share/ghostscript/5.50/vflib:/usr/share/ghostscript/fonts:/usr/share/fonts/default/ghostscript:/usı	
/usr/tmp/faxmailXXXXXX	temporary files

SEE ALSO

hylafax-client(1), *textfmt*(1), *sendfax*(1)

NAME

faxmodem − dynamically add a modem to a *HylaFAX* server system

SYNOPSIS

/usr/sbin/faxmodem [**−q** *queue-dir*] [**−c** *capabilities*] [**−p**] [**−P**] [**−u** *priority*] *modem*

DESCRIPTION

faxmodem sends a message to the *HylaFAX* queuer process *faxq*(8C) telling it that the specified modem is ready for use and informing it about its fax-related capabilities. This is the mechanism by which modems are added in a send-only configuration. Once a modem has been configured its status can be reconfigured using the *faxstate*(8C) program. *faxmodem* can also be used to alter the capabilities and usage priority of a previously configured modem.

The specified *modem* is either the terminal device name where the modem is attached or a *HylaFAX* device identifier (see *hylafax-config*(5F) for information on device identifiers). Device names may be given without a leading directory pathname; e.g. ttyf2 instead of /dev/ttyf2.

To figure out the capabilities of a modem either use the *probemodem*(8C) script or manually communicate with the modem. For a Class 2 modem the capabilities are given in the response to the “AT+FDCC=?” query command and for a Class 2.0 modem the “AT+FCC=?” command. To check if a modem is capable of polled retrieval of documents use “AT+FSP=?” for a Class 2.0 modem or “AT+FSPL=?” for a Class 2 modem; if “1” is indicated in the response then the modem supports polling. For example,

hyla% **cu -l ttyf2**
Connected
at+fclass=2.0
OK
at+fcc=?
(0,1),(0-5),(0-2),(0-2),0,0,0,(0-7)
OK
at+fsp=?
(0,1)
OK

In this case the modem would be added using the command:

```
/usr/sbin/faxmodem -c '(0,1),(0-5),(0-2),(0-2),0,0,0,(0-7)' ttyf2
```

(by default polling is assumed to be supported).

Class 1 modems require a different technique. The host implements most of the fax protocol so all that is needed is to identify the possible signalling rates the modem supports for transmitting; this is done with the “AT+FTM=?” command. For example,

hyla% **cu -l ttyf2**
Connected
at+fclass=1
OK
at+ftm=?
24,48,72,73,74,96,97,98,121,122,145,146
OK

In this case the modem supports 2400, 4800, 7200, 9600, 12200, and 14400 bps signalling rates so the modem would be added using:

```
/usr/sbin/faxmodem -c '(0,1),(0-5),(0-2),(0-2),(0,1),0,0,(0-7)' ttyf2
```

OPTIONS

The following options are available:

−c Specify the fax capabilities using the syntax defined by the “Class 2” specification: “(*vr*),(*br*),(*wd*),(*ln*),(*df*),(*ec*),(*bf*),(*st*)”. where, *vr* specifies vertical resolution, *br* specifies bit rate, *wd* specifies page width, *ln* specifies page length, *df* specifies data compression, *ec* specifies error correction, *bf* specifies binary file transfer, and *st* specifies scan time/line; and each of the above specifications is a range or list of numbers that defines the exact capabilities of the modem. For example, the default capabilities are ‘‘(0,1),(0-3),(0-4),(0-2),(0),(0),(0),(0-7)’’ which specifies the modem is capable of sending and receiving both low and high resolution facsimile with a variety of page sizes; supports signalling rates 0-3 (2400 bps through 9600 bps); supports only 1D-encoded data compression; and does not support error correction or binary file transfer. A modem’s ability to support polled retrieval of facsimile is specified separately with the **−p** and **−P** options.

−p Specify the modem is **not** capable of polling for remote documents.

−P Specify the modem is capable of polling for remote documents (default).

−q *dir* Use a spooling area other than */var/spool/hylafax*.

−u *priority* Assign the specified priority to the modem when scheduling it for outbound use. Modem priorities are numbers in the range [0..255] with lower numbers meaning higher priority. Modems are initially assigned priority 255.

FILES

/var/spool/hylafax	default spooling area
/var/spool/hylafax/FIFO	fifo for contacting *faxq*

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

hylafax-server(5F), *faxgetty*(8C), *faxq*(8C).

NAME

faxqclean − *HylaFAX* queue cleaner process

SYNOPSIS

/usr/sbin/faxqclean [*options*]

DESCRIPTION

faxqclean is a program that processes completed *HylaFAX* jobs and expunges unreferenced document files. This program is intended to be invoked by *cron*(8C) on behalf of the super user (i.e. root) one or more times a day. For example, the following crontab entry might be setup for root to run *faxqclean* each hour.

```
0    *    *    *    *    /usr/sbin/faxqclean
```

When *faxqclean* is run it scans the **doneq** subdirectory in the *HylaFAX* spooling area and process each job according to the **doneop** field specified in the job description file; c.f. *sendq*(5F). Jobs that are marked for removal are purged and references to documents are removed. Jobs that are marked for archival may be archived depending on the options supplied on the command line.

After scanning for completed jobs *faxqclean* scans the **docq** subdirectory and builds up a table of document files. Files that are not referenced by any job and that are older than a specified threshold are removed.

OPTIONS

−a Enable job archiving support. *NB: v4.0 does not include support for job archiving.*

−j *secs* Set the job age threshold (in seconds) that controls how long jobs may reside in the **doneq** directory before being processed. By default this value is 15 minutes (15*60 = 900 seconds).

−d *secs* Set the document age threshold (in seconds) that controls how long unreferenced files may reside in the **docq** directory before being removed. By default this value is 1 hour (60*60 = 3600 seconds).

−n Do not carry out any work; just show what would be done. This option is useful together with the **−t** option for debugging.

−q *dir* The specified directory is treated as the spooling area. The default spooling area, */var/spool/hylafax*, is defined at the time the software is built.

−t Trace in great detail exactly what *faxqclean* is doing. Messages are sent to the standard output.

−v Print messages on the standard output about jobs removed or archived and about document files removed.

FILES

/var/spool/hylafax/doneq	directory to scan for completed jobs
/var/spool/hylafax/docq	directory to scan for unreferenced documents
/var/spool/hylafax/archive	directory where archived jobs are placed

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

faxq(8C), *hylafax-server*(5F), *sendq*(5F), *doneq*(5F), *archive*(5F).

NAME

faxq − *HylaFAX* queue manager process

SYNOPSIS

/usr/sbin/faxq [**−D**] [**−q** *queue-dir*]

DESCRIPTION

faxq is the central queueing agent for *HylaFAX*. One (and only one) *faxq* process is normally running at all times. *faxq* is responsible for scheduling, preparing, and initiating outbound facsimile jobs (and potentially other types of jobs as well).

faxq may be run by itself to provide outbound-only services or it may be run together with one or more *faxgetty*(8C) processes, in which case inbound services (e.g. receiving facsimile) are provided. When *faxq* is run by itself the set of available modems must be specified using the *faxmodem*(8C) command. Each modem specified in this way is assumed to exist and to be capable of use as a facsimile modem. When *faxq* is run in conjunction with *faxgetty* processes, it is not necessary to use *faxmodem* to specify modems; instead the programs communicate information about the status and capabilities of each modem. The latter setup provides a more robust and powerful system and is the intended way for the software to operate.

faxq receives notification of outbound jobs to process from the *hfaxd*(8C) process that implements the server-side of the *HylaFAX* client-server protocol. Notification is through messages sent to a FIFO special file that *faxq* listens to. This file is also used by other administrative programs such as *faxquit*(8C).

Many aspects of *faxq* operation may be controlled through a configuration file. This file is located in the spooler hierarchy in the **etc** subdirectory and is named **config** (as opposed to the per-modem configuration files named **config.***devid***);** consult *hylafax-config*(5F) for detailed information on this configuration file. Note that *faxq* automatically re-reads its configuration file if it has been changed. It is also possible to dynamically change configuration parameters in a running process with the *faxconfig*(8C) program. Beware however that parameters changed in this way will be reset to the values specified in the configuration file if the file is re-read.

OPTIONS

−q *dir* The specified directory is treated as the spooling area. The default spooling area, */var/spool/hylafax*, is defined at the time the software is built.

−D This option forces *faxq* to remain attached to the controlling tty from which it was invoked. In normal operation *faxq* will *fork*(2) and dissassociate itself from the controlling tty. This option is useful for debugging.

FILES

/var/spool/hylafax/FIFO	FIFO special file for submitting commands
/var/spool/hylafax/etc/config	configuration file

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

faxabort(8C), *faxconfig*(8C), *faxmodem*(8C), *faxquit*(8C), *faxstate*(8C), *getty*(8C), *hylafax-server*(5F), *hylafax-config*(5F).

NAME

faxquit − tell a *HylaFAX* server to terminate

SYNOPSIS

/usr/sbin/faxquit [**−q** *queue-dir*] [*modem*]

DESCRIPTION

faxquit sends a message to a *HylaFAX* server process telling it to quit operation.

If *modem* is specified, the *faxgetty*(8C) process servicing the specified modem is notified. *modem* can either be the name (typically the last component of the terminal port the modem is attached to), or the full name of the associated FIFO named pipe file, e.g. ‘‘FIFO.ttym2’’.

If no modem is specified, *faxquit* will send the request to the ‘‘FIFO’’ named pipe file causing the *HylaFAX* queue manager process to terminate.

Note that server processes may not terminate immediately. In particular the *faxq* process will not exit until any active jobs are completed; this is necessary for proper operation (*faxq* must wait for child processes to terminate so that it can collect their exit status and update the state of the jobs being operated on).

OPTIONS

−q *dir* Use a spooling area other than */var/spool/hylafax*.

FILES

/var/spool/hylafax	default spooling area
/var/spool/hylafax/FIFO	fifo for contacting *faxq*

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

hylafax-server(5F), *faxq*(8C), *faxgetty*(8C)

NAME

faxrcvd − *HylaFAX* notification script for received facsimile

SYNOPSIS

/var/spool/hylafax/bin/faxrcvd *qfile devid commid error-msg cidnumber*

DESCRIPTION

bin/faxrcvd is the command script invoked by the facsimile server whenever a facsimile is received. The default script sends electronic mail to the *FaxMaster* user describing the content of the facsimile and other useful information such as the time spent receiving the document. The arguments are:

qfile the pathname of the received TIFF file relative to the root of the spooling hierarchy.

devid the modem device on which the facsimile was received.

commid the communication identifier for the inbound call.

error-msg an error message that is non-null if an error was encountered during the receive operation.

cidnumber
the value of **CIDNumber** when using **QualifyCID**

NOTES

This script can route facsimile directly to the intended receipient. To do this create a shell script **etc/FaxDispatch** in the spooling area that sets *SENDTO* to the receiver's electronic mail address. For example,

```
case "$SENDER" in
*1*510*526*1212*) SENDTO=sam;;          # Sam's test rig in Berkeley
*1*415*390*1212*) SENDTO=raster@asd;;   # 7L Xerox room, used for scanning
*5107811212)      SENDTO=peebles@mti;;  # stuff from home
esac
case "$DEVICE" in
ttyS1)            SENDTO=john;;         # all faxes received on ttyS1
ttyLT0)           SENDTO=mary@home;;    # all faxes received on ttyLT0
esac
case "$CIDNUMBER" in
435*)        SENDTO=lee; FILETYPE=pdf;; # all faxes from area code 435
5059627777)  SENDTO=amy; FILETYPE=tif;; # Amy wants faxes in TIFF
esac
```

Note that you must match any embedded white space. The facsimile will be sent as a MIME-encoded PostScript document as default. The attachment filetype is controlled by the *FILETYPE* shell variable, and can be controlled as shown above. Current filetype options are ‘‘ps’’, ‘‘tif’’, and ‘‘pdf’’. Consult the shell script for details.

FILES

/var/spool/hylafax spooling area
/usr/sbin/faxinfo for printing information about the facsimile
/usr/bin/fax2ps for converting TIFF to PostScript
/usr/sbin/sendmail for delivering mail

SEE ALSO

faxd(8C), *hylafax-server*(5F), *recvq*(5F)

NAME

faxrm – remove a *HylaFAX* job or document from a server

SYNOPSIS

faxrm [**–a**] [**–d**] [**–v**] [**–h** *hostname*] *id...*

DESCRIPTION

faxrm removes one or more jobs or documents from a *HylaFAX* server. If a job that is to be removed is being processed by a server, it is first aborted.

An *id* is either a number that identifies a *job* (default) or, if the **–d** option is specified, a document pathname. Job identifiers are printed by *sendfax*(1) when jobs are queued for transmission; they can also be displayed with *faxstat*(1). The names of documents associated with outbound jobs can be seen by supplying the **–f** option to *faxstat*. Documents in the receive queue can be viewed with the **–r** option to *faxstat*.

Unprivileged users can only remove jobs or documents that they own. Clients with administrative privileges can remove any job or document (see the **–a** option below).

OPTIONS

–a Use administrative privileges; *faxrm* will prompt for the password to set administrative privileges.

–d Interpret each *id* as a document pathname. If a pathname does not begin with "/" then *faxrm* automatically prepends "/docq/".

–h *host* Contact the server at the specified *host*. The *host* may be either a symbolic name or a network address. If no server is specified then If no **–h** option is supplied, *faxrm* uses the FAXSERVER environment variable to identify the *HylaFAX* server to contact. If this variable is not set, then *faxrm* checks for a setting in the configuration files (first in the per-user file and then in the system-wide file). If all of the above fails, then *faxrm* attempts to contact a server on the machine where it is run.

–v Trace the protocol messages to and from the server.

CONFIGURATION PARAMETERS

faxrm reads configuration information from the file **/usr/share/fax/hyla.conf** and then from the file **~/.hylarc**. Configuration files follow the conventions described in *hylafax-client*(1). The following configuration parameters are recognized:

Tag	Type	Default	Description
Host	string	localhost	host to contact for service
Port	integer	4559	port to use in contacting server
Protocol	string	tcp	protocol to use in contacting server
Verbose	boolean	No	whether or not to enable protocol tracing

The configuration parameters are explained below:

Host The host to contact for service. (Equivalent to the **–h** option.)

Port The network port to contact for service. (Eqvuialent to the **–h** option.)

Protocol The name of the communication protocol to use when contacting a server. (Equivalent to the **FAXSERVICE** environment variable.)

Verbose Control protocol tracing. (Equivalent to the **–v** option.)

DIAGNOSTICS

Job %s removed. A confirmation that the job was successfully removed from the **sendq** directory.

Job %s removed (from doneq). A confirmation that the job was successfully removed from the **done** directory; the directory where jobs are placed once they are finished.

%s removed. A confirmation that the document was successfully deleted from the server.

SEE ALSO

hylafax-client(1), *sendfax*(1), *faxalter*(1), *faxstat*(1), *hfaxd*(8C)

NAME

faxsend − *HylaFAX* facsimile transmit program

SYNOPSIS

/usr/sbin/faxsend [**−l**] [**−p**] [**−c** *config*] **−m** *device-ID qfile*

DESCRIPTION

faxsend is the program that implements the facsimile transmission protocol for the *HylaFAX* system. *faxsend* is invoked by the *faxq* scheduler process for each facsimile transmit attempt. *faxsend* assumes that its current working directory is the top of the spooling hierarchy, that a suitable modem device has been allocated for its use, and that all documents to be sent are already converted to TIFF/F (TIFF Class F) at the appropriate resolution and with an appropriate data encoding. In normal use *faxsend* will inherit a lockfile for the modem from the *faxq* process. The format of the job description file specified on the command line is described in *sendq*(5F).

faxsend initializes the modem before attempting to transmit the facsimile using the information in the appropriate modem configuration file. If the modem does not respond to this initialization, *faxsend* will repeatedly try to initialize the modem. This behaviour is required for send-only environments in which a *faxgetty*(8C) process is not run (and the modem would be known to be in a properly initialized state). *faxsend* catches SIGINT and SIGTERM and cleans up any resources it controls; this is the mechanism used by *faxq* to abort a job while it is in progress.

Many aspects of *faxsend* operation are controlled through a configuration file. The configuration file is located in the spooler hierarchy in the **etc** subdirectory and is named **config.***device-ID***;** consult *hylafax-config*(5F) for detailed information on the contents of configuration files.

OPTIONS

The following options are recognized:

−c *config* Treat *config* as a configuration parameter specification that is interpreted after reading the per-device configuration file. For example, ‘‘-c sessiontracing:0x4f’’ would set the **SessionTracing** configuration parameter to ‘‘0x4f’’, overriding any setting in the configuration file.

−m *devID* The device to use in processing the job. Note that this is a *device identifier* and not the pathname of the tty special file. A device identifier is formed from a device filename by removing any leading ‘‘/dev/’’ and converting any ‘‘/’’ characters to ‘‘_’’ characters.

−l Do the UUCP lockfile protocol in *faxsend*. By default *faxsend* assumes that it is invoked with the device already locked and that it does not need to manage the lockfile.

−p Do not change the process priority when transmitting. Normally *faxsend* will raise its priority to reduce i/o latency. This option is useful for debugging.

NOTES

faxsend returns the new job status to *faxq* through the *exit*(2) return value of the process: *send_retry* (0), retry job; *send_failed* (1), job finished without success, *send_done* (2), job completed successfully, *send_reformat* (3), job should be retried after documents are reformatted.

FILES

/var/spool/hylafax/etc/config.devID device-specific configuration file

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

faxq(8C), *hylafax-server*(5F), *hylafax-config*(5F)

NAME

faxsetup − setup a machine for use with *HylaFAX*

SYNOPSIS

/usr/sbin/faxsetup [**−client**] [**−server**] [**−with-VAR***[=value]*]

DESCRIPTION

faxsetup is an interactive shell script that does the necessary setup work to configure a client or server machine for the *HylaFAX* software. *faxsetup* must be run on every *HylaFAX* server machine before the software will function−if a *HylaFAX* server program or script is invoked before *faxsetup* is run, the command will abort with a message that says to run *faxsetup*. *faxsetup* may optionally be run on a *HylaFAX* client machine to verify the configuration of the software.

By default *faxsetup* checks a machine for use both as a *HylaFAX* client and server. Client-specific checks include:

- the installation of client programs and data,
- the installation of the Adobe Font Metric (AFM) information files used by the *HylaFAX* text formatting support.

Server-specific checks include:

- the installation of server programs and data,
- the existence of the UUCP lock file directory,
- the installation of the TIFF software tools,
- the configuration of the getty programs invoked for data and voice calls,
- the pathnames for various system programs used by server shell scripts,
- proper support for FIFO special files,
- proper installation of a POSTSCRIPT RIP,
- definition of a ‘‘fax’’ user,
- definitions for the ‘‘hylafax’’ and ‘‘snpp’’ services,
- startup of the *HylaFAX* server processes: *faxq*(8C) and *hfaxd*(8C),
- definition of a ‘‘FaxMaster’’ mail alias for reporting server problems.

For server machines *faxsetup* also creates two files that are required by the *HylaFAX* command scripts. One file contains master configuration parameters and the other contains various definitions used by scripts such as *faxaddmodem*(8C) that communicate with modems.

OPTIONS

−client Do client-specific consistency checks.

−server Do server-specific consistency checks.

−with-VAR*[=value]*
Override the default definition for the configuration parameter named **VAR**; optionally specifying a new *value* (if no value is specified then ‘‘yes’’ is used). This option is especially useful to override the default settings for the POSTSCRIPT interpreter package to use (e.g. ‘‘-with-PS=dps’’), the spooling directory (‘‘-with-DIR_SPOOL=/mumble’’), the directory where the tool applications of TIFF software distribution are located (‘‘-with-TIFFBIN=/usr/freeware’’), etc. A complete list of the possible parameters can be found by looking at the contents of the *faxsetup* shell script; many of these configuration parameters are also described in the HTML documentation

−help Print a help message and terminate.

−quiet Do not print verbose messages to the terminal.

−verbose Print informative messages to the terminal (default).

BUGS

faxsetup does not query a modem to find out its capabilities when invoking the *faxmodem* program to setup a server in a send-only configuration; it may be necessary to run *faxmodem* manually.

FILES

/var/spool/hylafax/etc/setup.cache	master configuration parameters
/var/spool/hylafax/etc/setup.modem	modem-related configuration support

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

hylafax-server(5F), *hylafax-config*(5F), *faxq*(8C), *faxgetty*(8C), *faxmodem*(8C), *faxconfig*(8C), *faxsend*(8C)

NAME

faxstate − control ''modem state'' of a device used by *HylaFAX*

SYNOPSIS

/usr/sbin/faxstate [**−q** *queue-dir*] [**−s** *state*] [**−n**] *modem*

DESCRIPTION

faxstate sends a message to the *HylaFAX faxgetty*(8C) process servicing *modem* telling it to use the specified state when notifying the *HylaFAX* scheduler that a modem is ready and available for use. This is useful for controlling outbound use of a modem; by marking a modem's state as *busy* or *down* the *HylaFAX* scheduler will not assign any outbound jobs to the modem.

The specified *modem* can either be the name (typically the last component of the terminal port the modem is attached to), or the full name of the associated FIFO named pipe file, e.g. ''FIFO.ttym2''.

If the **−n** option, *faxstate* emulates what *faxgetty* would do; sending a message directly to the *faxq* process marking the specified modem down, busy, or ready. This interface is useful for send-only environments in which *faxgetty* processes are not used. Note that modems manipulated in this way must previously have been configured with the *faxmodem*(8C) program.

OPTIONS

The following options are supported:

−n Contact *faxq* directly instead of *faxgetty*.

−q *dir* Use a spooling area other than */var/spool/hylafax*.

−s Specify the state to send the *faxq* process each time the modem becomes ready for use; one of ''*busy*'', ''*down*'', and ''*ready*''. The default state to use is *ready*.

FILES

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

hylafax-server(5F), *faxgetty*(8C)

NAME

faxstat − display *HylaFAX* status

SYNOPSIS

faxstat [*options*]

DESCRIPTION

faxstat reports the status of *HylaFAX* servers, including (potentially) the status of the send, receive, done, document, and archive directories. When invoked without options *faxstat* reports only the status of the server; the options described below can be used to override this behaviour.

Server status information indicates the state of the server (idle, sending, receiving, etc.) and the phone number that is associated with the fax modem.

The job queue report give the status of facsimile jobs queued for transmission; by default:

- the job identifier,
- the job state (see below for the description of the **a** format in the **JobFmt** string),
- the modem or modem class the job is bound to (‘‘any’’ if the job can be sent though any modem),
- the destination (phone number) of the job,
- the time to send the job, or a blank field if the job is being processed or is to be sent immediately,
- the job’s owner,
- and status information about the job (typically the reason why a previous attempt failed).

The receive queue report lists the facsimile documents that have been received and are awaiting delivery; by default it includes:

- the document protection mode,
- the number of pages,
- the document’s owner,
- the phone number of the sender (if supplied),
- the date and time at which the job was received, and
- the filename of the document in the **recvq** directory on the server machine.

OPTIONS

−a Display the status of all jobs that are stored in the **archive** directory on the server machine (*NB: this will always be empty for v4.0 because there is no support for job archiving*).

−d Display the status of all jobs that have completed; i.e. those jobs located in the **doneq** directory on the server machine. The **JobFmt** string defines the content and format of information reported with this option.

−f Display the status of document files located in the **docq** directory on the server machine. The **FileFmt** string defines the content and format of information reported with this option (see below).

−g Display times and dates in Greenwich Mean Time (GMT). NB: *this is the default*.

−h *host* Report the status of the server on a specific *host*. The *host* may be either a symbolic name or a network address. If no **−h** option is supplied, *faxstat* uses the FAXSERVER environment variable to identify the *HylaFAX* server to contact. If this variable is not set, then *faxstat* checks for a setting in the configuration files (first in the per-user file and then in the system-wide file). If all of the above fails, then *faxstat* attempts to contact a server on the machine where it is run.

−i Display additional status information for the server. This status typically has information such as the *HylaFAX* version, the physical location of the server machine, and an administrative contact for the server.

–l Display times and dates in the local timezone of the server.

–r Display the receive queue status for each server. The **RcvFmt** string defines the content and format of information reported with this option (see below).

–s Display the status of jobs in the send queue on the server machine. The **JobFmt** string defines the content and format of information reported with this option (see below).

–v Trace the protocol exchanges between *faxstat* and the *hfaxd* processes on the standard output.

CONFIGURATION PARAMETERS

faxstat reads configuration information from the file **/usr/share/fax/hyla.conf** and then from the file **~/.hylarc**. Configuration files follow the conventions described in *hylafax-client*(1). The following configuration parameters are recognized:

Tag	**Type**	**Default**	**Description**
FileFmt	string	*see below*	format string for file status results
Host	string	localhost	host to contact for service
JobFmt	string	*see below*	format string for job status results
ModemFmt	string	*see below*	format string for modem status results
Port	integer	4559	port to use in contacting server
Protocol	string	tcp	protocol to use in contacting server
RcvFmt	string	*see below*	format string for received facsimile status results
TimeZone	string	GMT	timezone to use for reporting dates and times
Verbose	boolean	No	whether or not to enable protocol tracing

The configuration parameters are explained below:

FileFmt The format string to use when returning file status information with the **–f** option. Formats are specified using *printf*(3S)-style conventions but using the field identifiers listed below. Each item can include field width, precision, left-justification, 0-filling, etc. just as for *printf*; e.g. %-8p for an 8-character wide, left-justified, blank-padded field containing the file protection flags.

Format	**Description**
a	Last access time
c	Creation time
d	Device number (octal)
f	Filename
g	Group identifier (decimal)
i	Inode number (decimal)
l	Link count (decimal)
m	Last modification time
o	Owner (based on file GID)
p	Fax-style protection flags (no group bits)
q	UNIX-style protection flags
r	Root device number (octal)
s	File size in bytes (decimal)
u	User identifier (decimal)

Host The host to contact for service. (Equivalent to the **–h** option.)

JobFmt The format string to use when returning job status information for the **–s** and **–d** options. Formats are specified using *printf*(3S)-style conventions but using the field identifiers listed below. Each item can include field width, precision, left-justification, 0-filling, etc. just as for *printf*; e.g. %-3j for a 3-character wide, left-justified, blank-padded field containing the job state.

Format	**Description**
A	Destination SubAddress
B	Destination Password

C	Destination company name
D	Total # dials/maximum # dials
E	Desired signalling rate
F	Client-specific tagline format string
G	Desired min-scanline time
H	Desired data format
I	Client-specified scheduling priority
J	Client-specified job tag string
K	Desired use of ECM (one-character symbol)
L	Destination geographic location
M	Notification e-mail address
N	Desired use of private tagline (one-character symbol)
O	Whether to use continuation cover page (one-character symbol)
P	# pages transmitted/total # pages to transmit
Q	Client-specified minimum acceptable signalling rate
R	Destination person (receiver)
S	Sender's identity
T	Total # tries/maximum # tries
U	Page chopping threshold (inches)
V	Job done operation
W	Communication identifier
X	Job type (one-character symbol)
Y	Scheduled date and time
Z	Scheduled time in seconds since the UNIX epoch
a	Job state (one-character symbol)
b	# consecutive failed tries
c	Client machine name
d	Total # dials
e	Public (external) format of dialstring
f	# consecutive failed dials
g	Group identifier
h	Page chop handling (one-character symbol)
i	Current scheduling priority
j	Job identifier
k	Job kill time
l	Page length in mm
m	Assigned modem
n	E-mail notification handling (one-character symbol)
o	Job owner
p	# pages transmitted
q	Job retry time (MM::SS)
r	Document resolution in lines/inch
s	Job status information from last failure
t	Total # tries attempted
u	Maximum # tries
v	Client-specified dialstring
w	Page width in mm
x	Maximum # dials
y	Total # pages to transmit
z	Time to send job

The **K** format produces: ‘‘D’’ if ECM is to be disabled, ‘‘ ’’ (space) if ECM use is enabled, ‘‘H’’ if T.30 Annex C half duplex is enabled, or ‘‘F’’ if T.30 Annex C full duplex is enabled.

The **N** format produces: “ ” (space) if the system-wide tagline format is to be used or “P” if a private tagline format is to be used.

The **O** format produces: “N” if no continuation cover page is to be used or “ ” (space) if the system default handling for continuation cover pages is to be used.

The **X** format produces: “F” for a facsimile job or “P” for a pager job.

The **a** format produces: “?” for a job in an undefined state, “T” for a suspended job (not being scheduled), “P” for a pending job (waiting for its time to send to arrive), “S” for a sleeping job (waiting for a scheduled timeout such as a delay between attempts to send), “B” for a job blocked by concurrent activity to the same destination, “W” for a job waiting for resources such as a free modem, “R” for a job that is actively running, and “D” for a job that is done.

The **h** format produces: “D” if page chopping is disabled, “ ” (space) for the system default page chop handling, “A” when all pages are to be chopped, or “L” if only the last page is to be chopped.

The **n** format produces: “ ” (space) when no notification messages are to be delivered, “D” when notification is to be sent when the job is done, “Q” when notification is to be sent each time the job is requeued, or “A” when notification is to be sent for either the job completing or being requeued.

It is recommended that all items include a field width so that the width of column title strings can be constrained when constructing headers from the format string.

ModemFmt The format string to use when returning modem status information. Formats are specified using *printf*(3S)-style conventions but using the field identifiers listed below. Each item can include field width, precision, left-justification, 0-filling, etc. just as for *printf*; e.g. %-8h for an 8-character wide, left-justified, blank-padded field containing the name of the host the server is running on.

Format	**Description**
h	Server hostname
l	Local identifier string
m	Canonical modem name
n	FAX phone number
r	Maximum pages that can be received in a single call
s	Status information string
t	Server and session tracing levels (xxxxx:yyyyy)
v	Modem speaker volume as one-character symbol
z	A “*” if a *faxgetty*(8C) process is running; otherwise “ ” (space)

Port The network port to contact for service. (Eqvuialent to the **–h** option.)

Protocol The name of the communication protocol to use when contacting a server. (Equivalent to the **FAXSERVICE** environment variable.)

RcvFmt The format string to use when returning status information for the **–r** option. Formats are specified using *printf*(3S)-style conventions but using the field identifiers listed below. Each item can include field width, precision, left-justification, 0-filling, etc. just as for *printf*; e.g. %-3b for a 3-character wide, left-justified, blank-padded field containing the signalling rate.

Format	**Description**
a	SubAddress received from sender (if any)
b	Signalling rate used during receive
d	Data format used during receive
e	Error description if an error occurred during receive
f	Document filename (relative to the **recvq** directory)
h	Time spent receiving document (HH:MM:SS)

l	Page length in mm
m	Fax-style protection mode string (“-rwxrwx”)
n	File size (number of bytes)
o	File owner
p	Number of pages in document
q	UNIX-style protection flags
r	Resolution of received data
s	Sender identity (TSI)
t	Compact representation of the time when the receive happened
w	Page width in mm
z	A “*” if receive is going on; otherwise “ ” (space)

It is recommended that all items include a field width so that the width of column title strings can be constrained when constructing headers from the format string.

TimeZone Control whether times and dates are reported in the local timezone of the server (“local”) or in GMT (“gmt”). By default times are reported in GMT.

Verbose Control protocol tracing. (Equivalent to the **−v** option.)

BUGS

There is no way to suppress the server status report. There is no way to control the printing of headers.

FILES

/usr/share/fax/hyla.conf	system-wide configuration file
~/.hylarc	per-user configuration file

SEE ALSO

hylafax-client(1), *sendfax*(1), *faxalter*(1), *faxrm*(1), *hfaxd*(8C)

NAME

faxwatch − monitor low-level *HylaFAX* server actions

SYNOPSIS

/usr/sbin/faxwatch [**−g**] [**−l**] [**−h** *host*] *trigger-expression*

DESCRIPTION

faxwatch registers interest in one or more events on a *HylaFAX* server and then prints subsequent event messages to the standard output. The *faxwatch* program is intended mainly for testing the event distribution mechanism used by the *HylaFAX* servers.

The *trigger-expression* is supplied as an argument to a ‘‘SITE TRIGGER’’ command passed to the *hfaxd*(8C) The syntax for this expression is:

[<*class*>['<'*id*'>'][<*mask*>|'*']]*

where <*class*> defines a class of events and is one of:

J for job-related events,

S for fax send-related events,

R for fax receive-related events, and

M for modem-related events.

A <*mask*> is a 4-hex-digit mask of trigger events (see the table below). If ‘‘*’’ is specified then all events in the class are matched.

An <*id*> can be used to restrict matches to a specific job or modem. Eventually this will need to be generalized for job groups.

Thus an example specification that would catch any event for the modem on ttyf2 would be ‘‘M<ttyf2>*’’, and to be notified when job 1932 is requeued or completes one would use ‘‘J<1932>4c60’’.

The output from *faxwatch* is ASCII text that describes each event. *faxwatch* runs until it is interrupted.

OPTIONS

−g Display times and dates in Greenwich Mean Time (GMT). NB: *this is the default*.

−h *host* Report the status of the server on a specific *host*. The *host* may be either a symbolic name or a network address. If no **−h** option is supplied, *faxwatch* uses the FAXSERVER environment variable to identify the *HylaFAX* server to contact. If this variable is not set, then *faxwatch* checks for a setting in the configuration files (first in the per-user file and then in the system-wide file). If all of the above fails, then *faxwatch* attempts to contact a server on the machine where it is run.

−l Display times and dates in the local timezone of the server.

−v Trace the protocol exchanges between *faxwatch* and the *hfaxd* processes on the standard output.

EVENTS

The following table specifies the current set of events that may be monitored. This information is subject to change; consult the source code for reference.

Event	**Class**	**Mask**	**Description**
JOB_CREATE	J	0x0001	job created
JOB_SUSPEND	J	0x0002	job suspended
JOB_READY	J	0x0004	job ready to send
JOB_SLEEP	J	0x0008	job sleeping awaiting time-to-send
JOB_DEAD	J	0x0010	job marked dead
JOB_PROCESS	J	0x0020	job processed by scheduler
JOB_REAP	J	0x0040	job corpus reaped
JOB_ACTIVE	J	0x0080	job activated

JOB_REJECT	J	0x0100	job rejected
JOB_KILL	J	0x0200	job killed
JOB_BLOCKED	J	0x0400	job blocked by other job
JOB_DELAYED	J	0x0800	job delayed by tod restriction or similar
JOB_ALTERED[1]	J	0x1000	job parameters altered
JOB_TIMEDOUT	J	0x2000	job kill timer expired
JOB_PREP_BEGIN	J	0x4000	job preparation started
JOB_PREP_END	J	0x8000	job preparation finished
SEND_BEGIN	S	0x0001	fax, send attempt started
SEND_CALL	S	0x0002	fax, call placed
SEND_CONNECTED	S	0x0004	fax, call answered by fax
SEND_PAGE	S	0x0008	fax, page transmit done
SEND_DOC	S	0x0010	fax, document transmit done
SEND_POLLRCVD	S	0x0020	fax, document retrieved by poll operation
SEND_POLLDONE	S	0x0040	fax, poll operation completed
SEND_END	S	0x0080	fax, send attempt finished
SEND_REFORMAT	S	0x0100	fax, job being reformatted
SEND_REQUEUE	S	0x0200	fax, job requeued
SEND_DONE	S	0x0400	fax, send job done
RECV_BEGIN	R	0x0001	fax, inbound call started
RECV_START	R	0x0002	fax, session started
RECV_PAGE	R	0x0004	fax, page receive done
RECV_DOC	R	0x0008	fax, document receive done
RECV_END	R	0x0010	fax, inbound call finished
MODEM_ASSIGN	M	0x0001	modem assigned to job
MODEM_RELEASE	M	0x0002	modem released by job
MODEM_DOWN	M	0x0004	modem marked down
MODEM_READY	M	0x0008	modem marked ready
MODEM_BUSY	M	0x0010	modem marked busy
MODEM_WEDGED	M	0x0020	modem considered wedged
MODEM_INUSE	M	0x0040	modem inuse for outbound work
MODEM_DATA_BEGIN	M	0x0080	inbound data call begun
MODEM_DATA_END	M	0x0100	inbound data call finished
MODEM_VOICE_BEGIN	M	0x0200	inbound voice call begun
MODEM_VOICE_END	M	0x0400	inbound voice call finished
MODEM_CID	M	0x0800	inbound caller-ID information

[1] Event to be removed soon.

SEE ALSO

hylafax-server(5F), *hfaxd*(8C).

NAME

hfaxd − *HylaFAX* client-server protocol server

SYNOPSIS

/usr/sbin/hfaxd [**−d**] [**−q** *dir*] [**−o** *port*] [**−O**] [**−i** *port*] [**−I**] [**−s** *port*] [**−S**]

DESCRIPTION

hfaxd is the *HylaFAX* program that implements the server portion of:

- the (new) client-server protocol,
- the old client-server protocol previously implemented by the *faxd.recv* program, and
- the Simple Network Paging Protocol (SNPP) used to submit message pager jobs to the IXO/TAP and UCP support.

Additional client-server protocols are planned and *hfaxd* is intended to be the ‘‘carrier’’ through which they are supported.

hfaxd is typically used in one of two ways; either as a standalone process that is started at system boot time to listen for client connections on one or more ports, or as a subservient process to the *inetd*(8C) program. The two forms of use may however be combined so long as the same service is not provided both by the standalone *hfaxd* and through *inetd*.

HYLAFAX (NEW) CLIENT-SERVER PROTOCOL SUPPORT

If *hfaxd* is started with the **−i** option it will service clients using the (new) *HylaFAX* client-server protocol. This protocol is strongly related to the Internet File Transfer Protocol (FTP); so much so in fact that FTP client programs that include support for ‘‘quoted commands’’ may be used to communicate with *hfaxd* using the new protocol. (It should also be possible to use FTP-aware World Wide Web browsers such as Mosaic and Netscape Navigator to access *HylaFAX* servers through the new protocol; but the current format for information returned in directory listings confuses them.)

The *hfaxd* server currently recognizes the following protocol requests; case is not distinguished. Entries marked with a [1] can be used only when the client has established administrative privileges with ADMIN.

Request	**Description**
ABOR	abort previous command
ACCT	specify account (ignored)
ADMIN	specify password for administrative privileges
ALLO	allocate storage (vacuously)
ANSWER[1]	request that call be answered
APPE	append to a file
CDUP	change to parent of current working directory
CHMOD	change mode of a file
CHOWN[1]	change owner of a file
CWD	change working directory
DELE	delete a file
DISABLE[1]	disable outbound use of modem
ENABLE[1]	enable outbound use of modem
HELP	give help information
FILEFMT	specify/query format for returning file status
FORM	specify data transfer *format*
IDLE	set idle-timer (in seconds)
JDELE	delete done or suspended job
JINTR	interrupt job
JKILL	kill job
JNEW	create new job
JOB	set/query current job
JOBFMT	specify/query format for returning job status
JPARM	specify/query job state parameter

JREST	reset current job state
JSUBM	submit job to scheduler
JSUSP	suspend job from scheduling
JWAIT	wait for job to complete
JGDELE	delete group of jobs
JGKILL	kill group of jobs
JGINTR	interrupt group of jobs
JGNEW	place current job in a new job group
JGPARM	set state parameter in a group of jobs
JGREST	reset current state for a group of jobs
JGRP	set/query current job group
JGSUBM	submit group of jobs to scheduler
JGSUSP	suspend group of jobs from scheduling
JGWAIT	wait for group of jobs to complete
LIST	list files in a directory
MDTM	show last modification time of file
MODE	specify data transfer *mode*
MDMFMT	specify/query format for returning modem status
NLST	give name list of files in directory
NOOP	do nothing
PASS	specify password
PASV	prepare for server-to-server transfer
PORT	specify data connection port
PWD	print the current working directory
QUIT	terminate session
RCVFMT	specify/query format for returing received facsimile status
REIN	reinitiate server state
REST	restart incomplete transfer
RETP	retrieve the next page in a file
RETR	retrieve a file
SHUT	schedule server shutdown
SITE	non-standard commands (see next section)
SIZE	return size of file
STAT	return status of server or file
STOR	store a file
STOT	store a temporary file with a unique name
STOU	store a file with a unique name
STRU	specify data transfer *structure*
SYST	show operating system type of server system
TYPE	specify data transfer *type*
TZONE	specify timezone handling for dates and times
USER	specify user name
VRFY	verify dialstring handling and/or least-cost routing

The following non-standard or experimental commands are supported through the SITE request.

Request	**Description**
ADDMODEM[1]	add/configure new modem for use
ADDUSER[1]	add client access control entry
CONFIG[1]	send configuration parameter setting to server
DELMODEM[1]	deconfigure/remove modem
DELUSER[1]	remove client access control entry
TRIGGER	register realtime event trigger
HELP	give help information, e.g., SITE HELP

In addition FTP requests that are specified in Internet RFC 959 but not listed here are recognized, but not implemented.

The *hfaxd* server will abort an active data transfer only when the ABOR command is preceded by a Telnet "Interrupt Process" (IP) signal and a Telnet "Synch" signal in the command Telnet stream, as described in Internet RFC 959. If a STAT command is received during a data transfer, preceded by a Telnet IP and Synch, transfer status will be returned.

OLD CLIENT-SERVER PROTOCOL SUPPORT

If *hfaxd* is started with the **−o** option it will service clients using the old *HylaFAX* client-server protocol that was used in distributions prior to the 4.0 release. Note however that this support is only available if *hfaxd* is compiled with the support enabled.

Emulation of the old protocol is important for supporting non-UNIX clients such as the WinFlex client for Windows systems and the MacFlex client for Macintosh systems. It is strongly recommended however that unless you need to support these old-style clients that you not enable support for the old protocol because of the inherent misdesign of this protocol.

SIMPLE NETWORK PAGING PROTOCOL (SNPP) SUPPORT

If *hfaxd* is started with the **−s** option it will service clients using the Simple Network Pager Protocol (SNPP) as specified in RFC 1861 (which obsoletes RFC 1645).

The *hfaxd* server currently recognizes the SNPP protocol requests listed below. Requests marked with a [1] are non-standard extensions to RFC 1861 that may be added to SNPP at some future time. Case is not distinguished and only the first four characters of requests are used in identifying commands.

Request	Description
2WAY	preface a 2-way transaction
ABOR[1]	abort previous command
ACKR	set read acknowledgement handling for subsequent requests
ALER	set the alter-level for subsequent requests
CALL	set the caller-ID for subsequent requests
COVE	set the alternate coverage area for subsequent requests
DATA	specify a multi-line message
EXPT	set the expiration time for subsequent requests
HELP	give help information
HOLD	set the time at which subsequent requests are to be delivered
KTAG	kill a previously submitted request
LEVE	set the service level for subsequent requests
LOGI	login to server
MCRE	specify multiple response text and code
MESS	specify a single-line message
MSTA	return the status of a previously submitted request
NOQU	diable message queueing
PAGE	specify the destination pager
PING	locate/validate a pager
QUIT	terminate session
RESE	reset server state
RTYP	set the reply type code for subsequent requests
SEND	send message(s)
SITE[1]	site-specific commands (see next section)
STAT[1]	return server status
SUBJ	set the message text for subsequent requests

The *hfaxd* server will abort an active SEND operation when an ABOR command is preceded by a Telnet "Interrupt Process" (IP) signal and a Telnet "Synch" signal in the command Telnet stream.

The following non-standard or experimental commands are also supported through the SITE request.

Request	Description
FROMUSER	specify the sender's identity

IDLE	set idle-timer (in seconds)
JPARM	query job parameter status
JQUEUE	control whether or not job is queued
LASTTIME	set the time to terminate an unfinished job
MAILADDR	set the e-mail address to use for notification
MAXDIALS	set the maximum number of times to dial the phone
MAXTRIES	set the maximum number of times to try sending the page
MODEM	set the modem or class of modems to use
NOTIFY	set the e-mail notification
RETRYTIME	set the time to delay between job retries
SCHEDPRI	set the scheduling priority for the job
HELP	give help information, e.g., SITE HELP

Note that *hfaxd* requires that SNPP clients login first with the LOGI directive while RFC 1861 permits clients to submit pages anonymously.

CLIENT ACCESS CONTROL

hfaxd controls client access according to the information in the file **/var/spool/hylafax/etc/hosts.hfaxd**. This file defines the set of users and machines that may receive service and, optionally, defines password challenges to use in authenticating clients. Clients may be permitted access to services with or without a password challenge. Access may also be restricted based on the host machine that a request for service originates from. Consult *hosts.hfaxd*(5F) for information on the format and content of this file. The SITE ADDUSER protocol request is provided for adding new users to a server (available only to clients with administrative privileges).

Server resources are controlled based on clients' identities. Jobs and documents, both received and submitted, are protected by the server. Typically clients are permitted access to anything they own or that is publicly accessible. There are also administrative privileges that clients may acquire and which permit them wide access to objects that reside on the server.

A complete client-server protocol specification is still outstanding.

hfaxd operates with its root directory set to the top of the *HylaFAX* spooling area; /var/spool/hylafax. This is done so that clients see a virtual file hierarchy that is completely contained within the *HylaFAX* operating environment on the server machine. Administrators however must be aware of this action when specifying files in the *hfaxd* configuration file: absolute pathnames relative to the root of the spooling should be used to specify filenames.

SERVER ACCESS CONTROL

The file **/var/spool/hylafax/etc/shutdown**, when present, specifies when to restrict access to a server machine. When this file is present and contains valid information *hfaxd* will permit only users with administrative privileges to access the server. Any other users that request service will be denied access and negative server responses will include any shutdown message specified in the shutdown file. Consult *hylafax-shutdown*(5F) for information on the format and content of this file.

The SHUT protocol request can be used to schedule a server shutdown; it is available only to clients with administrative privileges. To make a shutdown server available again the shutdown file can be deleted with the DELE protocol request (*this is to be replaced with an "unshut" protocol request so that implementation details are not part of the protocol*).

CONFIGURATION FILES

hfaxd reads configuration information from the file **/usr/share/fax/hfaxd.conf** each time a new server process is started (i.e. for each new client). This file uses the same conventions used by other *HylaFAX* configuration files; as described in *hylafax-config*(5F). The following configuration parameters are recognized; items marked "(SNPP)" are used only by the SNPP support.

Tag	**Type**	**Default**	**Description**
FaxContact	string	*see below*	contact address to show in help text
FileFmt	string	*see below*	format string for file status results
IdleTimeout	integer	900	client idle timeout in seconds

JobFmt	string	*see below*	format string for job status results
KillTimeMap	string	*see below*	mapping from service level to job kill time (SNPP)
LogFacility	string	daemon	*syslog* facility name for tracing messages
MaxAdminAttempts	integer	5	maximum admin attempts before disconnecting
MaxConsecutiveBadCmds	integer	10	maximum invalid commands before disconnecting
MaxIdleTimeout	integer	7200	maximum client idle timeout permitted
MaxLoginAttempts	integer	5	maximum login attempts before disconnecting
MaxMsgLength	integer	128	maximum pager message length (SNPP)
ModemFmt	string	*see below*	format string for modem status results
PagerIDMapFile	string	/var/spool/hylafax/etc/pagermap	name of file for mapping pager IDs (SNPP)
PriorityMap	string	*see below*	mapping from service level to job priority (SNPP)
RcvFmt	string	*see below*	format string for received facsimile status results
RetryTimeMap	string	*see below*	mapping from service level to job retry time (SNPP)
ServerTracing	integer	1	server tracing control vector
ShutdownFile	string	/var/spool/hylafax/etc/shutdown	name of shutdown control file
UserAccessFile	string	/var/spool/hylafax/etc/hosts.hfaxd	name of access control file
XferLogFile	string	/var/spool/hylafax/etc/clientlog	name of file for logging client data transfers

The configuration parameters are explained below:

FaxContact
The e-mail address to display as a point of contact in the help text returned to a client in response to the HELP or SITE HELP commands. By default this is ‘‘FaxMaster@*hostname*’’, where *hostname* is the fully qualified name for the machine where the server is running.

FileFmt The format string to use when returning file status information with the LIST and STAT commands. Formats are specified using *printf*(3S) style conventions but using the field identifiers listed below. Each item can include field width, precision, left-justification, 0-filling, etc. just as for *printf*; e.g. %-8p for an 8-character wide, left-justified, blank-padded field containing the file protection flags.

Format	**Description**
a	Last access time
c	Creation time
d	Device number (octal)
f	Filename
g	Group identifier (decimal)
i	Inode number (decimal)
l	Link count (decimal)
m	Last modification time
o	Owner (based on file GID)
p	Fax-style protection flags (no group bits)
q	UNIX-style protection flags
r	Root device number (octal)
s	File size in bytes (decimal)
u	User identifier (decimal)

The default format string is ‘‘%-7p %3l %8o %8s %-12.12m %.48f’’. It is recommended that all items include a field width so that client applications that construct headers from the format string can constrain the width of column title strings.

IdleTimeout
The initial/default timeout to use in timing out idle clients. This value defines the maximum amount of time (in seconds) that *hfaxd* will wait for a command from a client before terminating the connection. Unprivileged clients may alter the idle timeout up to the value of *MaxIdleTimeout*; privileged clients may set the timeout to any value.

JobFmt The format string to use when returning job status information for jobs in the **sendq** and **doneq** directories. Formats are specified using *printf*(3S) style conventions but using the field identifiers listed below. Each item can include field width, precision, left-justification, 0-filling, etc. just as for *printf*; e.g. %-3j for a 3-character wide, left-justified, blank-padded field containing the job state.

Format	Description
A	Destination SubAddress
B	Destination Password
C	Destination company name
D	Total # dials/maximum # dials
E	Desired signalling rate
F	Client-specific tagline format string
G	Desired min-scanline time
H	Desired data format
I	Client-specified scheduling priority
J	Client-specified job tag string
K	Desired use of ECM
L	Destination geographic location
M	Notification e-mail address
N	Desired use of private tagline (one-character symbol)
O	Whether to use continuation cover page (one-character symbol)
P	# pages transmitted/total # pages to transmit
Q	Client-specified minimum acceptable signalling rate
R	Destination person (receiver)
S	Sender's identity
T	Total # tries/maximum # tries
U	Page chopping threshold (inches)
V	Job done operation
W	Communication identifier
X	Job type (one-character symbol)
a	Job state (one-character symbol)
b	# consecutive failed tries
c	Client machine name
d	Total # dials
e	Public (external) format of dialstring
f	# consecutive failed dials
g	Group identifier
h	Page chop handling (one-character symbol)
i	Current scheduling priority
j	Job identifier
k	Job kill time
l	Page length in mm
m	Assigned modem
n	E-mail notification handling (one-character symbol)
o	Job owner
p	# pages transmitted
q	Job retry time (MM::SS)
r	Document resolution in lines/inch
s	Job status information from last failure
t	Total # tries attempted
u	Maximum # tries
v	Client-specified dialstring
w	Page width in mm
x	Maximum # dials
y	Total # pages to transmit
z	Time to send job

The default format string is ‘‘%−4j %3i %1a %6.6o %−12.12e %5P %5D %7z %.25s’’. This string constrains each status line to be less than 80 characters. It is recommended that all items include a field width so that client applications, such as *faxstat*(1) that construct headers from the format string can constrain the width of column title strings.

KillTimeMap

The mapping from SNPP service level (0-11) to job expiration time (kill time). A mapping is specified as a string of space-separate numbers where each value is the number of minutes to permit a job to run. The default mapping is ‘‘5 5 5 15 60 240 720 1440 1440 1440 1440 1440’’ which expires a job in 5 minutes for service levels 0-2, 15 minutes for level three, 60 minutes for level four, etc.

LogFacility

The symbolic name for the *syslog*(3) facility to use when logging error messages and informational/debugging messages requested through the **ServerTracing** parameter. The list of facility names is found in the system include file **<syslog.h>**; comparisons are case-insensitive.

MaxAdminAttempts

The maximum number of unsuccessful attempts gain adminstrative privileges with the ADMIN command that *hfaxd* will permit a client before terminating the connection. Note that the count of attempts is reset if/when the client successfully gains administrative privileges.

MaxConsecutiveBadCmds

The maximum number of consecutive unrecognized, unimplemented, syntactically incorrect, or otherwise unacceptable commands to permit a client before terminating the connection. This control has two purposes: to handle naive or malicious clients from sending long streams of nonsense commands to a server, and to insure that clients are forcibly terminated when a server is marked shutdown.

MaxIdleTimeout

The maximum value that a client may set the idle timeout to. This value is not enforced if the client has administrative privileges.

MaxLoginAttempts

The maximum number of unsuccessful attempts to login with the USER and PASS commands that *hfaxd* will permit a client before terminating the connection.

MaxMsgLength

The maximum number of characters to accept in a pager message specified with the DATA or MESS commands. Messages longer than this value are rejected.

ModemFmt

The format string to use when returning modem status information for modems listed in the **status** directory. Formats are specified using *printf*(3S) style conventions but using the field identifiers listed below. Each item can include field width, precision, left-justification, 0-filling, etc. just as for *printf*; e.g. %-8h for an 8-character wide, left-justified, blank-padded field containing the name of the host the server is running on.

Format	**Description**
h	Server hostname
l	Local identifier string
m	Canonical modem name
n	FAX phone number
r	Maximum pages that can be received in a single call
s	Status information string
t	Server and session tracing levels (xxxxx:yyyyy)
v	Modem speaker volume as one-character symbol
z	A ‘‘*’’ if a *faxgetty* (8C) process is running; otherwise ‘‘ ’’ (space)

The default format string is ‘‘Modem %m (%n): %s’’.

PagerIDMapFile
The absolute pathname of the file that contains directions for mapping pager identifiers to IXO/TAP or UCP service providers (and optionally a pager identification number). Consult *pagermap*(5F) for information on the format and content of this file. (Note that absolute pathnames are relative to the root of the spooling area).

PriorityMap
The mapping from SNPP service level (0-11) to job scheduling priority. A mapping is specified as a string of space-separate numbers where each value is the priority to assign to a job. The default mapping is ‘‘63 127 127 127 127 127 127 127 127 127 127 127’’ which assigns a high priority to service level zero and normal (default) priority to all other service levels.

RcvFmt The format string to use when returning status information for received facsimile in the **recvq** directory. Formats are specified using *printf*(3S) style conventions but using the field identifiers listed below. Each item can include field width, precision, left-justification, 0-filling, etc. just as for *printf*; e.g. %-3b for a 3-character wide, left-justified, blank-padded field containing the signalling rate.

Format	**Description**
a	SubAddress received from sender (if any)
b	Signalling rate used during receive
d	Data format used during receive
e	Error description if an error occurred during receive
f	Document filename (relative to the **recvq** directory)
h	Time spent receiving document (HH:MM:SS)
l	Page length in mm
m	Fax-style protection mode string (‘‘-rwxrwx’’)
n	File size (number of bytes)
o	File owner
p	Number of pages in document
q	UNIX-style protection flags
r	Resolution of received data
s	Sender identity (TSI)
t	Compact representation of the time when the receive happened
w	Page width in mm
z	A ‘‘*’’ if receive is going on; otherwise ‘‘ ’’ (space)

The default format string is ‘‘%−7m %4p%1z %−8.8o %14.14s %7t %f’’. This string constrains each status line to be less than 80 characters. It is recommended that all items include a field width so that client applications, such as *faxstat*(1) that construct headers from the format string can constrain the width of column title strings.

RetryTimeMap
The mapping from SNPP service level (0-11) to job retry time. A mapping is specified as a string of space-separate numbers where each value is the number of seconds to delay between delivery attempts. A value of zero causes retries to be scheduled using the default algorithm used by the *HylaFAX* job scheduler. The default mapping is ‘‘30 60 60 180 0 0 0 0 0 0 0 0’’ which retries a level 0 job after a 30 second delay, levels 1 and 2 after 60 seconds, level 3 after 3 minutes, and other jobs are retried according to the usual scheduling algorithm.

ServerTracing
A number that controls the generation of tracing information by a server. *areas* that are individually controlled. To enable tracing of multiple areas of operation, the flag associated with

each area should be bit-or'd to form the value for this tag.

Flag	Description
1 (0x00001)	General server operation
2 (0x00002)	Client-server protocol requests and responses
4 (0x00004)	File transfers from client to server
8 (0x00008)	File transfers from server to client
16 (0x00010)	Client logins
32 (0x00020)	All network connections
64 (0x00040)	FIFO messages to and from *faxq* (8C)
128 (0x00080)	TIFF Library errors and warnings
256 (0x00100)	Configuration file processing

Tracing messages are directed to *syslog*(3) using the *facility* specified with the **LogFacility** configuration parameter. Note that *syslogd*(8C) must be configured to capture *facility.info*, *facility.debug*, *facility.warning*, and *facility.err* messages.

ShutdownFile
The absolute pathname of the server shutdown file; see *hylafax-shutdown*(5F) for information on the format and content of this file. (Note that absolute pathnames are relative to the root of the spooling area).

UserAccessFile
The absolute pathname of the user access control file; see *hosts.hfaxd*(5F) for information on the format and content of this file. (Note that absolute pathnames are relative to the root of the spooling area).

XferLogFile
The absolute pathname of the file to use for logging client-server file transfers (when enabled through the **ServerTracing** parameter). (Note that absolute pathnames are relative to the root of the spooling area).

OPTIONS

−q *dir* The specified directory is treated as the spooling area. The default spooling area, */var/spool/hylafax*, is defined at the time the software is built.

−d Stop *hfaxd* from *detaching* itself from the controlling terminal. This option is normally used only when running *hfaxd* under a debugger or when *hfaxd* is started up from the *inetd*(8C) process.

−i *port* Listen on the specified *port* for service requests and respond with the (new) client-server protocol. The port may be specified either symbolically, e.g. ‘‘hylafax’’ or numerically. This flag may be specified multiple times to request service on multiple different ports.

−o *port* Listen on the specified *port* for service requests and respond with the old client-server protocol. The port may be specified either symbolically, e.g. ‘‘fax’’ or numerically. This flag may be specified multiple times to request service on multiple different ports.

−s *port* Listen on the specified *port* for service requests and respond with the Simple Network Paging (SNPP) protocol. The port may be specified either symbolically, e.g. ‘‘snpp’’ or numerically. This flag may be specified multiple times to request service on multiple different ports.

−I Service the (new) client-server protocol using the standard input and output. This option is useful when *hfaxd* is started up by *inetd*(8C).

−O Service the old client-server protocol using the standard input and output. This option is useful when *hfaxd* is started up by *inetd*(8C).

−S Service the Simple Network Paging (SNPP) protocol using the standard input and output. This option is useful when *hfaxd* is started up by *inetd*(8C).

DIAGNOSTICS

Diagnostics generated by *hfaxd* are logged using *syslog*(3).

FILES

/usr/share/fax/hfaxd.conf	server configuration file
/var/spool/hylafax	spooling area
/var/spool/hylafax/FIFO	for submitting the job
/var/spool/hylafax/sendq	where job description is placed
/var/spool/hylafax/sendq/seqf	for assigning job identifiers
/var/spool/hylafax/docq/seqf	for assigning document identifiers
/var/spool/hylafax/tmp	temporary location of job-related files
/var/spool/hylafax/docq	where document files are placed
/var/spool/hylafax/recvq	where received facsimile are found
/var/spool/hylafax/archive	where archived jobs are placed
/var/spool/hylafax/log	for server log files
/var/spool/hylafax/client	for FIFO files used in communicating with *faxq*
/var/spool/hylafax/status	for server status information
/var/spool/hylafax/config.*device*	for returning server status
/var/spool/hylafax/etc/hosts.hfaxd	host access control list
/var/spool/hylafax/etc/shutdown	server shutdown control
/var/spool/hylafax/etc/pagermap	SNPP pager ID mapping file

BUGS

To be filled in.

SEE ALSO

sendfax(1), *sendpage*(1), *faxalter*(1), *faxrm*(1), *faxstat*(1), *faxq*(8C), *syslog*(3) *hylafax-server*(5F), *hosts.hfaxd*(5F), *status*(5F), *hylafax-shutdown*(5F),

NAME

hosts.hfaxd − *HylaFAX* client access control list

DESCRIPTION

The ASCII file **etc/hosts.hfaxd** in the *HylaFAX* spooling area specifies the hosts and users that are permitted to access services through the *hfaxd*(8C) process. This file must exist for client access; if it is not present then *hfaxd* will deny all requests for service. Note also that this file must be readable only by the ‘‘fax’’ user; i.e. it should have mode 600 and be owned by ‘‘fax’’.

Each newline-terminated entry is a set of colon (:) separated fields, all but the first of which are optional. Trailing null fields and their separators may be omitted. The most general form is:

client:uid:passwd:adminwd

client is a regular expression to be matched against a string ‘‘*user@host*’’ that is formed from the *user* string passed to *hfaxd* with the **USER** command and the official *host* name or the DARPA Internet address, specified in ‘‘dot notation’’. If *client* does not contain an ‘‘@’’ then, for backwards compatibility, it is treated as a host for which any user may have access; i.e. it is automatically converted to the regular expression ‘‘^.*@*client*$’’.

Comments are introduced with the ‘‘#’’ character and extend to the end of the line. Any whitespace immediately preceding a comment is also ignored.

If *client* has a leading ‘‘!’’, then it is interpreted as a class of hosts and users to which access is to be **disallowed**. That is, if the pattern matches the client information, then access is denied.

Note that regular expressions are **not** anchored. That is, a regular expression may match a substring of the ‘‘*user@host*’’ string. Thus ‘*pb@.*\cl\.cam\.ac\.uk*’ matches ‘*cpb@mc.cl.cam.ac.uk.esd.sgi.com*’. Use ‘‘^’’ to match the start of the string and ‘‘$’’ to match the end.

Fields following *client* are optional and specify the following:

uid The numerical user ID to assign to clients that use the entry for access. *hfaxd* uses the *uid* to control access to server resources such as jobs and documents (the value is used to set the group ID of files created by a client).

Multiple clients/users may share the same *uid* or unique IDs may be created for each client. User IDs may be any number in the range [0..6002] with 6002 used, by convention, for entries that do not have a *uid* specified.

passwd The encrypted password. If this field is empty (null) then no password will be demanded when a client logs in; i.e. the **USER** command does not need to be followed by a **PASS** command.

adminwd The encrypted password for this user to gain administrative privileges. If this field is empty (null) then the user is not permitted to have administrative privileges.

EXAMPLE

The following is a sample hosts.hfaxd file. Note that the first entry that matches is taken, so more-specific entries should be placed first.

```
^pb@[^.]*\.cl\.cam\.ac\.uk$:::hFy8zXq2KaG8s
                        # pb on a machine directly in cl.cam.ac.uk can
                        # administer if an admin pw is given
127.0.0.1               # anyone on local host uses the default uid
^sam@flake.*sgi\.com$   # Sam on his work machine
^sam@oxford.*Berkeley.*# Sam on any machine starting oxford and containing
                        # Berkeley, e.g. sam@oxfordberkeley.cl.cam.ac.uk
^.*@.*.\.esd\.          # anyone in an esd domain
!^tom@                  # Tom Davis is denied access
.*\.sgi\.com$           # but anyone else at sgi is ok
```

SEE ALSO

sendfax(1), *hfaxd*(8C), *hylafax-server*(5F)

NAME

HylaFAX-client − introduction to *HylaFAX* client applications and usage

SYNOPSIS

sendfax [*options*] [*files*. . .]
sendpage [*options*] [*message*. . .]
faxstat [*options*]
faxrm [*options*]
faxalter [*options*] *jobid*. . .
fax2ps [*options*] [*files*. . .]

DESCRIPTION

HylaFAX is a telecommunication system for UNIX® systems. Among the features of *HylaFAX* are:

- *HylaFAX* runs as a network service; this means a modem may be effectively shared by a large number of users.
- *HylaFAX* can be configured to work with a wide variety of modems on a wide variety of systems.
- Access to the system can be restricted by the administrator to selected hosts and/or users.
- Transmission requests may be processed immediately (default) or queued for processing at a later time, in the manner of the *at*(1) command.
- Remote facsimile machines may be polled to retrieve publicly available documents.
- POSTSCRIPT® and TIFF Class F documents are passed directly to the fax server for transmission; the system attempts to convert other file formats to either POSTSCRIPT or TIFF through the use of an extensible file typing and conversion facility. In normal operation ASCII-text, *troff*(1) output, and Silicon Graphics images are automatically converted. Additional file formats can be added; see *typerules*(5F).
- The *faxcover*(1) program can be automatically invoked to create a cover page for each facsimile, using information deduced by the *sendfax* command. Alternatively, users may supply their own cover pages using their preferred tools.
- Facsimile are normally imaged in a system-default page size (usually letter-size pages, 8.5" by 11", for sites in North America). Alternate page sizes can be specified with a **−s** option to all *HylaFAX* programs. Well known page sizes include: ISO A3, ISO A4, ISO A5, ISO A6, ISO B4, North American Letter, American Legal, American Ledger, American Executive, Japanese Letter, and Japanese Legal. Note that it may not be permissible to image into the full page area; the guaranteed reproducible area for a page is typically smaller. Also, note that while arbitrary page sizes can be specified, only a limited number of page dimensions are supported by the facsimile protocol. Thus if an odd-size facsimile is submitted for transmission it may not be possible to determine if it can be sent until the fax server establishes communication with the remote facsimile machine.
- Facsimile can be sent at low resolution (98 lines/inch) or medium resolution (196 lines/inch)—often called *fine mode*. Documents with mixed resolution pages are handled correctly.
- Users are notified by electronic mail if a job can not be transmitted. It is also possible to receive notification by mail when a job has been completed successfully and each time that the job is requeued for retransmission. Any untransmitted documents are returned to the sender by electronic mail in a form suitable for resubmission.
- Support is provided for broadcasting facsimile. The *HylaFAX* server software optimizes preparation of broadcast documents and the client applications support the notion of a *job group* which permits a group of jobs to be manipulated together.
- Support is provided for transmitting alpha-numeric messages to pager devices or GSM mobiles using the Simple Network Paging Protocol (SNPP) and the IXO or UCP protocol (for message delivery).

The *HylaFAX* software is divided into two packages: software used on client machines and software used on machines where one or more modems reside. Client software includes:

- *sendfax*, a program to submit outgoing facsimile;
- *sendpage*, a program to submit alpha-numeric messages to SNPP servers;
- *faxstat*, a program obtain status information about *HylaFAX* servers;
- *faxrm*, a program to remove jobs and documents;
- *faxalter*, a program to change parameters of queued jobs; and
- *fax2ps*, a program that converts facsimile documents to POSTSCRIPT so that they may be viewed with a POSTSCRIPT previewer or printed on a POSTSCRIPT printer (this program is actually part of the companion TIFF distribution that is used by *HylaFAX*).

Many systems also support submission of outgoing facsimile by electronic mail and/or graphical interfaces to the *sendfax* program. Such facilities are site-dependent; consult local documentation for more information.

GETTING STARTED

To use the *HylaFAX* client software on your machine you need to either load the appropriate software on your machine, or you need to locate a machine that already has the client software installed and setup symbolic links to the appropriate directories. If you choose the latter, then beware that you need links to three directories: the directory where client applications reside, */usr/bin*, the directory where the client application database files reside, */usr/share/fax*, and the directory where document conversion programs reside, */usr/sbin* (the last two directories may be the same on your system).

Once the software is setup on your machine you need to locate a host that has a facsimile server that you can use. The host, and possibly the modem on the host, should be defined in your environment in the FAXSERVER variable. For example, for *csh* users,

    setenv FAXSERVER flake.asd

or for *ksh* or *sh* users,

    FAXSERVER=flake.asd; export FAXSERVER

If there are multiple modems on your server then you may be assigned to use a specific modem. For example, if you are to use the modem attached to the *ttym2* port on the server machine, then the FAXSERVER variable should be setup as

    FAXSERVER=ttym2@flake.asd; export FAXSERVER

(Note: the SNPPSERVER environment variable is used instead of FAXSERVER by the *sendpage* program; consult *sendpage*(8C) for more information.)

Note also, that before you can submit outgoing facsimile jobs the administrator for the facsimile server may need to register your identity in an access control list. You will encounter the message "*530 User %s access denied.*" if access to your server is controlled and you are not properly registered or you may be prompted for a password and then denied service with "*530 Login incorrect.*".

DIAL STRINGS

A dial string specifies how to dial the telephone in order to reach a destination facsimile machine. *HylaFAX* permits arbitrary strings to be passed to the facsimile server so that users can specify credit card information, PBX routing information, etc. Alphabetic characters are automatically mapped to their numeric key equivalents (e.g. "1800GotMilk" becomes "18004686455"). Other characters can be included for readability; anything that must be stripped will be removed by the server before the dialing string is passed to the fax modem. Private information such as credit card access codes are withheld from status messages and publicly accessible log files (with proper configuration). Facsimile servers also automatically insert any leading dialing prefixing strings that are required to place outgoing phone calls; e.g. dialing "9" to get an outside line. Additionally, if a phone number is fully specified with the international direct dialing digits (IDDD), then any prefixing long distance or international dialing codes that are required to place the call will be inserted in the dial string by the server. For example, "+31.77.594.131" is a phone number in the Netherlands; it would be converted to "0113177594131" if the call is placed in the United States. The number "+14159657824" is a phone number in California; if this number is called from within the 415 area code in the United States, then the server would automatically convert this to "9657824" because in the San Francisco Bay Area, local phone calls must not include the area code and long distance prefixing code.

The general rule in crafting dial strings is to specify exactly what you would dial on your telephone; and, in addition, the actual phone number can be specified in a location-independent manner by using the IDD syntax of "*+country-code local-part*".

COVER PAGES

The *sendfax* program can automatically generate a cover page for each outgoing facsimile. Such cover pages are actually created by the *faxcover*(1) program by using information that is deduced by *sendfax* and information that is supplied on the command line invocation of *sendfax*. Users may also request that *sendfax* not supply a cover page and then provide their own cover page as part of the data that is to be transmitted.

Automatically-generated cover pages may include the following information:

- the sender's name, affiliation, geographic location, fax number, and voice telephone number;
- the recipient's name, affiliation, geographic location, fax number, and voice telephone number;
- text explaining what this fax is "regarding";
- text commentary;
- the local date and time that the job was submitted;
- the number of pages to be transmitted.

Certain of this information is currently obtained from a user's personal facsimile database file; **~/.faxdb**. Note that this file is deprecated; it is described here only because it is still supported for compatiblity with older versions of the software.

The **.faxdb** file is an ASCII file with entries of the form

keyword **:** *value*

where *keyword* includes:

Name	a name associated with destination fax machine;
Company	a company name;
Location	in-company locational information, e.g. a building#;
FAX-Number	phone number of fax machine;
Voice-Number	voice telephone number.

Data is free format. Whitespace (blank, tab, newline) can be freely interspersed with tokens. If tokens include whitespace, they must be encloseed in quote marks (""""). The "#" character introduces a comment—everything to the end of the line is discarded.

Entries are collected into aggregate records by enclosing them in "[]". Records can be nested to create a hierarchy that that supports the inheritance of information—unspecified information is inherited from parent aggregate records.

For example, a sample file might be:

```
[   Company:        "Silicon Graphics, Inc."
    Location:       "Mountain View, California"
    [ Name: "Sam Leffler"       FAX-Number: +1.415.965.7824 ]
]
```

which could be extended to include another person at Silicon Graphics with the following:

```
[   Company:        "Silicon Graphics, Inc."
    Location:       "Mountain View, California"
    [ Name: "Sam Leffler"       FAX-Number: +1.415.965.7824 ]
```

```
    [ Name: "Paul Haeberli"    FAX-Number: +1.415.965.7824 ]
]
```

Experience indicates that the hierarchical nature of this database format makes it difficult to maintain with automated mechanisms. As a result it is being replaced by other, more straightforward databases that are managed by programs that front-end the *sendfax* program.

CONFIGURATION FILES

HylaFAX client applications can be tailored on a per-user and per-site basis through configuration files. Per-site controls are placed in the file **/usr/share/fax/hyla.conf**, while per-user controls go in **~/.hylarc**. In addition a few programs that have many parameters that are specific to their operation support an additional configuration file; these files are identified in their manual pages.

Configuration files have a simple format and are entirely ASCII. A configuration parameter is of the form

tag: *value*

where a *tag* identifies a parameter and a *value* is either a string, number, or boolean value. Comments are introduced by the ‘‘#’’ character and extend to the end of the line. String values start at the first non-blank character after the ‘‘:’’ and continue to the first non-whitespace character or, if whitespace is to be included, may be enclosed in quote marks (‘‘"’’). String values enclosed in quote marks may also use the standard C programming conventions for specifying escape codes; e.g. ‘‘\n’’ for a newline character and ‘‘\xxx’’ for an octal value. Numeric values are specified according to the C programming conventions (leading ‘‘0x’’ for hex, leading ‘‘0’’ for octal, otherwise decimal). Boolean values are case insensitive. For a true value, either ‘‘Yes’’ or ‘‘On’’ should be used. For a false value, use ‘‘No’’ or ‘‘Off’’.

RECEIVED FACSIMILE

Incoming facsimile are received by facsimile servers and deposited in a receive queue directory on the server machine. Depending on the server’s configuration, files in this directory may or may not be readable by normal users. The *faxstat* program can be used to view the contents of the receive queue directory:

```
hyla% faxstat -r
HylaFAX scheduler on hyla.chez.sgi.com: Running
Modem ttyf2 (+1 510 999-0123): Running and idle

Protect Page  Owner        Sender/TSI  Recvd@ Filename
-rw-r--    9  fax       1 510 5268781 05Jan96 fax00005.tif
-rw-r--    8  fax       1 510 5268781 07Jan96 fax00009.tif
-rw-r--    2  fax       1 510 5268781 07Jan96 fax00010.tif
-rw-r--    3  fax        +14159657824 08Jan96 fax00011.tif
-rw-r--    2  fax        +14159657824 08Jan96 fax00012.tif
```

Consult the *faxstat* manual page for a more detailed description of this information.

Received facsimile are stored as TIFF Class F files. These files are bilevel images that are encoded using the CCITT T.4 or CCITT T.6 encoding algorithms. The *fax2ps*(1) program can be used to view and print these files. A file can be viewed by converting it to POSTSCRIPT and then viewing it with a suitable POSTSCRIPT previewing program, such as *xpsview*(1) (Adobe’s Display POSTSCRIPT-based viewer), *ghostview*(1) (a public domain previewer), or image viewer programs such as *viewfax*(1) (public domain), *faxview*(1) (another public domain TIFF viewer program), *xv*(1) (shareware and/or public domain), or *xtiff*(1) (a program included in the public domain TIFF software distribution). Consult your local resources to figure out what tools are available for viewing and printing received facsimile.

CLIENT-SERVER PROTOCOL

HylaFAX client applications communicate with servers using either a special-purpose *communications protocol* that is modeled after the Internet File Transfer Protocol (FTP) or, when submitting alpha-numeric pages, the Simple Network Paging Protocol (SNPP), specified in RFC 1861. All client programs support a **−v** option that can be used to observe the protocol message exchanges. In some situations it may be more effective to communicate directly with a *HylaFAX* server using the client-server protocol. This can be accomplished with an FTP or Telnet client application; though an FTP client is recommended because it implements the protocol needed to obtain server status information. For information on the server-side support provided with *HylaFAX* consult *hfaxd*(8C). For documentation on the client-server fax protocol consult RFC XXXX (*to be filled in*).

EXAMPLES

This section gives several examples of command line usage; consult the manual pages for the individual commands for information on the options and program operation.

The following command queues the file **zall.ps** for transmission to John Doe at the number (123)456-7890 using fine mode; the server will attempt to send it at 4:30 A.M.:

```
sendfax -a "0430" -m -d "John Doe@1.123.456.7890" zall.ps
```

(the leading "1." is supplied to dial area code "123" in the United States.)

The following command generates a one-page facsimile that is just a cover page:

```
faxcover -t "John Doe" -n "(123)456-7890"
    -c "Sorry John, I forgot the meeting..." |
    sendfax -n -d "(123)456-7890"
```

(note that the line was broken into several lines solely for presentation.)

The following command displays the status of the facsimile server and any jobs queued for transmission:

```
faxstat -s
```

The following command displays the status of the facsimile server and any documents waiting in the receive queue on the server machine:

```
faxstat -r
```

The following command shows how to use an FTP client program to communicate directly with a *HylaFAX* server:

```
hyla% ftp localhost hylafax
Connected to localhost.
220 hyla.chez.sgi.com server (HylaFAX (tm) Version 4.0beta005) ready.
Name (localhost:sam):
230 User sam logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir sendq
200 PORT command successful.
150 Opening new data connection for "sendq".
208  126 S    sam 5268781       0:3   1:12   16:54 No local dialtone
226 Transfer complete.
ftp> quote jkill 208
200 Job 208 killed.
ftp> dir doneq
200 PORT command successful.
150 Opening new data connection for "doneq".
208  126 D    sam 5268781       0:3   1:12         No local dialtone
226 Transfer complete.
ftp> quote jdele 208
200 Job 208 deleted; current job: (default).
ftp> dir docq
200 PORT command successful.
150 Opening new data connection for "docq".
-rw----   1      sam    11093 Jan 21 16:48 doc9.ps
226 Transfer complete.
ftp> dele docq/doc9.ps
```

```
250 DELE command successful.
ftp> dir recvq
200 PORT command successful.
150 Opening new data connection for "recvq".
-rw-r--    4  fax       1 510 5268781 30Sep95 faxAAAa006uh
-rw-r--    9  fax        +14159657824 11Nov95 faxAAAa006nC
-rw----   25  fax        +14159657824 Fri08PM fax00016.tif
226 Transfer complete.
ftp> quit
221 Goodbye.
```

The following command shows how to use a Telnet client program to communicate directly with an SNPP server:

```
hyla% telnet melange.esd 444
Trying 192.111.25.40...
Connected to melange.esd.sgi.com.
Escape character is '^]'.
220 melange.esd.sgi.com SNPP server (HylaFAX (tm) Version 4.0beta010) ready.
login sam
230 User sam logged in.
help
214 The following commands are recognized (* =>'s unimplemented).
214 2WAY*   ALER*   DATA    HOLD    LOGI    MSTA*   PING    RTYP*   STAT
214 ABOR    CALL*   EXPT*   KTAG*   MCRE*   NOQU*   QUIT    SEND    SUBJ
214 ACKR*   COVE*   HELP    LEVE    MESS    PAGE    RESE    SITE
250 Direct comments to FaxMaster@melange.esd.sgi.com.
page 5551212
250 Pager ID accepted; provider: 1800SkyTel pin: 5551212 jobid: 276.
send
250 Message processing completed.
quit
221 Goodbye.
Connection closed by foreign host.
```

FILES

/usr/bin/sendfax	for sending facsimile
/usr/bin/sendpage	for sending alpha-numeric pages
/usr/bin/fax2ps	for converting facsimile to POSTSCRIPT
/usr/bin/faxalter	for altering queued jobs
/usr/bin/faxcover	for generating cover sheets
/usr/bin/faxmail	for converting email to POSTSCRIPT
/usr/bin/faxrm	for removing queued jobs
/usr/bin/faxstat	for facsimile server status
/usr/sbin/sgi2fax	SGI image file converter
/usr/sbin/textfmt	ASCII text converter
/usr/share/fax/typerules	file type and conversion rules
/usr/share/fax/pagesizes	page size database
/usr/share/fax/faxcover.ps	prototype cover page
/usr/share/fax/dialrules	optional client dialstring rules
/usr/tmp/sndfaxXXXXXX	temporary files

SEE ALSO

at(1), *fax2ps*(1), *faxalter*(1), *faxcover*(1), *faxmail*(1), *faxrm*(1), *faxstat*(1), *sgi2fax*(1), *faxq*(8C), *viewfax*(1), *hylafax-server*(5F), *dialrules*(5F), *pagesizes*(5F), *typerules*(5F), *services*(4)

NAME

config − *HylaFAX* configuration database

DESCRIPTION

HylaFAX uses a configuration file for the central queueing agent and for each facsimile modem attached to a machine. These files contain information about the modems and about how the *HylaFAX* server processes that service these modem should operate. Configuration files are located in the **etc** subdirectory. The configuration file for the *faxq*(8C) program is named **config**. The per-modem configuration files used by the *faxgetty*(8C), *faxsend*(8C), and *pagesend*(8C) programs have names of the form **config.***devid***,** where *devid* is an identifier derived from the name of the modem's device file; for example *ttym2* for */dev/ttym2*, *term_10* for */dev/term/10*.

Configuration files have a simple format and are entirely ASCII. A configuration parameter is of the form

tag: *value*

where a *tag* identifies a parameter and a *value* is either a string, number, or boolean value. Comments are introduced by the ''#'' character and extend to the end of the line. String values start at the first non-blank character after the '':'' and continue to the first non-whitespace character or, if whitespace is to be included, may be enclosed in quote marks (''"''). String values enclosed in quote marks may also use the standard C programming conventions for specifying escape codes; e.g. ''\n'' for a newline character and ''\xxx'' for an octal value. Numeric values are specified according to the C programming conventions (leading ''0x'' for hex, leading ''0'' for octal, otherwise decimal), except for a few parameters, such as **RecvFileMode**, whose value is interpreted strictly as an octal number and **PageChopThreshold** whose value is interpreted strictly as a floating point number. Boolean values are case insensitive. For a true value, either ''Yes'' or ''On'' should be used. For a false value, use ''No'' or ''Off''.

The following table itemizes the tags and the expected type of their value. The first part lists items specific to the operation of each server, while the second part has items related to the servicing of the modem. Parameters marked with a [1] are used only by the *HylaFAX* scheduler process and should appear only in the file named **config**. Parameters marked with a [2] are used by both the scheduler and the per-device applications and so should appear in all appropriate configuration files (i.e. **config** and any per-device files that make sense). Parameters not otherwise marked should appear only in per-device configuration files. More detailed information about each item is given following the table.

Tag	Type	Default	Description
AdaptiveAnswer	boolean	No	enable adaptive answer of inbound calls
AnswerRotary	string	Any	alternatives for answering calls
AnswerBias	integer	−	bias to apply to successful rotary answer
AreaCode[2]	string	−	local area code
CIDName	string	−	Caller ID name pattern string
CIDNumber	string	−	Caller ID number pattern string
ClocalAsRoot	boolean	No	set CLOCAL using root UID
ContCoverCmd[1]	string	bin/mkcover	continuation cover page generation script
ContCoverPage[1]	string	−	continuation cover page template filename
CountryCode[2]	string	−	local country code
DestControls[1]	string	−	per-destination controls file
DeviceMode	octal	0600	protection mode to use for modem device
DialStringRules[2]	string	−	dial string rules file
FAXNumber	string	−	facsimile modem phone number
FaxRcvdCmd	string	bin/faxrcvd	notification script for received facsimile
GettyArgs	string	−	arguments passed to getty program
InternationalPrefix[2]	string	−	dialing prefix for international calls
JobReqBusy	integer	180	requeue interval for BUSY dial result
JobReqDataConn	integer	300	requeue interval for data connection dial result
JobReqNoAnswer	integer	300	requeue interval for NO ANSWER dial result
JobReqNoCarrier	integer	300	requeue interval for NO CARRIER dial result
JobReqNoFCon	integer	300	requeue interval for carrier without +FCON dial result

JobReqOther[2]	integer	300	requeue interval for ‘‘other’’ problems
JobReqProto	integer	60	requeue interval after fax protocol error
LockDataCalls	boolean	Yes	lock modem for the duration of an inbound data call
LockVoiceCalls	boolean	Yes	lock modem for the duration of an inbound voice call
LogFacility[2]	string	daemon	*syslog* facility name for ServerTracing messages
LogFileMode	octal	0600	protection mode to use for session log files
LocalIdentifier	string	−	local station identification string
LongDistancePrefix	string	−	dialing prefix for long distance calls
MaxConcurrentJobs[1]	integer	1	max concurrent jobs to process for a destination
MaxConsecutiveBadLines	integer	5	max consecutive bad rows for OK page quality
MaxDials[1]	integer	12	max phone calls to make to transmit a job
MaxRecvPages	integer	*unlimited*	max pages to permit in a receive
MaxSendPages[1]	integer	*unlimited*	max pages to permit in a send
MaxSetupAttempts	integer	2	max attempts to initialize a modem
MaxTries[1]	integer	3	max attempts to transmit a job
ModemGroup[1]	string	−	define a name for a set of modems
ModemPriority	integer	255	scheduling priority for outbound jobs
ModemReadyState	string	R	‘‘ready state’’ sent by *faxgetty*
NoCarrierRetrys	integer	1	number of times to retry dialing on ‘‘NO CARRIER’’
NotifyCmd[2]	string	bin/notify	user notification command script
PageChop[1]	string	last	control automatic truncation of trailing whitespace
PageChopThreshold[1]	float	3.0	whitespace truncation threshold (inches)
PCL2FaxCmd[1]	string	bin/pcl2fax	PCL RIP command script
PercentGoodLines	integer	95	percentage of good rows for OK page quality
PollLockWait[2]	integer	30	polling interval when in ‘‘lock wait’’ state (secs)
PollModemWait	integer	30	polling interval when in ‘‘modem wait’’ state (secs)
PollRcvdCmd	string	bin/pollrcvd	delivery script for facsimile received by polling
PostScriptTimeout[1]	integer	300	timeout on POSTSCRIPT interpreter runs (secs)
PriorityScheduling	boolean	*see below*	use available priority job scheduling mechanism
PS2FaxCmd[1]	string	bin/ps2fax	POSTSCRIPT RIP command script
QualifyCID	string	−	file of Caller-ID patterns for checking inbound calls
QualifyTSI	string	−	file of TSI patterns for qualifying senders
RecvDataFormat	string	adaptive	format for received facsimile data
RecvFileMode	octal	0600	protection mode to use for received facsimile files
RingData	string	−	distinctive ring data call identifier
RingFax	string	−	distinctive ring fax call identifier
RingsBeforeAnswer	integer	0	rings to wait before answering phone
RingVoice	string	−	distinctive ring voice call identifier
RTNHandlingMethod	string	Retransmit	RTN signal handling method
SendFaxCmd[1]	string	bin/faxsend	fax transmit command script
SendPageCmd[1]	string	bin/pagesend	pager transmit command script
SendUUCPCmd[1]	string	bin/uucpsend	UUCP transmit command script
ServerTracing[2]	integer	1	non-session server tracing
SessionTracing[2]	integer	1	send and receive session tracing
SpeakerVolume	string	Quiet	volume level for modem speaker
TagLineFont	string	−	tag line font filename
TagLineFormat	string	*see below*	tag line format string
TIFF2FaxCmd[1]	string	bin/tiff2fax	TIFF converter command script
TimeOfDay[1]	string	Any	default time-of-day restrictions
Use2D[1]	boolean	Yes	restrict/permit use of 2D-encoded fax data
UUCPLockMode[2]	octal	0600	protection mode for UUCP lock files
UUCPLockDir[2]	string	*see below*	UUCP lockfile directory
UUCPLockTimeout[2]	integer	30	time before removing stale UUCP lockfile
UUCPLockType[2]	string	*see below*	UUCP lockfile type

VGettyArgs	string	–	arguments passed to voice getty program
WedgedCmd[1]	string	bin/wedged	command to run when modem is wedged
ModemAnswerCmd	string	ATA	command for answering phone
ModemAnswerDataBeginCmd	string	–	command for start of answered data call
ModemAnswerDataCmd	string	ATA	command for answering data call
ModemAnswerFaxBeginCmd	string	–	command for start of answered fax call
ModemAnswerFaxCmd	string	ATA	command for answering fax call
ModemAnswerResponseTimeout	integer	180000	answer command timeout (ms)
ModemAnswerVoiceBeginCmd	string	–	command for start of answered voice call
ModemAnswerVoiceCmd	string	ATA	command for answering voice call
ModemATCmdDelay	integer	0	delay before sending modem an AT cmd (ms)
ModemBaudRateDelay	integer	0	delay (ms) after setting baud rate
ModemClassQueryCmd	string	AT+FCLASS=?	command for querying modem services
ModemCommaPauseTimeCmd	string	ATS8=2	command for setting time to pause for ‘‘,’’ in dialing string
ModemDialCmd	string	ATDT%s	command for dialing (%s for number to dial)
ModemDialResponseTimeout	integer	180000	dialing command timeout (ms)
ModemDTRDropDelay	integer	75	delay (ms) between DTR OFF and DTR ON
ModemEchoOffCmd	string	ATE0	command for disabling command echo
ModemFlowControl	string	XONXOFF	DTE-DCE flow control scheme
ModemFrameFillOrder	string	LSB2MSB	bit order for HDLC frames
ModemHardFlowCmd	string	–	command for setting hardware flow control between DTE and
ModemMinSpeed	string	2400	minimum acceptable transmit speed
ModemMfrQueryCmd	string	–	command for querying modem manufacturer
ModemModelQueryCmd	string	–	command for querying modem model
ModemNoAutoAnswerCmd	string	ATS0=0	command for disabling auto-answer
ModemNoFlowCmd	string	–	command for disabling hardware flow control between DTE a
ModemOnHookCmd	string	ATH0	command for placing phone ‘‘on hook’’
ModemPageDoneTimeout	integer	180000	page send/receive timeout (ms)
ModemPageStartTimeout	integer	180000	page send/receive timeout (ms)
ModemRate	integer	19200	baud rate to use for DCE-DTE communication
ModemRecvFillOrder	string	*see below*	bit order for received facsimile data
ModemResetCmds	string	–	additional commands when resetting modem
ModemResetDelay	integer	2600	delay (ms) after placing DTR ON
ModemResultCodesCmd	string	ATQ0	command for enabling result codes
ModemRevQueryCmd	string	*see below*	command for querying modem firmware revision
ModemSendBeginCmd	string	–	command to send on establishing carrier
ModemSendFillOrder	string	LSB2MSB	bit order for sending facsimile data
ModemSetVolumeCmd	string	*see below*	commands for setting modem speaker volume level
ModemSetupAACmd	string	–	command for setting up adaptive answer
ModemSetupDCDCmd	string	–	command for setting up DCD handling
ModemSetupDTRCmd	string	–	command for setting up DTR handling
ModemSoftFlowCmd	string	–	command for setting software flow control between DTE and
ModemSoftResetCmd	string	ATZ	command for doing a soft reset
ModemType	string	*see below*	modem type
ModemVerboseResultsCmd	string	ATV1	command for enabling verbose result codes
ModemWaitForConnect	boolean	No	force server to wait for ‘‘CONNECT’’ response on answer
ModemWaitTimeCmd	string	ATS7=60	command for setting time to wait for carrier when dialing
FaxT1Timer	integer	35000	CCITT T.30 T1 timer (ms)
FaxT2Timer	integer	7000	CCITT T.30 T2 timer (ms)
FaxT4Timer	integer	3100	CCITT T.30 T4 timer (ms)
Class0Cmd	string	AT+FCLASS=0	Class 0: command to enter class 0
Class1Cmd	string	AT+FCLASS=1	Class 1: command to enter class 1

Class1HFLOCmd	string	−	Class 1: command to set hardware flow control
Class1FrameOverhead	integer	4	Class 1: extra bytes in a received HDLC frame
Class1NFLOCmd	string	−	Class 1: command to set no flow control
Class1RecvAbortOK	integer	100	Class 1: max wait (ms) for ‘‘OK’’ after recv abort
Class1RecvIdentTimer	integer	35000	Class 1: max wait (ms) for initial ident frame
Class1SFLOCmd	string	−	Class 1: command to set software flow control
Class1TCFMaxNonZero	integer	10	Class 1: max% of non-zero data in good TCF
Class1TCFMinRun	integer	1000	Class 1: minimum zero run in good TCF
Class1TCFRecvTimeout	integer	4500	Class 1: max wait (ms) for TCF
Class1TCFResponseDelay	integer	75	Class 1: delay between TCF and ack/nak
Class1SendEOPDelay	integer	200	Class 1: delay before sending EOP
Class1SendPPMDelay	integer	75	Class 1: delay before sending MPS post-page message
Class1SendTCFDelay	integer	75	Class 1: delay between sending DCS and TCF
Class1SwitchingDelay	integer	150	Class 1: delay in switching transmision direction (ms)
Class1TrainingRecovery	integer	1500	Class 1: delay after failed training
Class2Cmd	string	AT+FCLASS=2	Class 2: command to enter class 2/2.0
Class2AbortCmd	string	AT+FK	Class 2: command to abort active session
Class2APCmd	string	AT+FAP	Class 2: enable support for sending and receiving SUB, SEP, a
Class2APQueryCmd	string	AT+FAP=?	Class 2: query capabilities for sending and receiving SUB, SE
Class2BORCmd	string	AT+FBOR=0	Class 2: command to setup bit order
Class2BUGCmd	string	AT+FBUG=1	Class 2: command to enable HDLC frame tracing
Class2CIGCmd	string	AT+FCIG	Class 2: command to set polling identifier
Class2CRCmd	string	AT+FCR=1	Class 2: command to enable receive capability
Class2CQCmd	string	−	Class 2: command to setup copy quality parameters
Class2CQQueryCmd	string	AT+FCQ=?	Class 2: command to query modem copy quality capabilities
Class2DCCCmd	string	AT+FDCC	Class 2: command to set modem capabilities
Class2DCCQueryCmd	string	AT+FDCC=?	Class 2: command to query modem capabilities
Class2DISCmd	string	AT+FDIS	Class 2: command to set session parameters
Class2DDISCmd	string	−	Class 2: command to set session parameters before dialing
Class2HFLOCmd	string	−	Class 2: command to set hardware flow control
Class2LIDCmd	string	AT+FLID	Class 2: command to set local identifier string
Class2MINSPCmd	string	AT+FMINSP	Class 2: command to set minimum transmit speed
Class2NFLOCmd	string	−	Class 2: command to set no flow control
Class2PACmd	string	AT+FPA	Class 2: set polling address string
Class2PHCTOCmd	string	AT+FPHCTO=30	Class 2: command to set Phase C timeout parameter
Class2PTSCmd	string	AT+FPTS	Class 2: command to set received page status
Class2PWCmd	string	AT+FPW	Class 2: set password string
Class2RecvDataTrigger	string	‘‘\21’’	Class 2: character to send to trigger recv
Class2RELCmd	string	−	Class 2: command to enable byte-aligned EOL codes
Class2SACmd	string	AT+FSA	Class 2: set destination subaddress string
Class2SendRTC	boolean	No	Class 2: append RTC to page data on transmit
Class2SFLOCmd	string	−	Class 2: command to set software flow control
Class2SPLCmd	string	AT+FSPL	Class 2: command to set polling request
Class2TBCCmd	string	AT+FTBC=0	Class 2: command to enable stream mode
Class2XmitWaitForXON	boolean	Yes	Class 2: wait for XON before sending facsimile data
Class2Cmd	string	AT+FCLASS=2.0	Class 2.0: command to enter class 2/2.0
Class2AbortCmd	string	AT+FKS	Class 2.0: command to abort active session
Class2APCmd	string	AT+FAP	Class 2.0: enable support for sending and receiving SUB, SEF
Class2APQueryCmd	string	AT+FAP=?	Class 2.0: query capabilities for sending and receiving SUB, S
Class2BORCmd	string	AT+FBO=0	Class 2.0: command to setup bit order
Class2BUGCmd	string	AT+FBU=1	Class 2.0: command to enable HDLC frame tracing
Class2CIGCmd	string	AT+FPI	Class 2.0: command to set polling identifier
Class2CRCmd	string	AT+FCR=1	Class 2.0: command to enable receive capability

Class2CQCmd	string	–	Class 2.0: command to setup copy quality parameters
Class2CQQueryCmd	string	AT+FCQ=?	Class 2.0: command to query modem copy quality capabilities
Class2DCCCmd	string	AT+FCC	Class 2.0: command to set modem capabilities
Class2DCCQueryCmd	string	AT+FCC=?	Class 2.0: command to query modem capabilities
Class2DISCmd	string	AT+FIS	Class 2.0: command to set session parameters
Class2HFLOCmd	string	AT+FLO=2	Class 2.0: command to set hardware flow control
Class2LIDCmd	string	AT+FLI	Class 2.0: command to set local identifier string
Class2MINSPCmd	string	AT+FMS	Class 2.0: command to set minimum transmit speed
Class2NFLOCmd	string	AT+FLO=0	Class 2.0: command to set no flow control
Class2NRCmd	string	AT+FNR=1,1,1,1	Class 2.0: command to set negotiation message reporting
Class2PACmd	string	AT+FPA	Class 2.0: set polling address string
Class2PHCTOCmd	string	AT+FCT=30	Class 2.0: command to set Phase C timeout parameter
Class2PIECmd	string	AT+FIE=0	Class 2.0: command to set procedure interrupt handling
Class2PWCmd	string	AT+FPW	Class 2.0: set password string
Class2PTSCmd	string	AT+FPS	Class 2.0: command to set received page status
Class2RecvDataTrigger	string	‘‘\22’’	Class 2.0: character to send to trigger recv
Class2RELCmd	string	–	Class 2.0: command to enable byte-aligned EOL codes
Class2SACmd	string	AT+FSA	Class 2.0: set destination subaddress string
Class2SendRTC	boolean	No	Class 2.0: append RTC to page data on transmit
Class2SFLOCmd	string	AT+FLO=1	Class 2.0: command to set software flow control
Class2SPLCmd	string	AT+FSP	Class 2.0: command to set polling request
Class2TBCCmd	string	AT+FPP=0	Class 2.0: command to enable stream mode
PagerSetupCmds	string	–	commands for setting up modem for a pager call
PagerMaxMsgLength	integer	128	max length of a text message
IXOService	string	‘‘PG’’	IXO: service identification string
IXODeviceID	string	‘‘1’’	IXO: device identification string
IXOMaxUnknown	integer	3	IXO and UCP: max unknown responses before abort
IXOIDProbe	integer	2	IXO: time between sending \r during ID sequence (secs)
IXOIDTimeout	integer	20	IXO: max time to wait for ID= response (secs)
IXOLoginRetries	integer	3	IXO: max attempts to login
IXOLoginTimeout	integer	15	IXO: max time to complete login (secs)
IXOGATimeout	integer	30	IXO: max time to wait for Go-Ahead response (secs)
IXOXmitRetries	integer	3	IXO and UCP: max retries to send text msg block
IXOXmitTimeout	integer	15	IXO and UCP: max time to transmit text msg block
IXOAckTimeout	integer	30	IXO: max time to wait for msg block ack (secs)

SERVER-ORIENTED CONFIGURATION PARAMETERS

These configuration parameters affect the general operation of the fax server.

AdaptiveAnswer

Control whether or not an adaptive answering strategy is used whereby an incoming call is answered in multiple ways according to the list specified in the **AnswerRotary** parameter. For example, if adaptive answering is enabled and **AnswerRotary** specifies ‘‘fax data’’, then an incoming call will first be answered as fax using **ModemAnswerFaxCmd** and then, if that fails, as data using **ModemAnswerDataCmd**. If the adaptive answer strategy is not enabled, then calls are answered using the appropriate **ModemAnswer*Cmd** parameter (normally **ModemAnswer-Cmd** for unattended operation). Note that the adaptive answer strategy depends on many factors that limit its usefulness. When calls are answered first as fax, it typically only works with a Class 1 modem because it is important that the sequence of operations related to answering as fax be completed in a short enough time that a subsequent answer for data be done before the caller times out and hangs up the telephone (for this reason **Class1RecvIdentTimer** should be set to a value that is shorter than **FaxT1Timer**.) Also, note that it is sometimes necessary to arrange that the phone be placed on-hook between successive answers; this can typically be done by inserting the appropriate command at the start of the second and subsequent **ModemAnswer*Cmd**s.

AnswerRotary

The sequence of answering techniques the server should ''rotate through'' when answer incoming calls. Answering techniques are:

Name	Description
fax	answer a fax call
data	answer a data call
voice	answer a voice call
extern	use the *vgetty* application to answer a call
any	answer a call of any/unknown type

When a call is not answered succesfully, the server advances the rotary to the next specified answering technique. If **AdaptiveAnswer** is enabled, the server will try all the answering techniques for each incoming call. Otherwise, the rotary is advanced *between* unsuccessful phone calls. For example, if **AdaptiveAnswer** is disabled and **AnswerRotary** is ''fax data'', then the calls will initially be answered as fax until the first call is received that does not appear to be from a fax machine; after that calls will be answered as if they were from a data modem until the next unsuccessful call; after which they will one again be answered as fax. Only the first three techniques listed are used; any additional ones are (silently) ignored. See also **AnswerBias**.

AnswerBias

The index into the **AnswerRotary** list that the server should rotate the list to after each successful incoming phone call. List indices are numbered starting at zero. For example, if **AnswerRotary** is ''fax data'' and **AnswerBias** is 1, then after each successful inbound call, the next call will be answered as data.

AreaCode[2]

The local area code in which the facsimile modem resides. This value is passed to the dial string rules that are used, for example, to formulate canonical phone numbers for dialing (see **DialStringRules** below.)

CIDName

A string that identifies the caller's identity in any Caller-ID status messages. The specified string is compared against any unrecognized status messages received from the modem before the phone is placed off-hook. If there is a match, then the remainder of the message is returned as the ''caller name''. For example, for the ZyXEL U-1496 this parameter would be set to ''CALLER NAME: '' (note the trailing space). See also **CIDNumber** and **QualifyCID**.

CIDNumber

A string that identifies the caller's phone number in any Caller-ID status messages. The specified string is compared against any unrecognized status messages received from the modem before the phone is placed off-hook. If there is a match, then the remainder of the message is returned as the phone number of the caller. For example, for the ZyXEL U-1496 this parameter would be set to ''CALLER NUMBER: ''. See also **CIDName** and **QualifyCID**.

ClocalAsRoot

Control whether operations that set the CLOCAL bit on the modem device special file are done with the effective user-ID set to the super-user or the ''fax'' user. By default such operations are done as the fax user, except under IRIX where they are done as the super-user (because IRIX disallows manipulation of CLOCAL by anyone but the super-user).

ContCoverCmd[1]

The command to invoke to generate a continuation cover page; see **ContCoverPage** and *mkcover*(8C).

ContCoverPage[1]

A template file to use in creating *continuation cover pages*. If this parameter is non-null, then the server will pass the filename to the command specified by **ContCoverCmd** to generate cover pages for outbound jobs that are continued after protocol errors. These cover pages identify the receiver and indicate that the document is a continued transmission. If this parameter is not specified or is null, then the server will not generate continuation cover pages. The specified pathname must be relative to the top of the fax server's spooling area.

CountryCode[2]
The local country code in which the facsimile modem resides. As for **AreaCode**, this value is passed to the dial string rules for use in formulating canonical phone numbers for dialing (among other things.)

DestControls[1]
An optional file containing rules for controlling parameters on a per-destination basis; see *destctrls*(5F) The following parameters may be controlled with this mechanism: **MaxConcurrentJobs**, **MaxDials**, **MaxSendPages**, **MaxTries**, **RejectNotice**, **SessionTracing**, and **TimeOfDay**. If no **DestControls** parameter is specified, or if the controls file does not specify *default* values for these parameters, then any settings in the **config** file used by *faxq*(8C) are used for default settings. Note that *faxq* automatically rereads the contents of a **DestControls** file if it is modified while the server is running.

DeviceMode
The file protection mode that the server should set for the modem device file. Note that this value is given in octal. The default value of 0600 implies that only the facsimile user (usually *uucp*) can access the modem. See also *chmod*(2).

DialStringRules[2]
The pathname of the file that holds the rules for processing user-specified dial strings; c.f. *dialrules*(5F). The specified pathname must be relative to the top of the fax server's spooling area; e.g. **etc/dialrules**.

FAXNumber
The phone number associated with the facsimile modem. This string is used to generate the Transmitter Subscriber Identification (TSI) and Caller Subscriber Identification (CSI) information passed to remote machines—unless the **LocalIdentifier** parameter is explicitly set. The **FAXNumber** is also used to name the session log file where information for incoming phone calls is recorded; see *hylafax-log*(5F). If this phone number is not a fully qualified number the values of **AreaCode** and **CountryCode** are used to generate the transmitted CSI and TSI.

InternationalPrefix[2]
The string to use to place an international phone call. In the United States, this is ‘‘011’’. This string is passed to the dial string rules.

JobReqBusy
The delay in seconds to wait before retrying a job whose dialing attempt failed with a ‘‘BUSY’’ status result.

JobReqDataConn
The delay in seconds to wait before retrying a facsimile job whose dialing attempt failed because a data modem answered the phone.

JobReqNoAnswer
The delay in seconds to wait before retrying a job whose dialing attempt failed with a ‘‘NO ANSWER’’ status result.

JobReqNoCarrier
The delay in seconds to wait before retrying a job whose dialing attempt failed with a ‘‘NO CARRIER’’ status result.

JobReqNoFCon
The delay in seconds to wait before retrying a facsimile job whose dialing attempt failed because the initial facsimile protocol handshake failed (i.e. no ‘‘+FCON:’’ result was received from a Class 2/2.0 modem).

JobReqOther[2]

The delay in seconds to wait before retrying a job that failed for a reason not already covered by one of the **Job*** parameters.

JobReqProto

The delay in seconds to wait before retrying a facsimile job that failed because of a fax protocol error.

GettyArgs

A string that indicates whether or not the server should invoke the *getty*(8C) program in response to an incoming call from a data modem. If the string value is not null, then it is interpreted as a set of arguments to pass to the getty program. Before supplying the arguments, the string is first scanned for ‘‘%’’-escape sequences: any appearance of ‘‘%l’’ is replaced with the tty name and any appearance of ‘‘%s’’ is replaced with the serial port speed (in decimal). The ‘‘%’’ character can be specified with ‘‘%%’’. If the **GettyArgs** parameter is not specified in the configuration file or if the string value is null, then data connections will be rejected. Note that in addition to the specified command line arguments, the *getty* program is invoked with its standard input, output, and error descriptors directed to the controlling tty device.

LocalIdentifier

The local station identification string to use when transmitting TSI and CSI strings (see **FAXNumber** above). NB: while this string may contain any printable ASCII characters; beware that CCITT T.30 specifies a restricted character set and some fax modems and machines may reject or not display non-conforming strings. If the local identifier is not specified, a canonical form of the **FAXNumber** is used instead.

LockDataCalls

Hold the UUCP lockfile during the time an inbound data call is processed by the external getty program. If this is disabled then the lockfile will be removed just before the getty program is invoked. Note however that doing this introduces a race condition whereby an outbound program may sneak in and take control of the modem after the inbound call has been accepted, but before the getty program has started up and installed it’s own lockfile.

LockVoiceCalls

Hold the UUCP lockfile during the time an inbound data call is processed by the external voice getty program. If this is disabled then the lockfile will be removed just before the voice getty program is invoked. Note however that doing this introduces a race condition whereby an outbound program may sneak in and take control of the modem after the inbound call has been accepted, but before the voice getty program has started up and installed it’s own lockfile.

LogFacility[2]

The symbolic name for the *syslog*(3) facility to use when logging error messages and informational/debugging messages requested through the **ServerTracing** parameter. The list of facility names is standardized; it can be found in the system include file **<syslog.h>**; comparisons are case-insensitive.

LogFileMode

The file protection mode that should be used when creating files to hold session logs. Note that this value is given in octal. The default value of 0600 implies that the log files are not generally accessible, which is important if sensitive information such as calling card access codes are logged. If log files are to be publicly accessible, this value may be set, for example, to 0644. See also *chmod*(2).

LongDistancePrefix[2]

The string to use to place a long distance phone call. In the United States, this is ‘‘1’’.

MaxConcurrentJobs[1]

The default value to use for the maximum number of jobs for the same destination that are processed concurrently. Note that each job is processed with a separate phone call; job batching is not currently supported. Thus this parameter defines the maximum number of concurrent phone calls to the same destination. Unless all of your fax destinations have multiple fax lines using the same fax number per destination then **MaxConcurrentJobs** is probably best left as the default of 1 and exceptions should be accomodated by the **DestControls** file.

MaxConsecutiveBadLines

The maximum number of consecutive erroneous rows of image data that the server will receive before it deems a page to have unacceptable copy quality. Setting this parameter to zero causes the server to not check the quality of received facsimile. See also **PercentGoodLines**.

MaxDials[1]

The default value for the maximum number of times the server will place a call for an outbound job. This value may be overridden by rules in the **DestControls** file. See also **MaxTries**.

MaxRecvPages

The maximum number of pages the server will accept in a received facsimile before it aborts the session.

MaxSendPages[1]

The default value for the maximum number of pages the server will permit in a transmitted facsimile. Outbound jobs that exceed this limit are rejected. This value may be overridden by rules in the **DestControls** file.

MaxSetupAttempts

The maximum number of times a *HylaFAX* server program will attempt to initialize a modem before considering it ‘‘*wedged*’’.

MaxTries[1]

The default value for the maximum number of times the server will attempt to communicate with a remote site. Note that for a call to be counted against this limit, carrier must be established; this is in contrast to the value of **MaxDials** which limits the number of calls that will be placed on behalf of a job. This value may be overridden by rules in the **DestControls** file. See also **MaxDials**.

ModemGroup[1]

Define a logical name for a set of modem devices. This parameter can be used to define a name that clients can use to refer to a collection of modems. The syntax is ‘‘*<name>*:*<regex>*’’ where *<name>* is the name for the group and *<regex>* is a regular expression that *faxq*(8C) matches against known modems. Thus for example, the default modem to use for a job, **any**, is defined as ‘‘any:.*’’. This parameter may be specified multiple times to define multiple modem classes. Modem classes may overlap. The actual set of modems in a modem class that are considered for use may vary based on which modems are believed to be ready for use.

ModemPriority

The scheduling priority to use for a modem. Priority values are in the range 0 to 255 with lesser values signifying higher (more desirable) priority. The *HylaFAX* scheduler process assigns the highest priority modem that is ready for use when scheduling an outbound job.

If multiple available modems in a job's selected **ModemGroup** have the same **ModemPriority** value, then the *HylaFAX* scheduler assigns jobs to them in a round-robin fashion.

NoCarrierRetrys

The number of times to redial a phone number after receiving a ‘‘NO CARRIER’’ result code. In normal operation the fax server will treat this result code to mean that a facsimile modem/machine did not answer the phone and reject the transmit job. This is done to avoid repeatedly dialing wrong phone numbers and depends on the modem distinguishing between no carrier and no answer. However, some modems are not capable of reliably distinguishing between no carrier and no answer, or when instructed to do so do not identify a busy signal correctly. For these modems one may find it necessary to increase the value of this parameter to compensate. It is strongly recommended, however, that this value not be set to a large number.

ModemReadyState

Define the *state* transmitted to the *HylaFAX* scheduler process each time a modem is made ready by a *faxgetty* process. State is one of: ''R'' (ready for use), ''B'' (busy), or ''D'' (down). The modem state can be used to control whether or not modems are assigned to outbound jobs. A modem's state is initially set to ''R'' (ready). Modem state can be dynamically changed with the *faxstate*(8C) program or by setting this configuration parameter with the *faxconfig*(8C) program.

NotifyCmd[2]

The command to invoke to notify a client of a job status change; see *notify*(8C).

PageChop[1]

Whether or not to automatically truncate trailing whitespace on pages of outbound facsimile (when possible). If page chopping is enabled, then pages with at least **PageChopThreshold** trailing whitespace on a page will be transmitted as a ''short page'' if the receiver is capable of accepting variable-length pages. The possible values are: ''last'' to enable chopping of the last page of each document, ''all'' to enable chopping of all pages, or ''none'' to disable page chopping.

PageChopThreshold[1]

The minimum amount of trailing whitespace that must be present on a page before page chopping will be attempted; see **PageChop**. Note that this value is specified in inches.

PCL2FaxCmd[1]

The command to invoke to convert a HP-PCL document submitted for transmission to TIFF/F, suitable for transmission (for future use).

PercentGoodLines

The minimum percentage of good rows of image data required for a received facsimile page to be deemed to have acceptable copy quality. Setting this parameter to zero causes the server to not check the quality of received facsimile. See also **MaxConsecutiveBadLines**.

PollLockWait

When polling for the presence/removal of a UUCP lockfile, the time (in seconds) to wait between checks. Lockfile polling occurs when the fax server collides with another process using the modem for outbound use.

PollModemWait

The time (in seconds) to wait between checks for a modem to become ready for use. Modem polling occurs when a modem fails to reset cleanly.

PriorityScheduling

Indicates whether the *HylaFAX* scheduler should utilize available priority job scheduling mechanisms to enhance realtime execution, particularly with Class 1 configurations on older or slower systems.

Currently **PriorityScheduling** is available for IRIX, SVR/4, HP-UX, and POSIX-compliant (i.e. Linux) operating systems. The default value varies to maintain traditional *HylaFAX* behavior. On IRIX, SVR/4, ans HP-UX systems the default is true. On others the default is false.

PS2FaxCmd[1]

The command to invoke to convert a POSTSCRIPT document submitted for transmission to TIFF/F, suitable for transmission; see *ps2fax*(8C).

QualifyCID

A string that specifies whether or not *Caller-ID information* should be checked against an access control list before accepting an inbound call. If **QualifyCID** is non-null and there is Caller-ID service, then only calls from phone numbers identified in the file specified by this string (typically **etc/cid**) will be accepted; see *cid*(5F). If **QualifyCID** is not specified in the configuration file, or the value is null, then all incoming calls will be answered. Note that Caller-ID screening is only available when this service is enabled on the phone line and the modem is capable of presenting Caller-ID status information to the host. The **CIDNumber** and **CIDName** parameters must also be setup to reflect the manner in which the modem returns Caller-ID status information to the host.

QualifyTSI

A string that specifies whether or not the identity of calling facsimile machines should be checked against an access control list before receiving facsimile. If **QualifyTSI** is non-null, then only messages from facsimile machines identified in the file specified by the string (typically **etc/tsi**) will be accepted; see *tsi*(5F). If **QualifyTSI** is not specified in the configuration file, or the value is null, then all incoming facsimile messages will be accepted.

RecvDataFormat

The data format (compression scheme) to write received facsimile data when copy quality checking is enabled on the host. (When copy quality checking is not done by the server the received facsimile data is written exactly as it is received from the modem.) The format may be one: ‘‘1-D MR’’, ‘‘2-D MR’’, ‘‘2-D MMR’’, or ‘‘adaptive’’. An ‘‘adaptive’’ format causes the received data to be written using the data format negotiated by the sender and receiver. Note that while 2-D MMR is the most space-efficient data format the resultant file may not be viewable (certain commonly used programs do not properly read multi-strip TIFF with 2-D MMR data).

RecvFileMode

The file protection mode that should be used when creating files to hold incoming facsimile. Note that this value is given in octal. The default value of 0600 implies that only the facsimile user can read received facsimile. If incoming facsimile are to be publicly accessible, this value may be set, for example, to 0644. See also *chmod*(2).

RingData

A modem status string that identifies that an incoming call is for data use. This string should be set when a phone line has been setup with *distinctive ring* service. See also **RingFax** and **RingVoice**.

RingFax

A modem status string that identifies that an incoming call is for facsimile use. This string should be set when a phone line has been setup with *distinctive ring* service. See also **RingData** and **RingVoice**.

RingsBeforeAnswer

The number of rings to wait before answering the phone. If this value is set to 0, then the phone will only be answered in response to an explicit answer command; see *faxanswer*(8C).

RingVoice

A modem status string that identifies that an incoming call is for voice use. This string should be set when a phone line has been setup with *distinctive ring* service. See also **RingData** and **RingFax**.

RTNHandlingMethod

Specifies how to react to RTN signal, received from the remote; one of ‘‘Retransmit’’, ‘‘Giveup’’ and ‘‘Ignore’’. ‘‘Retransmit’’ assumes that the page is not sent succesfully if RTN signal has been received. HylaFAX will make up to 2 additional attempts to send the page, decreasing signalling rate and retraining. If RTN is still there, it will place up to 2 additional calls. So if the remote always respond with RTN, the page will be send 9 times. Although this algorithm comply with T.30 specs and was originally implemented by Sam Leffler as the only possible choice, real fax machines behave completely different. There is a non-written rule among fax developers, that RTN means ‘‘over and out’’ -- hang up immediately and never try to send the same page to the same destination again. That is because RTN usually indicates problems with flow control, incorrectly encoded T.4 data, incompatibility between local and remote equipment etc., but very rarely is caused by the real noise on the line. This ‘‘over and out’’ behaviour can be activated by ‘‘Giveup’’ value. There is also third option, not so radical as ‘‘Giveup’’. Yes, we will never retransmit the page, but we can try to send the next page, and let the remote to decide what to do (accept our decision or hang up). Thus one page will (or will not) be missed but we have a chance to successfully send all other pages. This behaviour can be activated by ‘‘Ignore’’ value.

SendFaxCmd[1]
The command to use to process outbound facsimile jobs; see *faxsend*(8C).

SendPageCmd[1]
The command to use to process outbound pager jobs; see *pagesend*(8C).

SendUUCPCmd[1]
The command to use to process outbound UUCP jobs. This parameter is not currently used, it is for future development.

ServerTracing[2]
A number that controls the generation of tracing information by a server when not actively sending or receiving facsimile. Tracing is divided into *areas* that are individually controlled. To enable tracing of multiple areas of operation, the flag associated with each area should be bit-or'd to form the value for this tag.

Flag	Area	Description
1 (0x00001)	Server Operation	queue management and general operation
2 (0x00002)	FAX/IXO Protocol	T.30 facsimile protocol or IXO/TAP protocol
4 (0x00004)	Modem Operations	modem hardware manipulation
8 (0x00008)	Modem Communications	comands passed between host and modem
16 (0x00010)	Timeouts	timer operations
32 (0x00020)	Modem Capabilities	modem capabilities
64 (0x00040)	HDLC Frames	binary T.30 HDLC frames
128 (0x00080)	Binary Modem I/O	binary communication between host and modem
256 (0x00100)	Server State Transitions	server program state transitions
512 (0x00200)	Queue Management	job queue management
1024 (0x00400)	Copy Quality	copy quality checking of received facsimile
2048 (0x00800)	Job Management	low-level job management
4096 (0x01000)	IXO Protocol	low-level IXO protocol
8192 (0x02000)	Config File Parsing	unknown configuration file parameters
16384 (0x04000)	FIFO Messages	inter-application messages
32768 (0x08000)	Modem State Transitions	modem state changes (down, busy, ready)
65536 (0x10000)	Dial Rules	dialstring rules parsing and execution
131072 (0x20000)	Docq Changes	document reference handling

For example, to enable tracing of server operations and protocol operations, a value of 1+2=3 should be specified. **NB: tracing timeouts and/or binary modem I/O can adversely affect the operation of the fax server; enabling these areas should be done with extreme care.**

Server tracing is directed to *syslog*(3) using the *facility* specified with the **LogFacility** configuration parameter. Note that *syslogd*(8C) must be configured to capture *facility.info*, *facility.debug*, *facility.warning*, and *facility.err* messages. See *hylafax-log*(5F) for a description of the logged messages.

SessionTracing[2]
A number that controls the generation of tracing information by a server while sending or receiving facsimile. The number is interpreted identically to **ServerTracing.** Note that session tracing is placed in log files in the **log** subdirectory; see *hylafax-log*(5F) for more information.

SpeakerVolume
The volume level that the in-modem speaker should be adjusted to while in command mode. The possible values are:

Value	Description
Off	silent
Quiet	almost inaudible
Low	tolerable in privacy
Medium	loud enough to hear in the next room
High	intolerable

Note that the speaker is always turned off once carrier is established. Also beware that some modems support fewer volume settings; see **ModemSetVolumeCmd**.

TagLineFont

The filename of the font to use in imaging *tag lines* across the top of each transmitted page. Fonts must be stored in the Portable Compiled Font (PCF) format used by the X11 Window System. Filenames are specified relative to the root of the spooling area; e.g. etc/lutRS18.pcf. If no font is specified, if a null filename is specified, or if the specified font file is not readable, then tag lines will not be generated. See also **TagLineFormat**.

TagLineFormat

The format string to use when imaging *tag lines* across the top of each transmitted page. This string may include escape codes that cause various items to be interpolated into the imaged tag line. Any escape code supported by *strftime*(3C) may be used as well as the following server-implemented codes:

Escape	**Description**
%%d	destination phone number
%%i	job identifier
%%j	user-specified job tag
%%l	**LocalIdentifier** or canonicalized **FAXNumber**
%%m	sender’s electronic mail address
%%n	canonicalized **FAXNumber**
%%p	current page number of session
%%P	current page number of job
%%s	sender’s name
%%t	total pages in session
%%T	total pages in job
%%%	“%”

In addition, the format string may indicate that text is to be centered in multiple equal-sized *fields* by separating text with “|” characters. For example, “a|b|c” would cause the tag line to be broken up into three equal-sized areas with the strings “a”, “b”, and “c” centered within each region. The default tag line format string is “From %%n|%c|Page %%P of %%T”. The differences between the %%p or %%P and the %%t or %%T options are noticed when a fax job is retried after an incomplete attempt and only the previously unsent pages are then queued in a successive session. See also **TagLineFont**.

TIFF2FaxCmd[1]

The command to invoke to convert a TIFF submitted for transmission to TIFF/F, suitable for transmission; see *tiff2fax*(8C).

TimeOfDay[1]

The default time-of-day restrictions to apply to outbound jobs. Outbound jobs will be processed only during the specified time. Any jobs submitted outside this time period will be held until the start of this time. The syntax for time-of-day restrictions is designed to be compatible with the syntax used by the UUCP software. The following BNF describes the syntax:

```
   Syntax = tod ["," tod]
      tod = <days><timerange>
     days = "Any" | "Wk" | <dayname>+ | nothing
  dayname = "Sun" | "Mon" | "Tue" | "Wed" | "Thu" | "Fri" | "Sat"
timerange = <start> "-" <end> | nothing
    start = <24hrtime>
      end = <24hrtime>
 24hrtime = {0-9}{0-9}{0-9}{0-9}
  nothing =
```

where *start* and *end* are 24-hour times, day names can be either 2- or 3-characters, and a null day or time specification means any time or day. White space and other syntactic sugar may be freely inserted between tokens but may not be inserted between 24-hour times in the time range.

This value may be overridden by rules in the **DestControls** file.

Use2D[1]
Control the use of 2D-encoded data for transmitted facsimile.

UUCPLockMode[2]
The file protection mode that should be used when creating UUCP lockfiles. Note that this value is given in octal.

UUCPLockDir[2]
The pathname of the directory in which UUCP lockfiles are to be created.

UUCPLockTimeout[2]
The time in seconds to wait before removing a *stale* UUCP lockfile (i.e. a lockfile whose owner does not appear to exist). If this value is set to 0, then the fax server will never remove a stale lockfile.

UUCPLockType[2]
A string that specifies the type of UUCP lockfile to create. The string may be one of ‘‘ascii’’ or ‘‘binary’’ depending on whether the process-ID of the lock owner is written to the file as an ascii string or as a binary value, respectively. In addition, two prefixes may be used to control the format of the lock filename. If the type string is prefixed with a ‘‘+’’, then SVR4-style filenames are generated using the major device number of the tty device and the major and minor device numbers for the filesystem on which the tty device resides. If the type string is prefixed with a ‘‘−’’, then any upper case letters in the device part of the lockfile name are converted to lower case letters; for example, ‘‘LCK..ttyA01’’ is converted to ‘‘LCK..ttya01’’. This upper-to-lower case conversion is useful for systems such as SCO where the convention is to name devices with upper-case letters, but create lockfiles with lower-case letters.

VGettyArgs
A string that indicates whether or not the server should invoke a voice getty program in response to an incoming voice call. If the string value is not null, then it is interpreted as a set of arguments to pass to the *vgetty* program. Before supplying the arguments, the string is first scanned for ‘‘%’’-escape sequences: any appearance of ‘‘%l’’ is replaced with the tty name and any appearance of ‘‘%s’’ is replaced with the serial port speed (in decimal). The ‘‘%’’ character can be specified with ‘‘%%’’. If the **VGettyArgs** parameter is not specified in the configuration file or if the string value is null, then voice connections will be rejected. Note that in addition to the specified command line arguments, the *vgetty* program is invoked with its standard input, output, and error descriptors directed to the controlling tty device.

WedgedCmd[1]
The command to invoke when a modem is deemed ‘‘*wedged*’’; see *wedged*(8C).

MODEM-ORIENTED CONFIGURATION PARAMETERS

This section lists configuration parameters that are specific to the setup and operation of the modem. All *commands* must be specified exactly as they are to be transmitted to the modem (note that this is a departure from previous versions of this software that automatically prepended ‘‘AT’’ to each line of commands sent to the modem). When multi-line commands are required, enclose the commands in quote marks (‘‘"’’) and insert a newline character ‘‘\n’’ where lines should be broken. An example of a multi-line command string is ‘‘AT+FCQ=1\nAT+FBADMUL=20\nAT+FBADLIN=10’’.

Command strings sent to the modem may include command escapes that force the host to alter the DTE-DCE communication rate, the flow control scheme used between the host and modem, delay for a period of time, flush input data, and wait for a specific response from the modem. The following escape codes are interpreted:

Escape	**Effect**
<none>	enable no flow control between DTE and DCE
<xon>	enable software flow control between DTE and DCE
<rts>	enable hardware flow control between DTE and DCE
<2400>	set the DCE-DTE baud rate to 2400 bps
<4800>	set the DCE-DTE baud rate to 4800 bps
<9600>	set the DCE-DTE baud rate to 9600 bps
<19200>	set the DCE-DTE baud rate to 19200 bps
<38400>	set the DCE-DTE baud rate to 38400 bps
<57600>	set the DCE-DTE baud rate to 57600 bps
<76800>	set the DCE-DTE baud rate to 76800 bps
<115200>	set the DCE-DTE baud rate to 115200 bps
<delay:N>	delay N x 10 ms on the host (0 <= N <= 255)
<flush>	flush any pending input data from the modem
<waitfor:R>	wait for the response R from the modem (see below)

Note that commands to the left of an escape code are sent to the modem before the associated actions are carried out on the host. This may be important when changing baud rates, as the result code from a command may be returned at the new baud rate. Also, beware that not all baud rates are supported by all systems and modems; the set of baud rates supported by a host is usually listed in *stty*(1).

The ‘‘<waitfor:..>’’ escape can be used to override the default behaviour which is to wait for an ‘‘OK’’ response to a command sent to the modem. The possible response codes are:

NOTHING	a null response
OK	the default response
CONNECT	‘‘CONNECT...’’
NOANSWER	‘‘NO ANSWER...’’
NOCARRIER	‘‘NO CARRIER...’’
NODIALTONE	‘‘NO DIALTONE...’’
BUSY	‘‘BUSY’’
OFFHOOK	‘‘PHONE OFF-HOOK’’
RING	‘‘RING...’’
ERROR	error status from modem
OTHER	any unrecognized modem response

For example ‘‘ATO\r<waitfor:CONNECT>’’ would send ‘‘ATO\r’’ to the modem and then wait for a response with a leading ‘‘CONNECT’’ in it; and ‘‘ATS99=2\r<delay:2><flush><waitfor:NOTHING>’’ would send ‘‘ATS99=2\r’’ to the modem, delay 20 ms, flush any input from the modem, and then continue (not wait for any response).

ModemAnswerCmd

The command to answer the phone. If possible, this command should generate a result string that identifies whether the incoming call is from a facsimile, voice, or data modem. Typically this auto-detection is configured ahead of time; e.g. for Class 2 modems with the ‘‘AT+FAA=1’’ command.

The following table describes how the result codes are interpreted by the facsimile server.

Result String	**Default**	**Class 1**	**Class 2**	**Class2.0**
CONNECT FAX	Fax	–	–	––
CONNECT	Data	Unknown	–	–
NO ANSWER	No Answer	–	–	–
NO CARRIER	No Carrier	–	–	–
NO DIALTONE	No Dialtone	–	–	–
ERROR	Error	–	–	–
FAX	Fax	–	–	–

DATA	Data	–	–	–
+FCON	–	–	Fax	Fax
+FCO	–	–	Fax	Fax
+FDM	–	–	Data	Data
+FHNG:	–	–	Error	Error
VCON	–	–	Voice	Voice

The ''Default'' column indicates the interpretation made for the result string in either class 1 or 2 unless explicitly overridden. The ''Class 1'' column indicates result strings handled specially for modems operating in class 1, the ''Class 2'' column indicates special handling for modems operating in class 2, and the ''Class 2.0'' column indicates special handling for modems operating in class 2.0. *Unknown* entries imply that interpretation is based on the way in which the call was answered; if answering is done for an explicit call type, that type is assumed, otherwise a Fax call is presumed.

ModemAnswerDataBeginCmd
The command to send to the modem once a data connection has been established.

ModemAnswerDataCmd
The command to explicitly answer the phone for a data connection. This command must generate a result string that identifies the incoming call is from a data modem. If this parameter is not specified or is null, then the value of **ModemAnswerCmd** is used instead. See also **AdaptiveAnswer** and **AnswerRotary**.

ModemAnswerFaxBeginCmd
The command to send to the modem once a facsimile connection has been established.

ModemAnswerFaxCmd
The command to explicitly answer the phone for a facsimile connection. This command must generate a result string that identifies the incoming call is from a facsimile modem. If this parameter is not specified or is null, then the value of **ModemAnswerCmd** is used instead. See also **AdaptiveAnswer** and **AnswerRotary**.

ModemAnswerVoiceBeginCmd
The command to send to the modem once a voice connection has been established.

ModemAnswerVoiceCmd
The command to explicitly answer the phone for a voice connection. This command must generate a result string that identifies the incoming call is for voice. If this parameter is not specified or is null, then the value of **ModemAnswerCmd** is used instead. See also **AdaptiveAnswer** and **AnswerRotary**.

ModemAnswerResponseTimeout
The maximum time, in milliseconds, to wait for a response from the modem after sending **ModemAnswerCmd**. Note that you may want to shorten this value when using the server-based adaptive answer strategy; see **AdaptiveAnswer**.

ModemATCmdDelay
A delay, in milliseconds, that should be performed before each ''AT'' command string is sent to the modem.

ModemBaudRateDelay
The amount of time, in milliseconds, to pause after setting the baud rate on the serial line. This parameter is only needed for hosts where (apparently) setting the serial line parameters does not take effect immediately.

ModemCommaPauseTimeCmd
The command to set the time, in seconds, that the modem should pause when encountering a '','' modifier in the dialing string.

ModemDialCmd
The command to place a phone call. The string is assumed to be suitable for use as a parameter to the *sprintf* (3S) function; so the ''%'' character should be escaped as ''%%''. The dial command must include a single ''%s'' where the number that is to be dialed is to be substituted. Facilities such as tone or pulse dialing, outgoing prefixes to route though a PBX, and so on should be included in the dialing command string. It is also important that, if possible, a trailing ''@'' symbol should be included so that the modem returns result codes that distinguish between no answer and no carrier. Use of the ''@'' permits the server to reduce the probability that a wrong number is continually redialed. If the modem does not support this facility, then it may be necessary to raise the number of retries done when a ''NO CARRIER'' result is returned.

ModemDialResponseTimeout
The maximum time to wait, in milliseconds, for a response from the modem after sending This value should be longer than the timeout programmed into the modem through the **ModemWaitTimeCmd** parameter. This additional server-based timeout is provided to guard against modems that can ''lock up'' when dialing the telephone.

ModemDTRDropDelay
The time, in milliseconds, to pause between placing DTR OFF and DTR ON while resetting a modem. This value should be at least equal to the time-value of the analogous S-register, usually S25.

ModemEchoOffCmd
The command to disable echoing of commands sent to the modem.

ModemFlowControl
The type of flow control to use between DTE and DCE; one of ''NONE'', ''XONXOFF'' and ''RTSCTS''. This value is used to select the string sent to the modem to initially establish DTE-DCE flow control; one of **ModemNoFlowCmd**, **ModemSoftFlowCmd**, and **ModemHardFlowCmd**. The current flow control setting is also used to select the appropriate flow control command to send to the modem when the software switches to Class 1, 2, or 2.0; see **Class1HFLOCmd**, **Class1NFLOCmd**, **Class1SFLOCmd**, **Class2HFLOCmd**, **Class2NFLOCmd**, and **Class2SFLOCmd**.

The server supports both software and hardware flow control for Class 1, Class 2, and Class 2.0 modems. Whether to use hardware or software flow control depends on the capabilities of the modem and the host hardware and operating system. Communication rates above 9600 baud often require that hardware flow control be used for reliable DTE-DCE communication. However, beware that many modems only support software flow control when sending or receiving facsimile.

Note that modems usually support software flow control even if they have no explicit AT-command to activate it; in this case it is switched on when the modem enters fax mode, having AT+FCLASS=... from DTE.

ModemFrameFillOrder
The bit order to expect for received HDLC frames and to use when formulating HDLC frames to send. This value may be either LSB2MSB when bits are ordered from least-significant-bit to most-significant-bit (as in the CCITT specification) or MSB2LSB when bits are ordered in the reverse direction.

ModemHardFlowCmd
The command to setup hardware (RTS/CTS) flow control between DTE and DCE.

ModemMinSpeed
The minimum acceptable signalling rate for transmitting facsimile page data. Possible values are: 2400, 4800, 7200, 9600, 12200, and 14400.

ModemMfrQueryCmd
The command to send to the modem to get the manufacturer identification string. If this parameter is not set, then it is initialized to ‘‘AT+FMFR?’’ for Class 2 modems, or to ‘‘AT+FMI?’’ for Class 2.0 modems, or to ‘‘ATI3’’ for Class 1 modems. If the parameter begins with a ‘‘!’’, then the remainder of the string is taken to be the identification string and no command is issued to the modem.

ModemModelQueryCmd
The command to send to the modem to get the model identification string. If this parameter is not set, then it is initialized to to ‘‘AT+FMDL?’’ for Class 2 modems, or to ‘‘AT+FMM?’’ for Class 2.0 modems, or to ‘‘ATI0’’ for Class 1 modems. If the parameter begins with a ‘‘!’’, then the remainder of the string is taken to be the identification string and no command is issued to the modem.

ModemNoAutoAnswerCmd
The command to stop the modem from automatically answering when the phone rings.

ModemNoFlowCmd
The command to disable flow control between DTE and DCE.

ModemOnHookCmd
The command to place the phone ‘‘on hook’’ (i.e. hangup).

ModemPageDoneTimeout
The maximum time to wait, in milliseconds, for a response from the modem after sending a page of facsimile data (e.g. the time to wait for a response to a Class 2/2.0 AT+FET command).

ModemPageStartTimeout
The maximum time to wait, in milliseconds, for an initial response from the modem when sending a page of facsimile data (e.g. the time to wait for a response to a Class 2/2.0 AT+FDR command).

ModemRate
The baud rate to use for DCE-DTE communication. This value may be one of: 115200, 76800, 57600, 38400, 19200, 9600, 4800, 2400, 1200. The default value is 19200 because many modems lock the rate at 19200 when sending or receiving facsimile. Note that not all values are supported by all operating systems and modems; consult *stty*(1) for the available rates on your system.

ModemRecvFillOrder
The bit order to expect for received facsimile data. This value may be either LSB2MSB when bits are ordered from least-significant-bit to most-significant-bit (as in the CCITT specification) or MSB2LSB when bits are ordered in the reverse direction. According to the various specifications all modems should return data in LSB2MSB order. However most Class 2 modems (except maybe only Multitech) use MSB2LSB for compatibility with modems that were built with Rockwell hardware/firmware that included a bug that was too widespread to correct.

If this parameter is not set, then it is autodetected and set to LSB2MSB for Class 1 and Class 2.0 modems and MSB2LSB for non-Multitech Class 2 modems. However this may be wrong for your modem, so you will have to specify this parameter explicitly.

ModemResetCmds
A string of commands to issue to the modem during initialization. This string is sent to the modem *before* the standard set of configuration commands required by the fax server. Note that these commands **should not include normal reset commands that are specified through other parameters**. For example, commands to setup flow control, DCD handling, DTR handling, etc. should be specified through the appropriate configuration parameters and not through this parameter. In addition the soft reset command (usually ‘‘ATZ’’) should not be included in this string; the servers issue this command explicitly.

ModemResetDelay
The time, in milliseconds, to pause after setting DTR ON, while resetting a modem. DTR ON does not respond with ‘‘OK’’, so this parameter should be long enough to allow the modem time to be ready for **ModemSoftResetCmd** successively.

ModemResultCodesCmd
The command to enable result codes.

ModemRevQueryCmd
The command to send to the modem to get a firmware revision identification string. If this parameter is not set, then it is initialized to ‘‘AT+FREV?’’ for Class 2 modems, or to ‘‘AT+FMR?’’ for Class 2.0 modems. If the parameter begins with a ‘‘!’’, then the remainder of the string is taken to be the identification string and no command is issued to the modem.

ModemSendBeginCmd
The command to send to the modem upon establishing carrier during a transmit operation. This parameter is useful for systems that are incapable of enabling hardware flow control without DCD asserted.

ModemSendFillOrder
The bit order the modem expects for transmitted facsimile data. This value may be either LSB2MSB or MSB2LSB (see also **ModemRecvFillOrder** above.) Virtually all modems expect transmitted facsimile data in LSB2MSB bit order.

ModemSetVolumeCmd
The commands to use to set the modem speaker volume levels. There should be five whitespace-separated commands, one each for setting the volume to ‘‘Off’’, ‘‘Quiet’’, ‘‘Low’’, ‘‘Medium’’, and ‘‘High’’; the default is ‘‘"ATM0 ATL0M1 ATL1M1 ATL2M1 ATL3M1"’’. See also **SpeakerVolume**.

ModemSetupAACmd
The command to setup *adaptive answer* support in the modem−if available. Adaptive answer is the term used for the ability to distinguish between calls from facsimile, voice, and data sources. Note that this string is the last command issued by the device drivers during setup, so the command string may, if necessary, switch to a different operating mode (e.g. on some Rockwell-based modems it is necessary to issue the ‘‘AT+FAA=1’’ command in Class 0).

ModemSetupDCDCmd
The command to setup DCD handling. On most systems the facsimile server will enable the CLOCAL flag on the tty device to which the modem is connected. This should insure that the system does not close an open file descriptor associated with the modem if carrier is dropped. Thus, for most systems and modems **ModemSetupDCDCmd** should setup DCD to reflect carrier.

ModemSetupDTRCmd
The command to setup DTR handling so that the modem is reset when the DTR signal is lowered by the host. The facsimile server uses this facility to insure that modems are not left in a ‘‘locked up’’ state.

ModemSoftFlowCmd
The command to setup software (XON/XOFF) flow control between DTE and DCE.

ModemSoftResetCmd
The command to force a soft reset of the modem.

ModemType
This parameter must be set to one of: ‘‘Class2’’, ‘‘Class2.0’’, or ‘‘Class1’’; to indicate that the modem is a Class 2-, Class 2.0-, or Class 1-style modem, respectively. If this parameter is not set, then it is autodetected and the highest supported fax class is used.

ModemVerboseResultsCmd
The command to enable verbose, as opposed to numeric, result codes.

ModemWaitForConnect
If enabled, the facsimile server will not consider a connection established when answering an incoming call until a ‘‘CONNECT’’ status string is received. This is useful mainly for Rockwell RC32ACL-based modems that send ‘‘FAX’’ and ‘‘DATA’’ status messages before sending ‘‘CONNECT’’.

ModemWaitTimeCmd
The command to set the number of seconds to wait for a carrier signal when placing a call or answering the phone.

T.30 PROTOCOL CONFIGURATION PARAMETERS

The following parameters apply to the implementation of the CCITT T.30 facsimile protocol in the Class 1 device driver. They should not be changed without full understanding of the operation of the server.

FaxT1Timer
The value of the T1 timer in milliseconds. This timer is used to time out the initial session contact; i.e. receipt of DIS/DCS.

FaxT2Timer
The value of the T2 timer in milliseconds. This timer is used to time out receiving responses and commands.

FaxT4Timer
The value of the T4 timer in milliseconds. This timer is used to time out the reception of HDLC frames and, usually, trigger frame retransmissions.

CLASS 1 MODEM CONFIGURATION PARAMETERS

The following parameters are specific to the configuration of Class 1-style modems; they should not be changed lightly:

Class1Cmd
The command to set the modem into Class 1 operation.

Class1FrameOverhead
The number of extraneous bytes in HDLC frames sent to the host in response to an ‘‘AT+FRH’’ command. For modems that properly implement the Class 1 interface, this number should be 4 (the default).

Class1HFLOCmd
The command to setup hardware (RTS/CTS) flow control between DTE and DCE when operating in Class 1. This command is issued immediately after sending the **Class1Cmd** to switch the modem to Class 1 operation.

Class1NFLOCmd
The command to setup no flow control between DTE and DCE when operating in Class 1. This command is issued immediately after sending the **Class1Cmd** to switch the modem to Class 1 operation.

Class1RecvAbortOK
The time, in milliseconds, to wait for an ‘‘OK’’ result code from the modem after aborting an HDLC frame receive (‘‘AT+FRH=3’’). If this number is zero, then the modem is assumed to not correctly implement aborting and instead the driver will wait 200ms, flush any input, and then send ‘‘AT\n’’ and wait 100ms for an ‘‘OK’’ result.

Class1RecvIdentTimer
The time, in milliseconds, to wait for an initial DCS when receiving a facsimile. CCITT recommendation T.30 specifies this as the value of the T1 timer. However, adaptive answering schemes such as that described above under the **AdaptiveAnswer** parameter may require that this timer be shortened.

Class1SendEOPDelay
The time, in milliseconds, to delay between dropping the high speed message carrier and sending the EOP post-page message/command. Empirically, this value needs to be greater than **Class1SendPPMDelay** to prevent modem response timeouts.

Class1SendPPMDelay
The time, in milliseconds, to delay between dropping the high speed message carrier and sending the MPS post-page message/command.

Class1SendTCFDelay
The time, in milliseconds, to delay between sending DCS and TCF.

Class1SwitchingDelay
The time, in milliseconds, to delay when switching the direction of transmission, as recommended by T.31. Default is 150.

Class1SFLOCmd
The command to setup software (XON/XOFF) flow control between DTE and DCE when operating in Class 1. This command is issued immediately after sending the **Class1Cmd** to switch the modem to Class 1 operation.

Class1TCFMaxNonZero
The maximum percentage of non-zero data bytes permitted in an acceptable received TCF. Note that this number does not include any leading non-zero data in the received data. See also **Class1TCFMinRun**.

Class1TCFMinRun
The duration, in milliseconds, of the minimum run of zero data in an acceptable received TCF. This value should be specified according to a 1.5 second transmission of zero data (i.e. it should be between 0 and 1500). See also **Class1TCFMaxNonZero**.

Class1TCFRecvTimeout
The maximum time to wait, in milliseconds, for the first byte of the Training Check (TCF) message data that is received during the training phase of the facsimile reception protocol.

Class1TCFResponseDelay
The time, in milliseconds, to delay between receiving TCF and sending a response on the v.21 carrier to indicate whether or not training was successful.

Class1TrainingRecovery
The time, in milliseconds, to delay after a failed training session. This time is used when sending training and when waiting for DCS and subsequent training during receive.

CLASS 2 AND 2.0 MODEM CONFIGURATION PARAMETERS

The following parameters are specific to the configuration of Class 2- and Class 2.0-style modems:

Class2Cmd
The command to set the modem into Class 2/2.0 operation.

Class2AbortCmd
The command to use to abort an established session. After using this command to abort a session, the fax software will send **ModemOnHookCmd** and then reset the modem by dropping DTR .

Class2APCmd
A largely unused option for modems supporting ITU-T.32, Class 2.1, standards. This command would be used to enable sending and receiving SUB, SEP, and PWD frames.

Class2APQueryCmd
A largely unused option for modems supporting ITU-T.32, Class 2.1, standards. This command would be used to query the capabilities of sending and receiving SUB, SEP, and PWD frames. The value ‘‘none’’ may be used if the modem does not support any **Class2APQueryCmd.**

Class2BORCmd
The command to setup the bit order of sent and received facsimile data. Usually the command ‘‘AT+FBOR=0’’ is used so that data is sent and received in *direct* bit order (LSB2MSB). Some modems, such as the Everex 24/96D, must use reversed bit order for Phase B and D data exchanges to avoid a bug that results in garbage data being placed in the padding of EOL codes. The bit order specified by this command must correspond to the settings of the **ModemSendFillOrder** and **ModemRecvFillOrder** parameters.

Class2BUGCmd
The command to use to enable or disable the tracing of HDLC frames sent and received by the modem. This tracing information should be returned to the host in ‘‘+FHR:’’ and ‘‘+FHT:’’ status strings. Note that many Class 2 modems do not support this facility, which is largely used for diagnostic purposes. The value ‘‘none’’ may be used if the modem does not support any **Class2BUGCmd.**

Class2CIGCmd
The command used to set a polling identifier. This string is inserted into the format ‘‘%s="<id>"’’.

Class2CQQueryCmd
The command to send to the modem to get the copy quality capabilities string. If the parameter begins with a ‘‘!’’, then the remainder of the string is taken to be the capabilities string and no command is issued to the modem; this can be used together with the **Class2CQCmd** to force copy quality checking to be done in the server instead of in the modem. See also **PercentGoodLines** and **MaxConsecutiveBadLines** for parameters used to do server copy quality checking.

Class2CRCmd
The command to use to enable the reception of facsimile.

Class2CQCmd
The command to use to set up parameters for copy quality checking. For example, for an Everex 24/96D modem this parameter might be set to ‘‘AT+FCQ=1\nAT+FBADMUL=20\nAT+FBADLIN=10’’.

Class2DCCCmd
The command used to set modem capabilities. This string is inserted into the format ‘‘%s=vr,br,wd,ln,df,ec,bf,st’’.

Class2DCCQueryCmd
The command to send to the modem to get the Class 2/2.0 capabilities. If the parameter begins with a ‘‘!’’, then the remainder of the string is taken to be the capabilities string and no command is issued to the modem.

Class2DDISCmd
The command to set session parameters before dialing. This string is inserted into the format ‘‘%s=vr,br,wd,ln,df,ec,bf,st’’. Setting this parameter enables support for Class 2 modems that do not properly implement the ‘‘AT+FDIS’’ command by setting up session parameters before dialing the telephone.

Class2DISCmd
The command used to set the current session parameters. This string is inserted into the format ‘‘%s=vr,br,wd,ln,df,ec,bf,st’’.

Class2HFLOCmd
The command to setup hardware (RTS/CTS) flow control between DTE and DCE when operating in Class 2/2.0. This command is issued immediately after sending the **Class2Cmd** to switch the modem to Class 2/2.0 operation. For Class 2.0 operation the default is ‘‘AT+FLO=2’’.

Class2LIDCmd
The command used to set the local identifier string. This string is inserted into the format ‘‘%s="<id>"’’

Class2MINSPCmd
The command used to set the minimum acceptable speed to be negotiated for transmitting page data. This string is inserted into the format ‘‘%s=<speed>’’

Class2NFLOCmd
The command to setup no flow control between DTE and DCE when operating in Class 2/2.0. This command is issued immediately after sending the **Class2Cmd** to switch the modem to Class 2 operation. For Class 2.0 operation the default is ‘‘AT+FLO=0’’.

Class2NRCmd

(*Class 2.0 only*) The command to setup negotiation message reporting. For the correct operation of the Class 2.0 driver this command must enable the reporting of: receiver parameters, transmitter parameters, and ID strings. It is not necessary to enable reporting of non-standard frames for correct operation of the Class 2.0 driver.

Class2PACmd

A largely unused option for modems supporting ITU-T.32, Class 2.1, standards. This command would be used to set up the polling address string enabled by the **Class2APCmd**.

Class2PHCTOCmd

The command to use to set the Phase C timeout parameter (in seconds). The value ‘‘none’’ may be used if the modem does not support any **Class2PHCTOCmd.**

Class2PIECmd

(*Class 2.0 only*) The command to use to control procedure interrupt handling. Procedure interrupts should be disabled because *HylaFAX* does not provide a mechanism for dispatching procedure interrupts to an administrator.

Class2PTSCmd

The command to use to set the received page status code. When copy quality checking is done in the host, this command may be used to control the post-page response delivered to the sender. Beware that some modems do not properly implement this command in which case the server should be configured to not do copy quality check: see the **PercentGoodLines** and **MaxConsecutiveBadLines** parameters to understand how to defeat copy quality checking.

Class2PWCmd

A largely unused option for modems supporting ITU-T.32, Class 2.1, standards. This command would be used to set up the password string enabled by the **Class2APCmd**.

Class2RecvDataTrigger

The character to send to the modem to trigger the transmission of received data from the modem to the host. This character is specified to be DC1 (octal 21) in the draft specification 2388-A and DC2 (octal 22) in the 2388-B specification. Most Class 2 modems accept DC1 or both DC1 and DC2. Some modems however only accept DC2. Note that string parameters may use C-style escape sequences, so DC2, for example, would be specified as ‘‘"\022"’’.

Class2RELCmd

The command to use to enable the delivery of byte-aligned EOL codes in received facsimile. If this parameter is defined, then received facsimile data will be marked to indicate that EOL codes are byte-aligned; otherwise they will be marked as not (necessarily) having byte-aligned codes.

Class2SACmd

A largely unused option for modems supporting ITU-T.32, Class 2.1, standards. This command would be used to set up the destination subaddress string enabled by the **Class2APCmd**.

Class2SendRTC

Whether or not to append an explicit ‘‘Return To Control’’ (RTC) signal to the page data when transmitting. The Class 2 and Class 2.0 specs (i.e. SP-2388-A and TIA/EIA-592) state that the modem will append RTC when it receives the post-page message command from the host; this parameter is provided in case the modem does not correctly implement this.

Class2SFLOCmd

The command to setup software (XON/XOFF) flow control between DTE and DCE when operating in Class 2/2.0. This command is issued immediately after sending the **Class2Cmd** to switch the modem to Class 2/2.0 operation. For Class 2.0 operation the default is ‘‘AT+FLO=1’’.

Class2SPLCmd

The command to use to enable a polling request. The value ‘‘none’’ may be used if the modem does not support any **Class2SPLCmd.**

Class2TBCCmd

The command to use to enable stream-mode communication between the host and modem. The value "none" may be used if the modem does not support any **Class2TBCCmd.**

Class2XmitWaitForXON

Whether or not to wait for an XON character from the modem before sending facsimile data to the modem for transmission. Note that this is only relevant for modems that conform to the Class 2 spec (i.e. SP-2388-A). The Class 2.0 specification states that the host may transmit data immediately upon receiving CONNECT and that no XON character will be sent to the host.

PAGER-RELATED CONFIGURATION PARAMETERS

The following parameters are specific to the configuration and operation of the IXO/TAP and UCP support for sending pager messages and GSM SM. Parameter not used for UCP are marked with "(*IXO/TAP only*)".

PagerSetupCmds

The commands to send to a modem to prepare the modem for a call to pager service provider. Typically these commands place direct the modem to communicate with the service provider at 300 bps using the V.21 protocol. Per-service provider command strings can be setup in the info database; see *hylafax-info*(5F).

PagerMaxMsgLength

The maximum length for a pager text message. Messages longer than this number are truncated. Per-service provider message lengths can be setup in the info database; see *hylafax-info*(5F).

IXOService

(*IXO/TAP only*) The service identification string transmitted as part of the IXO/TAP protocol.

IXODeviceID

(*IXO/TAP only*) The terminal device identification string transmitted as part of the IXO/TAP protocol.

IXOMaxUnknown

The maximum number of unrecognized messages that will be accepted at various stages of the IXO/TAP protocol before the sender will abort and hangup the phone.

IXOIDProbe

(*IXO/TAP only*) The time, in seconds, between sending a "\r" during the initial ID recognition sequence of the IXO/TAP protocol.

IXOIDTimeout

(*IXO/TAP only*) The maximum time, in seconds, to wait for the initial ID response from the service provider.

IXOLoginRetries

(*IXO/TAP only*) The maximum number of attempts to login to a service provider.

IXOLoginTimeout

(*IXO/TAP only*) The maximum time, in seconds, to wait to complete the login procedure.

IXOGATimeout

(*IXO/TAP only*) The maximum time, in seconds, to wait for a Go-Ahead message from the service provider.

IXOXmitRetries

The maximum number of times to try sending a text message block in a single call.

IXOXmitTimeout

The maximum time, in seconds, to try transmitting a text message block.

IXOAckTimeout

(*IXO/TAP only*) integer30IXO: max time to wait for msg block ack (secs) The maximum time, in seconds, to wait for an acknowledgement to a transmitted message.

PROTOTYPE CONFIGURATION FILES

Per-modem configuration files are typically derived from prototype files that have been created for known modems. These prototype files are kept in the **config** subdirectory and, by convention, have names that identify a brand or type of modem and the DTE-DCE flow control scheme the prototype files configures. The *faxaddmodem*(8C) program that is used to configure a modem for use with *HylaFAX* selects a prototype configuration file using information retrieved from the modem and comments embedded in the prototype files. For Class 1 modems the product ID code returned by the command ‘‘ATI0’’ and the response from the comannd ‘‘ATI3’’ are used to select a prototype configuration file, while for Class 2 modems the manufacturer and model as returned by ‘‘AT+FMFR?’’ and ‘‘AT+FMDL?’’, respectively, are used (or ‘‘AT+FMI?’’ and ‘‘AT+FMM?’’ for Class 2.0 modems).

A Class 1 prototype configuration file is identified for use by *faxaddmodem* by searching for a comment of the form:

```
# CONFIG:CLASS1:144:.*:RTSCTS: Manufacturer='AT&T' Model=Dataport
```

In this example ‘‘144’’ is the product ID code for an AT&T DataPort modem, ‘‘.*’’ is a regular expression matched against the result string returned by the ‘‘ATI3’’ command, and ‘‘RTSCTS’’ indicates the modem is configured to use hardware flow control during fax operation. The remainder of the line is evaluated by the *sh*(1) and used to specify the modem’s manufacturer and model (since Class 1 modems do not have standard commands to query this information).

Class 2 and 2.0 prototype configuration files match the string ‘‘*manufacturer-model-flowcontrol*’’ against a *sh*(1) glob pattern specified in the configuration file, where *manufacturer* and *model* are the strings returned by querying the modem and *flowcontrol* is either ‘‘RTSCTS’’ for hardware flow control or ‘‘XONXOFF’’ for software flow control. For example:

```
# CONFIG: CLASS2: ZyXEL*-RTSCTS
# CONFIG: CLASS2.0: USRobotics*-XONXOFF
```

are configuration comments that appear in the prototype file for a ZyXEL 1496E with Class 2 support, and for a US Robotics Courier modem with Class 2.0 firmware.

The *faxaddmodem* program merges server-specific configuration parameters into a prototype configuration according to comments placed in the prototype file. All lines between ‘‘BEGIN-SERVER’’ and ‘‘END-SERVER’’ comments are placed with the appropriate server configuration parameters. Note that this means modem-related configuration parameters must be placed outside this area of the file.

SEE ALSO

faxaddmodem(8C), *faxq*(8C), *faxgetty*(8C), *faxsend*(8C), *hylafax-server*(5F)

NAME

info − remote device capability database

DESCRIPTION

The **info** subdirectory in the *HylaFAX* spooling area holds information about the capabilities and status of devices that have been called. One file exists for each device, with the filename created from the remote device's fully-qualified phone number. Files have a simple ASCII format. Each line is of the form

[**&**]**tag**: *value*

where a *tag* identifies a capability and a *value* is either a string, number, or boolean value. An optional ''&'' at the front of the line indicates that the tag's value is *locked down* and should not be updated by the server. This facility is useful for restricting the capabilities used with a particular machine.

The following items are recorded:

Tag	**Type**	**Description**
calledBefore	boolean	device has previously answered at this number
dialFailures	number	count of consecutive failed dial operations
lastDialFailure	string	reason for last failed dial operation
lastSendFailure	string	reason for last failed send attempt
maxPageWidth	number	maximum page width in pixels
maxPageLength	number	maximum page length in millimeters
maxSignallingRate	string	maximum signalling rate (bits/sec) to use
minScanlineTime	string	minimum scanline time
pagerMaxMsgLength	number	length of longest allowed text message
pagerPassword	string	password string send to service provider
pagerSetupCmds	string	commands for setting up modem for a specific service provider
pagerTTYParity	string	parity & # bits configuration for IXO/TAP communication
pageSource	string	parameter to tell the paging central who we are
pagingProtocol	string	protocol (IXO or UCP) for this provider
remoteCSI	string	remote device Called Subscriber Identification
sendFailures	number	count of consecutive failed send attempts
supportsHighRes	boolean	accepts 196 line/inch images
supports2DEncoding	boolean	accepts Group 2D encoding
supportsPostScript	boolean	accepts Adobe POSTSCRIPT transfer protocol

The *HylaFAX* scheduler, *faxq*(8C), uses the information stored in this directory when deciding if a call should be placed and when preparing documents for transmission. If a remote device's capabilities are known, then document preparation is constrained and/or optimized according to the capabilities. Otherwise, documents are prepared without regard for the remote device's capabilities and if a capability mismatch is encountered the session is terminated and the documents are reformatted using the newly discovered capabilities.

The *calledBefore* item is used by *faxsend* to decide whether or not to redial a phone number when encountering certain errors. For example, if a dialing command results in a ''NO CARRIER'' result, then the number will not be retried unless the number has previously been dialed successfully.

The *minScanlineTime* item indicates the minimum scanline time parameter used for facsimile transmissions. Acceptable values are: ''0ms'', ''5ms'', ''10ms/5ms'', ''10ms'', ''20ms/10ms'', ''20ms'', ''40ms/20ms'', and ''40ms''. (Values of the form *X/Y* mean to use *X* for 98 lpi images and *Y* for 196 lpi images.)

The *pagerTTYParity* item specifies the communication parameters to use when communicating with the IXO/TAP service provider. Acceptable values are: ''even'' (7 bits, even parity), ''odd'' (7 bits, odd parity), and ''none'' (8 bits, no parity). If nothing is specified then ''even'' is used per the standard.

The *pagerMaxMsgLength*, *pagerPassword*, and *pagerTTYParity* items can be setup according to the requirements of a pager service provider; they typically are locked down so that their values are not lost during updates.

The *pagerSetupCmds* can be used to define AT commands for setting up the modem for a specific service provider. This value overwrites the settings defined in *hylafax-config*(5F).

The *pageSource* parameter can be used to tell the paging central who we are. It usually contains a SMS answerable mobile number. It is only used for the UCP part of *sendpage*(8C). **Note:** This parameter should be used from user-land *sendpage*(1) and will be removed later.

The *pagingProtocol* defines which protocol, UCP or IXO, should be used to send the message to the service provider. It defaults to ‘‘ixo’’; for UCP set ‘‘ucp’’.

NOTES

faxsend, *pagesend*, and *faxq* automatically create and update entries in the **info** directory, as required.

The data in this directory is simply a *cache* of remote capabilities and, except for locked down items, can be safely purged at any time. The *faxcron*(8C) script, for example, periodically purges unlocked entries in this directory that have not been accessed in 30 days.

SEE ALSO

faxcron(8C), *faxq*(8C), *faxsend*(8C), *pagesend*(8C), *hylafax-config*(5F)

NAME

log/* − session log files

DESCRIPTION

Files in the **log** directory in the *HylaFAX* spooling area contain logging/tracing information about transmit and receive sessions. One file exists for each inbound or outbound call, with the filename of the form ‘‘cXXXXXXXX’’ where XXXXXXXX is a decimal sequence number termed a *communication identifier*. The amount and kind of tracing information that is recorded for a call is defined by the **SessionTracing** parameter specified in the modem configuration file; see *hylafax-config*(5F).

Note that logging/tracing information generated by the server outside of a session is directed to the *syslog*(3) service and is controlled by the **LogFacility** and **ServerTracing** parameters specified in the modem configuration file.

[**NB: The remainder of this manual page is out of date.**]

SERVER OPERATION

The following messages may appear when server operation tracing (bit 1) is enabled:

MODEM: Unknown type "%s" ignored. The **ModemType** configuration paremeter was not recognized.

MODEM %s %s/%s. This message is generated each time the modem is reset and made ready for use.

%s: Can not lock device.. The UUCP lockfile for the modem could not be created because another process has a lockfile already in place.

%s: Can not open modem (%m). *open*(2) on the special device file for the modem failed; the error status is presented; c.f. *intro*(2).

%s: Can not open modem (timed out).. *open*(2) on the special device file blocked and was timed out. This typically is because the system does not support a non-blocking mechanism for opening the device and the presented interface requires that carrier be present before the open can succeeed.

%s: setuid root failed (%m). The server process was unable to establish root privileges. This is usually caused by server being improperly installed.

%s: fcntl: %m. The server was unable to turn off the O_NDELAY bit on the open modem file descriptor after doing a non-blocking open of the modem device special file.

schedctl: %m. (*Only on Silicon Graphics systems.*) A *schedctl*(2) system call to set the server process priority failed.

sched policy=%d, priority=%d. (*Only on POSIX compliant systems.*) Describes a *setscheduler*(2) system call to enhance real-time job scheduling.

Unable to set %s scheduling parameters: %m. (*Only on SVR4-style systems.*) A *priocntl* system call to set the server process priority failed.

priocntl(%s): %m. (*Only on SVR4-style systems.*) A *priocntl* system call to get the current process scheduling information failed.

sched policy=%d, priority=%d. (*Only on POSIX-compliant systems.*) Indicates a possible change in the system job scheduling due to use of **PriorityScheduling**

%s: Can not deduce modem type.. The server was unable to reset and initialize the modem. This can be caused by many things: wrong device filename, improper cabling, incorrect modem type, incorrect modem configuration, etc.

RECV: Permission denied (unacceptable client TSI). A receive request was denied because the client was denied access through the TSI access control list mechanism; c.f. *tsi*(5F).

RECV: Unable to create file "%s" for received data. The server was unable to create a file for writing the received facsimile data. Check the permission on the **recvq** subdirectory. Beware also that the server uses *tempnam*(3) to create these files and the presence of the TMPDIR environment variable may cause the server to try to create files in a directory other than the **recvq** directory.

RECV data in "%s". This is an information message; the received facsimile is being placed in the specified filename.

RECV: empty file "%s" deleted. A receive operation terminated without any good pages being received; consequently the empty file is deleted.

ACCEPT TSI "%s". The TSI received during a receive operation was accepted and the receive will continue.

REJECT TSI "%s". The TSI received during a receive operation was rejected and the receive will be aborted at the next opportunity.

POLL: Unable to create file "%s" for received data. The server was unable to create a file for the polled data. See also above under "**RECV: Unable...**".

POLL: empty file "%s" deleted. A poll operation terminated without any good pages being received; consequently the empty file is deleted.

internal muckup, lost polling request. This message should not appear; it indicates the server went to do a poll but did not find the user-specified poll request in its job description; c.f. *sendq*(5F).

SEND FAILED: %s. A transmission failed for the reason shown.

SEND: Can not open document file "%s", dirnum %d. The server was unable to open the TIFF document file that was to be transmitted.

SEND: Can not set directory in document file "%s", dirnum %d. The server was unable to access the TIFF directory associated with the next page to transmit. The document file and directory number for the page are shown.

SEND: %s "%s", dirnum %d. A transmission failed because three attempts were made to transmit the same page. The full reason for the failure is displayed along with the document filename and directory number of the page that failed.

SEND: Document is not in a Group 3-compatible format (file %s, compression %d). The TIFF document submitted for transmission was not was not compressed with CCITT T.4 compression; the TIFF compression code number is shown. This is treated as a fatal error and the job is aborted.

REJECT: Document was encoded with 2DMR, but client does not support this data format. The document to be transmitted was encoded using the optional 2DMR form of the CCITT T.4 standard, but the client capabilities indicate this data format is not supported. This is not a fatal error; the server will disconnect and reimage the document using 1DMR encoding.

REJECT: Client does not support document page width, max remote page width %u, image width %lu. The document to be transmitted was imaged using a page width that is not supported by the client. This is not a fatal error; the server will disconnect and reimage the document using the newly discovered client capabilities.

REJECT: High resolution document is not supported by client, image resolution %g line/mm. The document to be transmitted was imaged using 7 line/mm, but the client only supports 3.85 line/mm. This is not a fatal error; the server will disconnect and reimage the document using the newly discovered client capabilities.

REJECT: Client does not support document page length, max remote page length %d mm, image length %lu rows (%.2f mm). The document to be transmitted was imaged using a page length that is not supported by the client. This is not a fatal error; the server will disconnect and reimage the document using the newly discovered client capabilities.

ABORT: job abort requested. The server received a request on a command FIFO file to abort the current send/receive job.

ANSWER: Can not lock modem device. The server went to answer an incoming call, but found the device locked for use by another process. This is the usual scenario that takes place when an outbound call is made through the modem: the echo from data written to the modem causes the fax server to wakeup, only to discover the UUCP lockfile for the outgoing call. The fax server will discard its handle on the modem and wait for the lockfile to be removed.

ANSWER: FAX CONNECTION. The server is initiating service for an incoming facsimile call.

ANSWER: DATA CONNECTION. The server is initiating service for an incoming data call.

ANSWER: Data connections are not permitted. The server rejected an incoming data call because the configured **GettyArgs** parameter is null.

ANSWER: Could not switch modem to data service. The server was unable to initiate service for an incoming data call because of some problem in setting up the modem for data use.

ANSWER: VOICE CONNECTION. The server is initiating service for an incoming voice call.

ANSWER: Could not switch modem to voice service. The server was unable to initiate service for an incoming voice call because of some problem in setting up the modem for voice use.

ANSWER: %s. An error was detected by the modem layer while setting up an incoming call.

GETTY: could not create. The server was unable to create an internal data structure used to track a getty process.

GETTY: can not fork. The server was unable to *fork*(2) a new process in order to startup the getty program.

runGetty::setregid: %m.
runGetty::setreuid: %m. The server was unable to change the process identity before starting up the getty program.

GETTY: start pid %u, "%s". The server has started up a getty program to service the incoming data call. The getty process identifier is displayed along with the arguments passed to the program.

GETTY: terminated, exit status %#o. The server has reaped a getty process which it started up.

Parse error in dial string rules "%s". The dial string rules in the specified file contain a syntax error. The *dialtest*(8C) program should be used to debug the problem.

Unknown answer type "%s". The **AnswerRotary** configuration parameter has an invalid answer type string; consult *hylafax-config*(5F).

Syntax error, missing ':' in "%s".
Syntax error, missing quote mark in "%s".
Unknown configuration parameter "%s" ignored. A syntax error was encountered in the server configuration file or in a message received through the FIFO special file.

FAX PROTOCOL

The following messages may appear when the facsimile protocol tracing (bit 2) is enabled:

%s: %s (Phase B). An error occurred during Phase B of Group 3 protocol processing.

DELAY %u ms. A %u-millisecond delay was done on the host.

DIAL %s. The phone number %s was dialed. Note that this is the phone number passed as an argument to the *ModemDialCmd* parameter and so may not include additional dialing prefixes sent to the modem.

Error sending T.30 prologue frames. An error occurred while trying to transmit the T.30 DCS frames.

HDLC frame too short (%u bytes). An HDLC protocol message was not at least 3 bytes long.

HDLC frame with bad address field %#x. An HDLC protocol message was sent/received with an address field other than 0xff.

HDLC frame with bad control field %#x. An HDLC protocol message was sent/received with an invalid control field.

MODEM ERROR: %s. While waiting for a modem response/result an error was encountered (e.g. an ‘‘ERROR’’ result code).

MODEM protocol botch ("%s"), cannot parse PPR. The Class 2 modem driver encountered an error parsing the post-page response code in a “+FPTS:” status message.

MODEM protocol botch, can not parse "%s". The Class 2 modem driver encountered an error parsing the displayed session capabilities string. This is usually due to a communication problem between host and modem (though it can also be caused by bogus modem firmware).

MODEM protocol botch: OK without +FPTS:. The Class 2 modem driver encountered an error in the host-modem protocol for receiving facsimile. While waiting for the post-page status (“+FPTS:”), an “OK” result was received first. This should never happen.

MODEM protocol botch: +FET: without +FPTS:. The Class 2 modem driver encountered an error in the host-modem protocol for receiving facsimile. While waiting for the post-page status (“+FPTS:”), a “+FET:” status message was received first. This should never happen.

POLL with CIG "%s". A poll operation is beginning; the Calling Subscriber Identification string that is transmitted to the peer is displayed.

POLL: begin. The server is beginning a polled receive operation.

POLL: end. The server just completed a polled receive operation.

POLL: %s (begin). The specified error message occurred when initiating the poll operation.

POLL: %s (end). The specified error message occurred while terminating the poll operation.

Problem receiving post-page HDLC frame. An error occurred while receiving the HDLC frame that is expected to hold the post-page message.

Problem sending TCF data. An error occurred while sending the TCF data during training.

RECV PPM: %s. The specified post-page message was received during a receive operation.

RECV PPR: %s. The specified post-page response was transmitted during a receive operation.

RECV training at %s %s. During a receive operation, begin training at the specified modulation and signalling rate.

RECV: begin. The server is beginning a receive operation.

RECV: end. The server just completed a receive operation.

RECV: %d bytes of data. During a receive operation the specified number of bytes of facsimile data were received for the current page.

RECV: %lu bytes, %lu total rows. During a receive operation the specified number of bytes of facsimile data were received. The current running total of rows in the current page being received is also displayed.

RECV: %s (begin). The specified error message occurred when initiating the receive operation.

RECV: %s (end). The specified error message occurred while terminating the receive operation.

RECV: TCF %u bytes, %u%% non-zero, %u zero-run. When receiving training, the received TCF had the specified number of bytes of which a certain percentage was non-zero and the longest consecutive run of zero data was found.

RECV: reject TCF (too many non-zero, max %u%%). The received TCF was deemed unacceptable because there was too high a percentage of non-zero data in it. The **Class1TCFMaxNonZero** configuration parameter controls this decision.

RECV: reject TCF (zero run too short, min %u). The received TCF was deemed unacceptable because the longest run of zero data was too short. The **Class1TCFMinZero** configuration parameter controls this decision.

RECV: bad code word 0x%x, len %d, row %lu.

RECV: begin page. During a receive operation a new page of data is about to received on the high speed message carrier.

RECV: end page. During a receive operation, receipt of page of data on the high speed message carrier just completed.

RECV: premature EOF. The server did not receive <DLE><ETX> to signal the end of a page of facsimile data during a page receive operation.

REMOTE supports POSTSCRIPT transfer.
REMOTE does not support POSTSCRIPT transfer. The peer does or does not support the Adobe POSTSCRIPT transfer extensions to the T.30 protocol.

RECV: Missing EOL after 5 seconds. The server timed out while receiving page data. This typically is caused by a flow control configuration problem that causes data from the modem to the host to be stopped.

RECV: %u bytes of data, %lu total lines. While receiving page data, %u bytes of raw data were received; the current total number of rows in the page are also displayed.

RECV: %lu bad lines %lu consecutive bad lines. After the completion of a page receive, the total number of rows received in error and the maximum consecutive rows that were received in error are printed. These numbers should always be zero unless copy quality checking is done in the host.

RECV: REJECT page quality, %u%% good lines (%u%% required). When the server does copy quality checking; the received page is deemed to have unacceptable quality because the percentage of good rows in the page is too low. This check is controlled by the **PercentGoodLines** configuration parameter.

RECV: REJECT page quality, %u-line run (max %u). When the server does copy quality checking; the received page is deemed to have unacceptable quality because the maximum consecutive rows that were received in error exceeds the threshold specified by the **MaxBadLines** configuration parameter.

REMOTE HANGUP: %s (code %s). The modem returned a hangup status code (‘‘+FHNG:’’ or ‘‘+FHS:’’) to the host. This indicates a protocol problem that was determined **within the modem**; there is usually nothing that the server can do to work around such errors.

REMOTE TSI "%s". During a receive operation, the specified TSI was received. This string is checked against the TSI access control list described in *tsi*(5F).

REMOTE best format %s. The peers’ ‘‘best’’ data format according to the received DIS frame. ‘‘Best’’ here is based on the belief that 2DMMR data is better than 2DMR with optional uncompressed data, is better than 2DMR data, is better than 1DMR data.

REMOTE best rate %s. The peers’ best signalling rate according to the received DIS frame.

REMOTE best vres %s. The peers’ best vertical resolution according to the received DIS frame.

REMOTE has document to POLL. The peer indicates it has a document that is available for retrieval.

REMOTE max %s. The peers’ maximum page width/length according to the received DIS frame.

REMOTE supports error correction. The peer supports the optional Error Correct Mode (ECM) according to the received DIS frame.

REMOTE wants %s. The received DCS frame indicates the peer wants the specified capability.

SEND file "%s". The specified file is about to be transmitted.

SEND 1D RTC.
SEND 2D RTC. A 1D- or 2D-encoded Return To Control (RTC) packet was sent to terminate the transmission of a page of facsimile data.

SEND begin page. A page of facsimile data is about to be transmitted on the high speed message carrier.

SEND end page. The transmission of a page of facsimile data on the high speed message carrier was completed.

SEND training at %s %s. Training is about to be sent to the peer using the specified modulation and signalling rate.

SENT %u bytes of data. The specified amount of raw facsimile page data was transmitted on the high speed message carrier.

TRAINING failed.
TRAINING succeeded. Peer training succeeded or failed. When sending this is based on a message returned by the peer. When receiving this is based on an analysis of the received TCF data.

USE %s. Use the specified session capability during a send or receive. The capability is based on the peers' best capabilities, the capabilities of the modem, and, when transmitting, the characteristics of the document.

MODEM OPERATIONS

The following messages may appear when modem operation tracing (bit 3) is enabled:

MODEM set baud rate: %d baud, input flow %s, output flow %s. The DTE-DCE communication rate was set %d bits/sec on the host and the specified flow control method was setup for data received from the modem (input) and transmitted to the modem (output).

MODEM set baud rate: %d baud (flow control unchanged). The DTE-DCE communication rate was changed without modifying the current flow control setup.

MODEM set XON/XOFF/NOW: input %s, output %s.
MODEM set XON/XOFF/DRAIN: input %s, output %s.
MODEM set XON/XOFF/FLUSH: input %s, output %s. The software flow control setup for data exchanged between host and modem was changed. The NOW , DRAIN , and FLUSH , actions indicate whether the operation is done immediately, after any pending data is transmitted to the modem, or after pending data is discarded. On input XON/XOFF characters are either *ignored* (passed through to the server uninterpreted) or *interpreted* (by the system's terminal driver in order to control the flow of data). Likewise, on output XON/XOFF characters are either *disabled* (not generated by the terminal driver), or *generated* (by the terminal driver to effect flow control of data from the modem to the host).

MODEM input buffering enabled.
MODEM input buffering disabled. On systems that buffer input data in the operating system before delivering it to applications the server will enable and disable buffering as best it can in those cases where low input latency is critical.

TIMEOUT: %s. A timeout occurred while doing a modem-related operation; typically reading or writing data.

Error #%u reading from modem. An error was returned by a *read*(2) system call. This usually happens when the modem prematurely drops the DCD signal.

MODEM COMMUNICATION

The following messages may appear when modem communication tracing (bit 4) is enabled:

--> [%d:%s]. A %d-byte line of data was received from the modem. Printable data are displayed.

<-- data [%d]. %d bytes of unprintable data were sent to the modem, typically facsimile data.

<-- [%d:%s]. %d bytes of printable data were sent to the modem. The data are displayed.

TIMEOUTS

The following messages may appear when timeout tracing (bit 5) is enabled:
start %d second timer,
stop timer (stop the current timeout before it expires), and
stop timer - timer expired (the current timeout expired.)

HDLC FRAMES

When tracing of raw HDLC frames (bit 6) is enabled each frame sent and received is logged: **<-- HDLC<%u:%.*s>** for sent frames and **--> HDLC<%u:%.*s>** for received frames. These messages indicate that a %u-byte long HDLC frame was sent or received. The first byte should always be 0xff, the address field specified for dialup communication. The second byte should be a frame control field. The contents of the frame is displayed as sequence of bytes in hexadecimal.

BINARY I/O

When binary input/output (bit 7) is traced there are two types of messages logged: **<-- <%u:%.*s>** for data sent to the modem and **--> <%u:%.*s>** for data received from the modem. Each message indicates that %u bytes of data were transferred. The data is displayed as a sequence of bytes in hexadecimal.

SERVER STATE TRANSITIONS

Bit 8 controls the tracing of state transitions in the facsimile server. There are currently eight states:

State	Description
BASE	uring startup and initialization
RUNNING	normal operation
MODEMWAIT	waiting for the modem to come ready after a reset
LOCKWAIT	waiting for a UUCP lockfile to be removed
GETTYWAIT	waiting for a *getty* process to terminate
SENDING	transmitting a facsimile
ANSWERING	answering an incoming telephone call
RECEIVING	receiving a facsimile

STATE CHANGE: %s -> %s (timeout %ld). The server is changing its internal state. The timeout (in seconds) is used when polling for asynchronous events such as a modem becoming ready.

STATE CHANGE: %s -> %s. The server is changing its internal state without starting a timeout.

QUEUE MANAGEMENT

Bit 9 controls the tracing the management of the job queue.

SEND NOT READY: "%s" in %s. It was not time to send the specified job; it will be requeued for the indicated time interval.

SEND BLOCKED BY CONCURRENT CALL: "%s" in %s. The specified job is blocked by another call to the same phone number; it will be requeued for the indicated time interval.

SEND BLOCKED BY CONCURRENT JOB: "%s" by "%s". The specified job is blocked by another job to the same phone number; it will be requeued for the indicated time interval.

JOB "%s". The specified job is about to be processed.

CONVERT POSTSCRIPT: "%s". The server is about to invoke the POSTSCRIPT interpreter to image a document; the shell command used is displayed.

JOB TIMER START %s. The timer used to control scanning of the job queue is started with the displayed value.

JOB TIMER STOP (Q head %s tts %s). The timer used to control scanning of the job queue is stopped. The job at the head of the queue and that job's time-to-send are displayed.

JOB TIMER STOP (Q empty). The timer used to control scanning of the job queue is stopped. The job queue was found to be empty.

COPY QUALITY

When doing copy quality checking during a receive operation the following messages may be logged if bit 10 is set:

RECV/CQ: adjusting for trailing noise (%lu run). The total number of rows of data in the received page is adjusted (downward) by %lu. This is done to handle the case where garbage data are received after the RTC resulting in some number of consecutive bad rows after the page data.

RECV/CQ: Invalid %s code word, row %lu, x %d. An invalid 1D- or 2D- T.4 code word was encountered on the specified row and horizontal offset within the row. When this occurs the server will scan for an EOL code and then regenerate the bad row with the last good row of data.

RECV/CQ: Premature EOL (%s), row %lu, x %d. An EOL code was encountered before a complete row of pixels was received. When this occurs the server will discard the bad row and replace it with the last good row that was received.

RECV/CQ: Bad %s pixel count, row %lu, x %d. A row of data was received with pixel count greater than the negotiated page width. When this occurs the server will discard the bad row and replace it with the last good row that was received.

RECV/CQ: Panic, bad %s decoding state, row %lu, x %d. The decoder state machine was found in a bad internal state. When this occurs the server will scan for an EOL code and regenerate the row using the last good received row.

SEE ALSO

faxq(8C), *faxgetty*(8C), *faxsend*(8C), *hylafax-config*(5F)

NAME

HylaFAX − introduction to *HylaFAX* server operation and file formats

DESCRIPTION

HylaFAX is a system for sending and receiving facsimile. It supports queued transmission and asynchronous reception of facsimile. Ancillary programs are invoked by the system for flexibility and configurability. *HylaFAX* includes client and server programs to support remote submission of jobs for transmission, remote removal of queued jobs, and to remotely query the status of jobs queued for transmission. This document describes the organization of the filesystem spooling area in which *HylaFAX* server and server-related processes operate, and introduces the various files that reside in the spooling area.

OVERVIEW

The spooling area is typically located under the directory */var/spool/hylafax*. Ancillary command scripts used by the server programs *faxq*(8C), *faxsend*(8C), *pagesend*(8C), and *faxgetty*(8C) are located in the **bin** subdirectory. Configuration, access control, and accounting information are maintained in the **etc** and **config** subdirectories. Outgoing jobs are described by files in the **sendq** subdirectory, while received facsimile are deposited in the **recvq** subdirectory. The **docq** and **temp** subdirectories are also used in the preparation of outbound jobs; the latter holds files that may be freely purged while the former holds client files that may reside on the server independent of an associated job. The **doneq** subdirectory holds jobs that have completed but have not yet been purged or archived. On systems with job archival support, completed jobs that have been archived are placed in the **archive** subdirectory. The **pollq** subdirectory holds documents that are available for polled retrieval from the server. The **info** subdirectory contains files that describe the capabilities of facsimile machines that *HylaFAX* has called−this information is used in preparing documents for transmission. The **status** subdirectory contains files that server processes write their current status to. The **log** subdirectory contains logging information about send and receive sessions. The **client** subdirectory contains FIFO special files used for communication with *faxq*.

HylaFAX supports multiple modems on a host. A single process acts as central queueing agent for all outbound jobs. Typically each modem also has a server process that monitors the modem status and handles inbound phone calls. Per-modem server processes communicate with the central queueing agent using FIFO special files; see *mknod*(2) or *mkfifo*(2). Any other synchronization between server processes is done using file-level locking. The *faxq* process listens for commands written to the file named **FIFO**, while each *faxgetty* process listens for commands written to a per-device file named **FIFO**.*devid* (where *devid* is an identifier derived from the name of the device special file to which the modem is connected; e.g. *ttym2* for */dev/ttym2*, *term_10* for */dev/term/10*.) To send a command to the queueing agent, one writes to **FIFO**. This is useful, for example, for submitting a job for transmission. To send a command to a specific *faxgetty* process, the **FIFO**.*devid* file is used.

Client applications interact with a *HylaFAX* server machine using a communications protocol implemented by the *hfaxd*(8C) program. The *hfaxd* program is typically started at system startup; it listens for client requests for service and creates a process for each client. *hfaxd* supports the submission of outbound jobs, querying the status of the send and receive queues, and altering parameters of previously submitted jobs. The *hfaxd* processes communicate with the *faxq* process through FIFO special files. Commands sent to *faxq* are sent to **FIFO** and responses are received on FIFO files that each *hfaxd* creates in the **client** subdirectory.

SETUP

Each *HylaFAX* server machine must run the *faxsetup*(8C) command prior to starting up *HylaFAX* server processes. *faxsetup* verifys that the *HylaFAX* software has been installed correctly and that any parameters that were specified at the time the software was built are appropriate for the system.

SENDING

Each outgoing facsimile job has a job description file that is located in the **sendq** subdirectory. This file contains all the information necessary to manage the transmission; c.f. *sendq*(5F). The actual documents that are to be sent are usually located in the **docq** subdirectory (though it is also possible to reference documents from the **recvq** directory). *HylaFAX* accepts POSTSCRIPT, PDF, PCL, and TIFF documents for transmission (though support for PCL documents is not currently implemented). Documents are automatically converted to TIFF/F documents prior to transmission according to the capabilities of the remote facsimile machine: maximum page width and length, ability to handle 2D-encoded data, and ability to handle high resolution (7 line/mm) data. This remote machine capability information is stored in files in the **info** subdirectory. If a machine has not been called before, *HylaFAX* assumes the remote machine has the requested capabilities. If a capabilities mismatch is detected while sending a facsimile *HylaFAX* will disconnect and reconvert the submitted documents according to the newly discovered capabilities. Users may also restrict the session parameters used to format documents on a per-job basis.

The actual transmission is handled by a *faxsend*(8C) process that is initiated by the scheduler. This program may be substituted for by specifying the **FaxSendCmd** configuration parameter in the *faxq* configuration file.

While a job is being processed by a server process, its job description file is locked for exclusive use with *flock*(2). The *hfaxd*(8C) program uses this information to tell if a job is actively being processed.

If the **SessionTracing** parameter in a server's configuration file is non-zero, then tracing information for an outgoing job will be logged in a file in the **log** subdirectory. Each destination machine has a separate log file named by its canonical phone number.

The remote job submission facility includes host and user access control. The file **etc/hosts.hfaxd** must be present and list those hosts and users that are permitted to queue jobs for transmission or do other operations that alter the status of a job. Note that it is necessary to include the ''local host'' definition (usually 127.0.0.1) if local submission is to be permitted. For more information consult *hosts.hfaxd*(5F).

There are a number of controls on outbound calls that can be specified using the **etc/destcontrols** file (or whatever file is specified in a **DestControls** configuration parameter in the *faxq* configuration file). This file, described in *destctrls*(5F), allows an administrator to restrict calls by phone number and to control the time of day that calls may be placed. In addition, operational parameters such as the maximum number of pages in a facsimile transmission can be constrained on a per-destination basis.

If an error is encountered during transmission and a subsequent retransmission would not include the original cover page, then *HylaFAX* can be configured to generate a *continuation cover page* that is prepended to the retransmitted pages. Such cover pages are usually generated by the **bin/mkcover** command; though the exact command to use can be specified in the configuration file read by *faxq*.

HylaFAX can be configured to generate a line of status information across the top of each page of an outbound facsimile. This information, termed a ''tagline'', typically includes the sender's identity (i.e. phone number), the time and date of the transmission, and the page number. The exact format of the tagline is configurable and applications can override the default configuration parameters on a per-job basis. Note that in the United States the law requires that a tagline that identifies the sender's phone number must appear on each transmitted page of facsimile.

Facsimile transmitted to receivers that accept variable-length pages may have short pages ''*chopped*''. That is, if a page has a significant amount of trailing whitespace and the receiver accepts variable-length pages then only the top part of the page will be transmitted. *faxq* can be configured so that only the last page of each document is potentially chopped, all pages are potentially chopped, or chopping is disabled. The minimum whitespace threshold is also configurable. Applications can override the default configuration parameters on a per-job basis.

RECEIVING

faxgetty server processes can be configured to answer incoming phone calls and automatically receive facsimile. Received documents are placed in the **recvq** subdirectory as TIFF Class F files. The *faxgetty* processes can be configured to make these files publicly accessible, or they can be made private in which case an administrator must manage their delivery and/or the assignment of ownership to particular users. When a facsimile is received, the *faxgetty* process usually invokes the **bin/faxrcvd** command; though the exact command to invoke can be specified in the per-modem configuration file. The default *notify* command is a shell script that sends a mail message to a well known user, the *FaxMaster*, but one might also, for example, automatically spool the document for printing.

HylaFAX supports a simple form of access control for receiving facsimile. Each *faxgetty* process may be configured to check the Transmission Subscriber Identifiers (TSI) of the remote fax machine against an access control list, typically **etc/tsi**. Only if the TSI is matched by a regular expression pattern in the file, is the remote machine permitted to transmit a document. This mechanism can be used, for example, to guard against *junk fax*.

HylaFAX can be configured to do *copy quality checking* on received facsimile data. When this feature is enabled *faxgetty* decodes and analyzes the received facsimile data as it is received. If data is received with too many errors, according to the setting of the **MaxConsecutiveBadLines** and **PercentGoodLines** configuration parameters, then the sender will be told to retransmit the page. When copy quality checking is enabled it is also possible to force received facsimile data to be saved with a different compression scheme than was used for transmission. This function is known as *transcoding* and can significantly reduce the space needed to store received facsimile.

POLLING

HylaFAX supports the polled retrieval of facsimile documents. Documents that are received because of a poll request are stored in the **recvq** subdirectory and also delivered directly to the requester using the **bin/pollrcvd** command; though the exact command to invoke can be specified with the **PollRcvdCmd** configuration parameter. The **pollrcvd** script typically encodes the binary facsimile data and returns it to the user via electronic mail.

INBOUND CALL HANDLING

In environments where Caller-ID information is available, *HylaFAX* also supports a call screening facility similar to the TSI access control facility. *faxgetty* can be configured to check the phone number of each caller against an access control list, typically **etc/cid**. Only if the number is matched by a regular expression pattern in the file is the call answered. All Caller ID information is logged, irregardless of whether or not it is used to screen incoming calls.

faxgetty is also capable of using *distinctive ring* information to identify whether an inbound call is voice, data, or fax. Consult the **RingData**, **RingFax**, and **RingVoice** parameters in *hylafax-config*(5F) for a description of this facility.

DATA CALLS

Most fax modems also support non-facsimile communication. *HylaFAX* uses the locking mechanism employed by *uucp*(1C), *cu*(1C), *slip*(8C), and *ppp*(8C). Any *faxgetty* processes will transparently ‘‘get out of the way’’ when an application wants to use a modem for an outgoing call. In addition, *HylaFAX* can be configured to deduce whether an incoming call is for facsimile or data use. If a call from a data modem is recognized and the **GettyArgs** parameter is specified in the configuration file, *faxgetty* will invoke the *getty*(8C) program so that caller may login to the system. Similar functionality is also available for invoking a ‘‘voice getty’’ process, though auto-detection of inbound voice calls is less extensive.

STATUS

HylaFAX maintains status information in several forms. General status information for each server process is maintained in the **status** subdirectory and returned to users by the *faxstat*(1) program. The *syslog*(3) facility is used by all server processed for logging status and error diagnostics. The server processes may also be configured to log various kinds of debugging and tracing information; c.f. the **ServerTracing** parameter description in *hylafax-config*(5F).

Any problems encountered when transmitting a facsimile are described in messages returned to the user by electronic mail. A user may also request notification by mail when a job is requeued; for example, because a call failed. Notification by electronic mail is implemented by the **bin/notify** command script; though the name of the script may be overridden with the **NotifyCmd** configuration parameter.

The *faxstat* utility provides (remote) status of jobs queued for transmission, jobs received, and the general status of server processes.

The file **etc/xferfaxlog** contains status information about all facsimile sent and received on a machine. This file is in a simple ASCII format that is easy to manipulate with programs such as *awk*(1), to generate accounting information. See *xferfaxlog*(5F) for information about the format. See *xferfaxstats*(8C) and *recvstats*(8C) for example scripts that print summarized accounting information.

Finally, the *hfaxd* process supports a event monitoring facility that can be access via the *faxwatch*(8C) program. This facility permits clients to register interest in various events and receive ‘‘realtime notification’’ when such events occur on the server. Using this facility it is/should-be simple to construct applications that do things like monitor modem status and use.

MODEM STATE CHANGES

In normal operation each modem is managed by a *HylaFAX* server process such as *faxgetty*. These processes communicate with the central scheduler process to notify it when a modem is ready for use, busy for outbound use, or possibly in an unusable state (either purposely marked unavailable or potentially found to be wedged). Modem usage can be explicitly controlled with the *faxstate*(8C) program. The *faxconfig*(8C) program can also be used to dynamically make changes to configuration parameters that may cause a modem to be treated differently (e.g. setting **RingsBeforeAnswer** to zero will cause *faxgetty* to not answer incoming calls).

When *HylaFAX* is used in a send-only configuration there are no *faxgetty* processes and communication must be done directly with the *faxq* process. The *faxstate* program can still be used to manipulate modem use for outbound jobs but the *faxconfig* program is not frequently needed.

JOB SCHEDULING

Outbound jobs are scheduled by a single process. Jobs have a ‘‘scheduling priority’’ that is assigned at the time the job is submitted. This priority can be changed at any time the job is not actively being processed using the *faxalter*(8C) program. A job’s scheduling priority may also be altered by *faxq* in response to certain scheduling events (e.g. after a failed attempt to send).

Modems are assigned to outbound jobs if they are deemed ready for use. Modem readiness is usually communicated to *faxq* by per-modem *faxgetty* processes. In a send-only environment however this is not possible; instead modems configured for use with *faxmodem* are considered always ready for use unless they are presently assigned to an outbound job or their state is explicitly changed through the *faxstate*(8C) program (*faxstate* can also be used in a send-recv environment).

Each modem has a ‘‘modem priority’’ in the range [0..255]. Modems with a lower priority number are assigned to outbound jobs first. Modem priority is statically configured through configuration files, the *faxmodem* program, and the *faxconfig* program. If multiple modems share the same priority value, then *faxq*(8C) will allocate jobs to them in a round-robin fashon.

JOB MANAGEMENT

Outbound jobs are considered to be in one of several states that reflect their treatment by the central scheduling process. Jobs are initially created in a *suspended* state, and may be returned to this state at any time that they are not actively being processed (e.g. a *faxsend* program is running to process the job). Jobs that are suspended are not processed by the scheduler; and their internal state may be safely altered by the owner or a system administrator. Jobs that are ready for processing by the scheduler are ‘‘submitted’’ and their state is changed to be either *pending* (delayed waiting for a future time to send), *sleeping* (delayed waiting for a scheduled timeout), *blocked* (delayed by concurrent activity to the same destination), or *ready* (ready for transmission, waiting only for available resources). When a job is actively processed by the *faxsend* program its state is marked *active*. Jobs that have completed, either successfully or unsuccessfully are placed in a *done* state and their job description files are moved to the **doneq** subdirectory. Clients may still access the state of jobs that are done; until a ‘‘cleaner process’’ either purges them from the system or archives their state. This delayed removal of a completed job’s state permits clients to resubmit failed jobs using previously transmitted documents and other job state information. The exact mechanics of how and when done jobs are processed is system-dependent; for example, how long a job is left in the done queue before being purged, and whether job archival support is present.

CONFIGURATION

HylaFAX server programs read configuration information from a configuration file. Multiple files are used, one for the *faxq* program and one for each modem. Long-running server programs all automatically re-read their configuration file if it is modified. Typically this re-reading is done frequently enough that administrators do not need to be aware of exactly when it takes place. However in some esoteric cases the file may not be read when expected (the one important case is that the *faxgetty* process reads its configuration file only when answering a call or when resetting a modem; this means that it will not recognize changes when the modem is idle).

In addition to the static configuration files, *HylaFAX* server programs accept commands on their FIFO special files to alter configuration parameters in the running executable (the *faxconfig*(8C) program can be used to dynamically change configuration parameters). Values set in this way however are lost when the process exits or if the configuration file is re-read.

NOTES

Automatic routing of incoming facsimile is desirable.

FILES

FIFO	fifo for job submission
FIFO.<devid>	fifo for communicating with a faxgetty process
/usr/sbin/faxinfo	command that prints information about received facsimile
/usr/sbin/faxquit	command to force server to quit
bin/faxrcvd	faxd command for handling newly received facsimile
bin/mkcover	faxd command for generating continuation cover pages
bin/notify	faxd command for doing user notification
bin/pollrcvd	faxd command for delivering facsimile received by poll
bin/ps2fax	faxd command for converting POSTSCRIPT to TIFF
docq/doc*	documents available for transmission
etc/setup.cache	server setup file created by *faxsetup*
etc/cid	caller id access control list
etc/config.<devid>	configuration data for <devid>
etc/hosts.hfaxd	hosts that may submit jobs for transmission
etc/tsi	fax machine receive access control list
etc/xferfaxlog	log of facsimile sent and received
info/*	data base of remote fax machine capabilities
client/*	FIFO special files created by client processes
config/*	prototype configuration files used by *faxaddmodem*
log/*	session logging records
recvq/fax*	received facsimile
sendq/q*	descriptions of jobs queued for transmission
doneq/q*	descriptions of jobs that are done
status/*	server status information
tmp/*	temporary files created when submitting a job
archive/*	database of archived jobs

SEE ALSO

faxsetup(8C), *faxq*(8C), *faxgetty*(8C), *hfaxd*(8C), *faxsend*(8C), *faxrcvd*(8C), *faxconfig*(8C), *faxmodem*(8C), *faxstate*(8C), *notify*(8C), *pollrcvd*(8C), *recvstats*(8C), *xferfaxstats*(8C), *archive*(5F), *hylafax-config*(5F), *dialrules*(5F), *doneq*(5F), *hosts.hfaxd*(5F), *hylafax-info*(5F), *hylafax-log*(5F), *tsi*(5F), *recvq*(5F), *sendq*(5F), *status*(5F), *xferfaxlog*(5F),

Extensive documentation is available in online at http://www.hylafax.org/. Many of these materials are also included in the software distribution.

NAME

etc/shutdown − server shutdown control file

DESCRIPTION

If the file **etc/shutdown** is present it describes when a *HylaFAX* server is to be shutdown and may contain a descriptive message to return to clients explaining the reason for the shutdown. This file is used by the *hfaxd*(8C) program.

The file has the following format:

<*year*> <*month*> <*day*> <*hour*> <*minute*> <*deny*> <*disconnect*>
<*shutdown message text*>

The first line defines the date and time the server should be shutdown and the time preceding at which unprivileged clients should be denied service or disconnected; these fields are described in more detail below. Any lines that follow the first are taken as the text of a message to return to clients when notifying them of a scheduled shutdown or when denying them service.

<*year*> The decimal year; e.g. 1996.

<*month*> The decimal month; e.g. 10 for October.

<*day*> The decimal day of the month in the range [1..31].

<*hour*> The decimal hour since midnight; in the range [0..23].

<*minute*> The decimal minute after the hour; in the range [0..59].

<*deny*> The time prior to the shutdown time at which to deny service to unprivileged clients. This value is specified as decimal number in the form HHMM; e.g. 130 for 1 hour and 30 minutes.

<*disconnect*>
Time time prior to the shutdown time at which to disconnect unprivileged clients that are logged in to the server. This value is specified as a decimal number in the form HHMM.

NOTES

This file is typically created using the ‘‘SHUT’’ command supported by *hfaxd*(8C).

SEE ALSO

hfaxd(8C), *hylafax-server*(5F)

NAME

mkcover – *HylaFAX* continuation cover page generation command

SYNOPSIS

/var/spool/hylafax/bin/mkcover *qfile template result*

DESCRIPTION

bin/mkcover is the command invoked by the facsimile server when a *continuation cover page* is required for an outbound job that is about to be retried. It generates a single POSTSCRIPT® page that is imaged according to the remote capabilities and transmitted as the cover page. The arguments are:

qfile the job description file for the outbound job.

template the cover page template file, as used by the *faxcover*(1) program.

result the filename where the resultant POSTSCRIPT document should be written.

The default *mkcover* command is a shell script that uses *awk*(1) to parse the job description file and generate the cover page document. The *template* file may be optionally compressed with *compress*(1), *pack*(1), or *gzip*(1); the filename suffix is used to decide whether or not the template file contains compressed data.

SEE ALSO

faxq(8C), *hylafax-config*(5F), *sendq*(5F)

NAME

notify − *HylaFAX* user notification script

SYNOPSIS

/var/spool/hylafax/bin/notify *qfile why jobtime commid* [*nextTry*]

DESCRIPTION

bin/notify is the command script invoked by the facsimile server to notify a user about the status of an outbound job. The default notification script sends the user electronic mail describing the event or action. The arguments are:

qfile the pathname of the job description file.

why a string that specifies why the user is being notified; one of:

Why	**Notes**	**Meaning**
done		job was completed successfully
failed	1	job was not completed
rejected		job was rejected for specified reasons
blocked		job is blocked by concurrent jobs
requeued	1	job was not completed, but was requeued for retry
removed		job was deleted from the queue
killed		job was deleted from the queue
timedout		job could not be sent before kill time expired
format_failed		document conversion failed
no_formatter		the document conversion program was not found
poll_rejected	1	a polling request was rejected by the remote side
poll_no_document	1	a polling request failed because nothing was available
poll_failed	1	a polling request failed for an unknown reason

[1] A transcript of the failed communication is returned to the sender.

jobtime the total transmission time of a successful job or the connect time of a failed request (passed as a string).

commid the communication identifier of the *HylaFAX* process that processed the job (used for locating communication transcripts).

nextTry the time (passed as a string) at which the next attempt will be made to complete the job. This parameter is passed only if **why** is ‘‘requeued’’.

FILES

/var/spool/hylafax spooling area
/usr/sbin/sendmail for delivering mail

SEE ALSO

faxq(8C), *hylafax-server*(5F), *sendq*(5F)

NAME

pagermap − SNPP pager identifier mapping file

DESCRIPTION

The file **etc/pagermap** (by default) is used by the SNPP support in *hfaxd*(8C) to map pager identifiers specified with the PAGE directive to IXO/TAP or SMS service providers and optionally a Pager Identification Number (PIN). (The ability to specify a PIN makes it possible to define *aliases* for pagers.) Note that **etc/pagermap** may be called a different name; the filename is specified with the *PagerIDMapFile* configuration parameter read by *hfaxd*(8C).

The pagermap file is comprised of one or more lines of the following format:

<*pattern*> <*dialstring*>[/<*pin*>]

Each line specifies a <*pattern*> that is matched against a client-specified pager identifier string. A pattern may be a plain string or a POSIX 1003.2 extended regular expression; plain strings must compare exactly while regular expressions are unanchored and any substring match constitutes a ‘‘hit’’. Regular expressions are distinguished from plain strings by the presence of one or more of the magic characters ‘‘[]*.^$-+{}()’’. The <*dialstring*> identifies the IXO/TAP or SMS service provider to contact for delivery or, if <*dialstring*> is the string ‘‘reject’’ (case insensitive) then the pager identifier is rejected. If the optional <*pin*> is specified then it used to deliver the page instead of the client-specified identifier.

If the matching expression is a regular expression the optional <*pin*> may reference the entire string matched by the regular expression with the ‘‘&’’ character. Substrings matched with the ‘‘(...)’’ constructs may be referenced by using ‘‘*n*’’ where *n* is a single numeric digit between 1 and 9 that refers to the *n*-th matched substring; c.f. *re_format*(7), *sed*(1), etc.

hfaxd processes lines in the mapping file sequentially; the first line with a pattern that matches the pager identifier is used to deliver or reject the page. Lines that begin with a ‘‘#’’ are treated as comments and discarded. Leading whitespace on a line is also ignored.

EXAMPLES

The following defines an alias for ‘‘sam’’, submits all destinations with a leading ‘‘Sky’’ to a particular IXO/TAP service provider, and otherwise rejects the pager ID. (Note that using alphabetics in dial strings requires a dialstring rules file that understands how to map alphabetics to key codes—or one that passes them through verbatim.)

```
# a simple mapping example
sam        5551212/99
Sky(.*)    1800SkyTel/\1
.*         reject
```

SEE ALSO

hfaxd(8C), *hylafax-server*(5F)

NAME

pagesend − *HylaFAX* IXO/TAP and UCP transmit program

SYNOPSIS

/usr/sbin/pagesend [**−l**] [**−p**] [**−c** *config*] **−m** *device-ID qfile*

DESCRIPTION

pagesend is the program that implements the IXO/TAP and the UCP transmission protocol for the *HylaFAX* system. *pagesend* is invoked by the *faxq* scheduler process for each pager transmit attempt. *pagesend* assumes that its current working directory is the top of the spooling hierarchy and that a suitable modem device has been allocated for its use. In normal use *pagesend* will inherit a lockfile for the modem from the *faxq* process. The format of the job description file specified on the command line is described in *sendq*(5F).

pagesend initializes the modem before attempting to transmit the pager message using the information in the appropriate modem configuration file. If the modem does not respond to this initialization, *pagesend* will repeatedly try to initialize the modem. This behaviour is required for send-only environments in which a *faxgetty*(8C) process is not run (and the modem would be known to be in a properly initialized state). *pagesend* catches SIGINT and SIGTERM and cleans up any resources it controls; this is the mechanism used by *faxq* to abort a job while it is in progress.

Many aspects of *pagesend* operation are controlled through a configuration file. The configuration file is located in the spooler hierarchy in the **etc** subdirectory and is named **config.***device-ID***;** consult *hylafax-config*(5F) for detailed information on the contents of configuration files.

The protocol which must be used for a specific service provider is controled from the *hylafax-info*(5F) file.

OPTIONS

−c *config* Treat *config* as a configuration parameter specification that is interpreted after reading the per-device configuration file. For example, ‘‘-c sessiontracing:0x4f’’ would set the **SessionTracing** configuration parameter to ‘‘0x4f’’, overriding any setting in the configuration file.

−m *devID* The device to use in processing the job. Note that this is a *device identifier* and not the pathname of the tty special file. A device identifier is formed from a device filename by removing any leading ‘‘/dev/’’ and converting any ‘‘/’’ characters to ‘‘_’’ characters.

−l Do the UUCP lockfile protocol in *pagesend*. By default *pagesend* assumes that it is invoked with the device already locked and that it does not need to manage the lockfile.

−p Do not change the process priority when transmitting. Normally *pagesend* will raise its priority to reduce i/o latency. This option is useful for debugging.

NOTES

pagesend returns the new job status to *faxq* through the *exit*(2) return value of the process: *send_retry* (0), retry job; *send_failed* (1), job finished without success, *send_done* (2), job completed successfully, *send_reformat* (3), job should be retried after documents are reformatted.

FILES

/var/spool/hylafax/etc/config.devID device-specific configuration file

Consult *hylafax-server*(5F) for a complete discussion of the structure and content of the spooling area.

SEE ALSO

faxq(8C), *hylafax-server*(5F), *hylafax-config*(5F), *hylafax-info*(5F)

NAME

pagesizes − *HylaFAX* page size definitions

DESCRIPTION

The *pagesizes* database defines the page dimensions and guaranteed reproducible areas (GRA) for well-known page sizes. The GRA is the portion of the page that is guaranteed to be imaged during facsimile transmission. This region is typically less than the full page dimensions because of paper roller contacts and other mechanical aspects of the printing process in a facsimile machine.

All *HylaFAX* programs that require page size information read the information from this database using a page size name. Documents should be prepared such that the full page dimensions are employed with the imaged area contained within the GRA.

The system-wide default page size to use in preparing documents for transmission is given by the ‘‘*default*’’ entry in the database. (NB: the default entry should be placed last so that inverse matches find the real page size name and not the default entry.)

The page size database is an ASCII file with the following format. Each entry consists of whitespace-separated fields:

```
name abbrev width height gra-width gra-height top-margin left-margin
```

Fields have the following interpretation:

name the full name for the page size; e.g. ISO A4;

abbrev an abbreviated version of the full name for use in compact listings such as the receive queue listing printed by *faxstat*(1);

width the full width of the page;

height the full height of the page;

gra-width the width of the GRA;

gra-height the height of the GRA;

top-margin the margin between the top of the full page and the top of the GRA;

left-margin the margin between the left side of the full page the left side of the GRA.

The first two fields must be separated from the subsequent fields by a tab character (possibly followed by more whitespace); this is done to easily permit blank characters to be included in names. Otherwise fields can be separated by any amount of any kind of whitespace. Numbers are all base 10 and in basic measurement units (BMU); defined as 1/1200 x 25.4 millimeters for paper output with a scale factor of one. All fields must be present on a single line; otherwise the entry is ignored. Comments are introduced by the ‘‘#’’ character and continue to the end the line.

Page size database lookups are either by name or by page dimensions. Lookups by name are done by sequentially scanning the entries in the database for the first entry that has a matching abbreviation or substring of the page size name field. The string comparisons ignore case so, for example, ‘‘a4’’ would match a full name of ‘‘ISO A4’’. Lookups by dimension scan the entire database and return the page with the closest dimensions using a straightforward distance metric. If the difference in dimensions of the closest match is greater than 1/2 inch on each side, then no page entry is returned for a lookup by dimension.

SEE ALSO

hylafax-client(1), *faxmail*(1), *faxstat*(1), *sendfax*(1), *sgi2fax*(1), *textfmt*(1).

NAME

pdf2fax − image Portable Document Format file for facsimile transmission by *HylaFAX*

SYNOPSIS

/var/spool/hylafax/bin/pdf2fax [*options*] [*files*...]

DESCRIPTION

pdf2fax takes Portable Document Format (PDF) input and generates a TIFF Class F image file suitable for facsimile transmission. It is usually invoked by the *HylaFAX* scheduler process, *faxq*(8C). PDF imaging is typically done using the freely available *Ghostscript* software.

pdf2fax is actually a link (symbolic or otherwise) to the script appropriate for the PDF imaging tool. The link is setup when the *faxsetup*(8C) script is run to configure a *HylaFAX* server machine for use.

Input files containing PDF may be specified on the command line. If none are given, then *pdf2fax* will read from standard input.

By default, *pdf2fax* will generate a 98 line/inch output document whose page width is 1728 pixels and page length is 297 mm (a standard A4 page at low resolution). Horizontal resolution is always 204 dpi. Command line options may be used to select alternate vertical resolution and page dimensions.

If no filename is given, *pdf2fax* writes its output to a file named **pdf.fax**; the **−o** option can be used to specify an alternate filename.

By default, output is compressed using the standard Group 3 1D-encoding scheme, with EOL codes padded to byte boundaries. If the **−2** option is specified, output is encoded with the optional Group 3 2D-encoding scheme (EOL codes are still padded to byte boundaries).

Note that **bin/pdf2fax** is usually just a link to a shell script that invokes the appropriate PDF RIP to do the work.

OPTIONS

The following operations are recognized:

−1 Write Group 3 1D-encoded data to the output file.

−2 Write Group 3 2D-encoded data to the output file.

−m *pages* Set the maximum number of pages that are permitted in the output file. If more pages are present in the input document then abort processing.

−o *file* Write output to the specified *file* instead of the default **pdf.fax** filename.

−w *width* Set the output page *width*, in pixels. By default, pages are 1728 pixels wide.

−l *length* Set the output page *length* in millimeters. By default, pages are 297 millimeters long.

−r *vres* Set the vertical resolution, in lines/inch, at which to image the PDF.

RUNTIME ENVIRONMENT

Before user-supplied PDF is passed to the PDF interpreter, a limited number of operators are defined. These operators are operators commonly found in a LaserWriter printer and (sometimes) generated by Macintosh-based applications.

The default Display PDF-based execution environment includes support for the following fonts:

AvantGarde-Book	Helvetica-Bold	Palatino-BoldItalic
AvantGarde-BookOblique	Helvetica-BoldOblique	Palatino-Italic
AvantGarde-Demi	Helvetica-Narrow	Palatino-Roman
AvantGarde-DemiOblique	Helvetica-Narrow-Bold	StoneSerif
Bookman-Demi	Helvetica-Narrow-BoldOblique	Symbol
Bookman-DemiItalic	Helvetica-Narrow-Oblique	Tekton
Bookman-Light	Helvetica-Oblique	Times-Bold
Bookman-LightItalic	Lithos-Bold	Times-BoldItalic
Courier	NewCenturySchlbk-Bold	Times-Italic
Courier-Bold	NewCenturySchlbk-BoldItalic	Times-Roman

Courier-BoldOblique	NewCenturySchlbk-Italic	ZapfChancery-MediumItalic
Courier-Oblique	NewCenturySchlbk-Roman	ZapfDingbats
Helvetica	Palatino-Bold	

Note that this is the same set of fonts supported by the DPS extension to the X server.

The default Ghostscript-based environment is more uncertain; consult the Ghostscript documentation for information on what fonts are available.

BUGS

Output from the Display PDF version of this command is encoded according to the Display PDF binary encoding protocol.

Ghostscript-based versions of this program fail to exit with non-zero status when an invalid PDF document is processed; this makes it problematic for the *HylaFAX* queue manager to determine when a problem arises when preparing a job for transmission.

SEE ALSO

sendfax(1), *faxq*(8C), *hylafax-server(5F)*

NAME

pollrcvd − *HylaFAX* script for delivering facsimile received by polling

SYNOPSIS

/var/spool/hylafax/bin/pollrcvd *mailaddr faxfile devid commid error-msg*

DESCRIPTION

bin/pollrcvd is the command script invoked by the *HylaFAX* server whenever a facsimile is received as a result of a polling operation. The default script uses electronic mail to deliver the facsimile to the user that submitted the poll request. The arguments are:

mailaddr the electronic mail address of the user that submitted the poll request.

faxfile the pathname of the received TIFF file relative to the root of the spooling hierarchy.

commid The communication identifier associated with the call (used to include a transcript when an error occurs).

devid the modem device on which the document was received.

error-msg an error message that is non-null if an error was encountered during the receive operation.

NOTES

The default script delivers facsimile as PostScript wrapped in MIME.

FILES

/var/spool/hylafax	spooling area
/usr/sbin/faxinfo	for printing information about the facsimile
/usr/bin/fax2ps	for converting TIFF to PostScript
/usr/sbin/sendmail	for delivering mail

SEE ALSO

faxq(8C), *hylafax-server*(5F), *recvq*(5F)

NAME

ps2fax − image POSTSCRIPT® for facsimile transmission by *HylaFAX*

SYNOPSIS

/var/spool/hylafax/bin/ps2fax [*options*] [*files...*]

DESCRIPTION

ps2fax takes POSTSCRIPT input and generates a TIFF Class F image file suitable for facsimile transmission. It is usually invoked by the *HylaFAX* scheduler process, *faxq*(8C). On Silicon Graphics' IRIX systems, this application is based on Adobe's Display POSTSCRIPT software. On other machines POSTSCRIPT imaging is typically done using the freely available *Ghostscript* software.

ps2fax is actually a link (symbolic or otherwise) to the script appropriate for the POSTSCRIPT imaging tool. The link is setup when the *faxsetup*(8C) script is run to configure a *HylaFAX* server machine for use.

Input files containing POSTSCRIPT may be specified on the command line. If none are given, then *ps2fax* will read from standard input.

By default, *ps2fax* will generate a 98 line/inch output document whose page width is 1728 pixels and page length is 297 mm (a standard A4 page at low resolution). Horizontal resolution is always 204 dpi. Command line options may be used to select alternate vertical resolution and page dimensions.

If no filename is given, *ps2fax* writes its output to a file named **ps.fax**; the **−o** option can be used to specify an alternate filename.

By default, output is compressed using the standard Group 3 1D-encoding scheme, with EOL codes padded to byte boundaries. If the **−2** option is specified, output is encoded with the optional Group 3 2D-encoding scheme (EOL codes are still padded to byte boundaries).

Note that **bin/ps2fax** is usually just a link to a shell script that invokes the appropriate POSTSCRIPT RIP to do the work.

OPTIONS

The following operations are recognized:

−1 Write Group 3 1D-encoded data to the output file.

−2 Write Group 3 2D-encoded data to the output file.

−m *pages* Set the maximum number of pages that are permitted in the output file. If more pages are present in the input document then abort processing.

−o *file* Write output to the specified *file* instead of the default **ps.fax** filename.

−w *width* Set the output page *width*, in pixels. By default, pages are 1728 pixels wide.

−l *length* Set the output page *length* in millimeters. By default, pages are 297 millimeters long.

−r *vres* Set the vertical resolution, in lines/inch, at which to image the POSTSCRIPT.

RUNTIME ENVIRONMENT

Before user-supplied POSTSCRIPT is passed to the POSTSCRIPT interpreter, a limited number of operators are defined. These operators are operators commonly found in a LaserWriter printer and (sometimes) generated by Macintosh-based applications.

The default Display POSTSCRIPT-based execution environment includes support for the following fonts:

AvantGarde-Book	Helvetica-Bold	Palatino-BoldItalic
AvantGarde-BookOblique	Helvetica-BoldOblique	Palatino-Italic
AvantGarde-Demi	Helvetica-Narrow	Palatino-Roman
AvantGarde-DemiOblique	Helvetica-Narrow-Bold	StoneSerif
Bookman-Demi	Helvetica-Narrow-BoldOblique	Symbol
Bookman-DemiItalic	Helvetica-Narrow-Oblique	Tekton
Bookman-Light	Helvetica-Oblique	Times-Bold
Bookman-LightItalic	Lithos-Bold	Times-BoldItalic
Courier	NewCenturySchlbk-Bold	Times-Italic

Courier-Bold	NewCenturySchlbk-BoldItalic	Times-Roman
Courier-BoldOblique	NewCenturySchlbk-Italic	ZapfChancery-MediumItalic
Courier-Oblique	NewCenturySchlbk-Roman	ZapfDingbats
Helvetica	Palatino-Bold	

Note that this is the same set of fonts supported by the DPS extension to the X server.

The default Ghostscript-based environment is more uncertain; consult the Ghostscript documentation for information on what fonts are available.

FILES

/var/spool/hylafax/etc/dpsprinter.ps	POSTSCRIPT code read at startup for DPS
/usr/lib/DPS/default.dps_vm	startup VM file for DPS
/usr/lib/DPS/outline	Type 1 fonts for DPS
/usr/lib/DPS/prebuilt	prebuilt versions of fonts for DPS

BUGS

Output from the Display POSTSCRIPT version of this command is encoded according to the Display POSTSCRIPT binary encoding protocol.

No prebuilt bitmap fonts are currently available for the DPS-based imager; this would greatly improve legibility at low resolutions, as the rendering of outline fonts at 98 lpi is particularly poor.

Both DPS- and Ghostscript-based versions of this program fail to exit with non-zero status when an invalid POSTSCRIPT document is processed; this makes it problematic for the *HylaFAX* queue manager to determine when a problem arises when preparing a job for transmission.

SEE ALSO

sendfax(1), *faxq*(8C), *hylafax-server(5F)*

NAME

recvq/fax* – received facsimile document format

DESCRIPTION

Facsimile received by *HylaFAX* are placed in the **recvq** subdirectory. Each facsimile document is stored as a single Class F Tagged Image File Format (TIFF) file. This format includes the following information:

- page width and length
- horizontal and vertical resolution
- page orientation
- phone number of the sender
- date and time document was received
- number of scanlines received with errors
- maximum number of consecutive scanlines received with errors
- whether or not bad scanlines were regenerated by the receiver
- make and model of the modem used to receive the document
- name of the host computer on which the document was received

Image data is written using either CCITT T.4 or CCITT T.6 compression. By default data is written using the compression scheme/data format negotiated by the sender and receiver. However, if *faxgetty*(8C) is configured to do copy quality checking then data is written using the compression scheme defined by the **RecvDataFormat** configuration parameter; see *hylafax-config*(5F).

A scanline that is regenerated by the receiver because of an error is typically just a copy of the previous good scanline.

The files in the **recvq** directory are created with the protection mode specified in the server configuration file(s). By default, files are created with mode 0600 and owned by the *fax* user, which means that they are accessible only to the fax user.

To print a facsimile on a POSTSCRIPT® printer, the *fax2ps*(1) program can be used.

SEE ALSO

viewfax(1), *faxview*(1), *fax2ps*(1), *faxgetty*(8C), *libtiff*(3), *hylafax-config*(5F)

NAME

recvstats − summarize *HylaFAX* receive accounting information

SYNOPSIS

/usr/sbin/recvstats [*options*] [*files*]

DESCRIPTION

recvstats is a command script that prints a summary of the *HylaFAX* accounting statistics for received facsimile. The statistics can be sorted using one of several keys: the caller's CSI (default), the communication speed, and the communication data format. Statistics are read from the file **/var/spool/hylafax/etc/xferfaxlog** unless filenames are specified on the command line.

The following table is an example of the output from *recvstats* (using the default sort key):

Sender	**Pages**	**Time**	**Pg/min**	**Errs**	**TypRate**	**TypData**
	61	30:47	2.0	16	9600	1-DMR
+13037904674	5	1:34	3.2	2	9600	1-DMR
+14159657824	1	1:00	1.0	0	9600	1-DMR
+15086636852	2	0:00	2.0	0	9600	1-DMR
+15105268781	740	5:47:51	2.1	59	9600	1-DMR
+15105287164	20	7:00	2.9	0	9600	1-DMR
+15123713545	3	1:19	2.3	0	9600	1-DMR
+15123713641	5	1:04	4.6	0	9600	1-DMR
+16099211926	3	2:00	1.5	0	9600	1-DMR
+17088987617	1	0:18	3.3	0	9600	1-DMR
415 390 6175	22	6:01	3.6	0	9600	1-DMR
415 965 7651	56	34:52	1.6	14	9600	1-DMR
415 973 6723	1	2:09	0.5	0	9600	2-DMR
49 211 161514	1	0:00	1.0	0	9600	2-DMR
8185970503	1	0:28	2.1	0	9600	2-DMR
ALDUS CORPORATION	3	2:34	1.2	0	9600	1-DMR

Total	925	7:19:03	2.1	91		

The *Sender* column shows the caller's TSI (notice that some callers have not setup a TSI). The *Pages* column is the total number of good pages received. The *time* column shows the total amount of time spent receiving facsimile from a sender. The *Pg/min* column displays the average transfer rate. The *Errs* column indicates how many protocol errors were encountered while receiving facsimile. The *TypRate* column displays the most common communication speed with which facsimile were received. The *TypData* column shows the most common data format in which facsimile data were received.

OPTIONS

−csi* Sort output by caller's CSI.

−format Sort output by data format.

−send* Sort output by sender (i.e. caller's CSI).

−speed Sort output by signalling rate.

−since *time* Show information only for the facsimile received since the specified *time*; where *time* is of the form "MM/DD/YY HH:MM" (the *date*(1) format string %D %H:%M).

FILES

/var/spool/hylafax spooling area
/var/spool/hylafax/etc/xferfaxlog default file from which to read statistics

SEE ALSO

faxq(8C), *xferfaxstats*(8C), *hylafax-log*(5F)

NAME

sendfax − submit a facsimile job for transmission

SYNOPSIS

sendfax [*options*] [*files...*]

DESCRIPTION

sendfax submits one or more facsimile transmission requests to a *HylaFAX* facsimile server. Requests are normally processed immediately, although they may also be queued for transmission at a later time using a syntax identical to the *at*(1) command. For each job that is queued, *sendfax* prints a *job identifier* and a *job group identifier* on the standard output. A job identifier is the token by which jobs are identified within *HylaFAX*; job group identifiers are used to reference a set of jobs that have been submitted at the same time. Job and job group identifiers can be supplied to the *faxrm*(1) command to remove the jobs or to the *faxalter*(1) command to alter job parameters.

Facsimile are made from the concatenation of a cover page (see below) and one or more files specified on the command line. If no files are supplied on the command line, *sendfax* will read data from the standard input unless polling is requested. Cover pages are distinguished for the purpose of generating ‘‘*continuation cover pages*’’ on the server; these are makeup cover pages that are automatically generated if a job fails after sending the original user-specified cover page.

sendfax passes POSTSCRIPT® and TIFF documents directly through to the fax server for transmission, and attempts to convert other file formats to either POSTSCRIPT or TIFF. In normal operation *sendfax* automatically converts ASCII-text, *troff*(1) output, Portable Document Format (PDF), and Silicon Graphics format images for transmission. Support for other file formats can be easily added through an extensible file typing and conversion facility; see *typerules*(5F). Note that since TIFF Class F documents are processed directly it is simple to forward a received facsimile.

By default *sendfax* will generate a cover page for each facsimile that is to be transmitted. This cover page is created by the ancillary *faxcover*(1) program using information deduced by *sendfax* and by information supplied on the command line. Automatic cover page generation may be suppressed with the **−n** option.

In addition to sending documents, *sendfax* can be used to poll a remote facsimile machine to retrieve waiting documents. If the **−p** option is specified, a polling request will be submitted to each destination. If a poll request is made in conjunction with the transmission of documents, the poll request will take place after the documents are transmitted. Documents retrieved by polling are returned to the requestor by electronic mail and may be viewed with a suitable viewer program.

One or more destinations must be specified on the command line using the **−d** option. A destination is the dialing string to use to contact the remote fax machine. Phone numbers may include non-alphanumeric characters for readability—only alphanumerics and ‘‘+’’ (for denoting a country code) are significant in placing a call. Alphabetic characters are automatically mapped to their numeric key equivalents: ‘‘abcABC’’ to ‘‘2’’, ‘‘defDEF’’ to ‘‘3’’, etc. Dialing prefixes, such as ‘‘1’’ and ‘‘011’’ (for long distance calling in the United States) need not be included if an unambigous number is specified. Examples of unambiguous phone numbers are:

+31.77.594.131	*to call the Netherlands*
+14159657824	*to call California*
9657824	*to make a local call*
1-215-356-1666	*to call Pennsylvania from within the US*

Non-numeric strings, such as ‘‘*70’’ (to defeat call waiting), may be included in the destination phone number. Whether or not they are stripped from the string used to dial the destination fax machine is dependent on the dialstring rules configured for use on the server; c.f. *config*(5F) and *dialrules*(5F).

A destination may also be specified as *user@fax-number*. In this case the *user* string will be passed to *faxcover* as the identity of the recipient. Note that multi-word names must be enclosed in quote marks; for example, ‘‘Sam Leffler@+1.415.555.1212’’.

The order of options on the command line is significant. Each fax to be transmitted is assigned the current value of parameters specified on the command line; e.g. whether or not to generate a cover page. This means that options that specify parameters for a particular destination must precede the dialstring specified with the **−d** option. For example,

sendfax -n -d 5551212 /etc/passwd

submits a fax to 5551212 that has no cover page while

sendfax -d 5551212 -n /etc/passwd

submits the same fax but with the default cover page handling. Note also that parameter values persist across **−d** options so the following

sendfax -n -d 5551212 -d 5551313 /etc/network

submits two facsimile, both of which have no cover page.

OPTIONS

−1 Transmit facsimile documents using 1-D MR compression. By default *HylaFAX* will use the optimal compression scheme supported by the transmitting modem and receiving facsimile device.

−2 Transmit facsimile documents using 2-D MR compression, if possible. By default *HylaFAX* will use the optimal compression scheme supported by the transmitting modem and receiving facsimile device.

−a *time* Schedule transmission up to one year in advance at the indicated *time*. *Time* is specified in the format: ‘‘[HH:MM [AM|PM]|now] [+ N *period* | *month* DD | *dayofweek*]´´ where *month* can be Jan(uary), Feb(ruary), Mar(ch), Apr(il), May, Jun(e), Jul(y), Aug(ust), Sep(tember), Oct(ober), Nov(ember), or Dec(ember); *period* can be minute(s), hour(s), day(s), week(s), or month(s); *dayofweek* can be Mon(day), Tue(sday), Wed(nesday), Thu(rsday), Fri(day), Sat(urday), Sun(day); e.g. ‘‘now + 30 minutes´´ to schedule transmission 30 minutes in the future. Trailing information in the *time* field (such as YYYY) is ignored. By default, *sendfax* schedules facsimile for immediate delivery. Time should be given in the user’s local time.

−b *speed* Transmit facsimile documents using a signalling rate that is constrained to be at least *speed* bits/second. If the specified speed is greater than the transmitting modem or receiver is capable of, it is reduced to the highest possible speed both devices support. If a negotiated signalling rate greater than or equal to *speed* cannot be negotiated then the job is aborted. Possible speeds are: 2400, 4800, 7200, 9600, 12000, and 14400. By default *HylaFAX* will use the fastest possible rate supported by the transmitting modem and receiving facsimile device.

−B *speed* Try to transmit facsimile documents using the desired signalling rate. Possible speeds are: 2400, 4800, 7200, 9600, 12000, and 14400. If the specified speed is greater than the transmitting modem or receiver is capable of, it is reduced to the highest possible speed both devices support. By default *HylaFAX* will use the fastest possible rate supported by the transmitting modem and receiving facsimile device (unless restricted by other server configuration); this option can be used to reduce the speed. Note that international calls are frequently more reliable when transmissions are restricted to 9600 bits/second.

−c *comments* Pass *comments* to the *faxcover*(1) program for inclusion in the comments field of any generated cover page.

−C *file* Pass *file* to the *faxcover*(1) program for use as the cover page template.

−D Enable notification by electronic mail when the facsimile has been delivered. By default *HylaFAX* will notify the submitter only if there is a problem with a job.

−d *destination*
Create a job to transmit the supplied documents to a facsimile machine at the indicated *destination*. If the *destination* is of the form ‘‘*user@phone-number*’’ then any cover page will show *user* as the intended receipient.

−E Disable use of the Error Correction Mode (ECM) when transmitting page data. By default *HylaFAX* will attempt to use ECM when both the trasmitting modem and receiving device support it. Note that international calls are sometimes faster when ECM is disabled. Beware also that some modems and/or facsimile devices do not properly implement ECM causing transmission failures.

−f *from* Use *from* as the identity of the facsimile sender. Otherwise the sender is the user that invoked *sendfax*. The sender is an account name to which the *HylaFAX* software should direct notification messages. The *from* identity may be a fullblown ''From:'' line, e.g. e.g. ''Joe Blow <joe@foo.com>'' and ''joe@foo.com (Joe Blow)'' are acceptable. If no host-part is specified in the *from* string the local hostname is used. This option is useful when *sendfax* is invoked on behalf of another user, such as in the implementation of an electronic mail to fax gateway.

−F *format* Use the specified *format* when imaging tag lines on outgoing pages. Consult the description of the **TagLineFormat** parameter in *config*(5F) for an explanation of how to construct a format string.

Note that many locales require that proper identification appear on all transmitted facsimile; when using this facility one should always include proper identification (e.g. a phone number) in the imaged taglines.

−h [*modem*@]*host*[*:port*]
Force the jobs to be processed on a specific *host* and, optionally, using a particular *modem*. The *host* may be either a symbolic name or a network address. If no **−h** option is supplied, *sendfax* uses the FAXSERVER environment variable to identify the *HylaFAX* server to which the jobs should be directed. If no server is specified then *sendfax* attempts to contact a server on the local host. If no modem is specified, the job will be submitted to any available modem.

−i *identifier* Use the *identifier* string in any notification messages instead of the usual job identifier number. This option is useful for applications that cross reference notification messages against job submissions.

−I *time* If a job must be requeued because of a communication failure schedule the retry according to the specified *time*. Times are given in seconds. Minutes, hours, and days can be specified with ''*min*'', ''*hour*'', and ''*day*'' suffixes, respectively. By default *HylaFAX* reschedules jobs using retry times that depend on the manner in which the job failed; this interface permits a user to override this algorithm and specify a single fixed retry time for all communication failures.

−k *time* Kill the job if it does not complete in the indicated *time*. As for the **−a** option, the *time* is specifed using notation compatible with *at*(1). Note that the *time* is relative to the time of the initial transmission attempt. The default kill time is ''now + 3 hours'' (three hours from the time the job is first processed). To retry for an hour, ''now +1 hour'' could be used.

−l Transmit the facsimile at a vertical resolution of 98 lines/inch; otherwise known as *low resolution*.

−m Transmit the facsimile at a vertical resolution of 196 lines/inch; otherwise known as *medium resolution* or *fine mode*.

−M *mst* Transmit facsimile documents using the specified minimum scanline time. Possible times are: 0ms, 5ms, 10ms2, 10ms, 20ms2, 20ms, 40ms2, and 40ms. If the specified time is less than the transmitting modem or receiver is capable of, it is increased to the lowest possible time both devices support. By default *HylaFAX* will use the lowest possible time supported by the transmitting modem and receiving facsimile device (unless restricted by other server configuration). This option is mostly useful for debugging.

−n Suppress the automatic generation of a cover page for subsequent facsimile. Note that polling requests do not receive a cover page unless they are combined with the transmission of one or more files.

−N Do not notify the sender by electronic mail when the facsimile has been delivered or requeued.

−p Poll each destination for any available documents. If a poll request is made together with document transmission, the poll operation is done after the documents are transmitted. Documents received by polling are returned to the sender by electronic mail.

−P *priority* Assign the specified scheduling *priority* to subsequent jobs. A priority is an integer value in the range [0-255]. Lower values result in higher priority processing. By default, each job is assigned 127 for an initial scheduling priority. A job's priority is raised (numerically lowered) for each failed attempt to transmit so that retransmit attempts are done ahead of newly submitted jobs. The *priority* may also be specified as one of the following symbolic names: *default* or *normal* (127), *bulk* or *junk* (207), *high* (63).

−R Enable notification by electronic mail when the facsimile has been delivered and when it is requeued for retransmission. By default *HylaFAX* will notify the submitter only if there is a problem with a job.

−r *regarding* Pass *regarding* to the *faxcover*(1) program as the value of the ‘‘Re:’’ field of any generated cover page.

−s *size* Set the page size to use for the transmitted facsimile. Facsimile are normally imaged with a system-default page size (usually letter-size pages, 8.5" by 11", for sites in North America). Alternate page sizes are specified symbolically using either the name or abbreviation of an entry in the *pagesizes*(5F) database; e.g. *a3* (ISO A3), *a4* (ISO A4), *a5* (ISO A5), *a6* (ISO A6), *b4* (ISO B4), *na-let* (North American Letter), *us-leg* (American Legal), *us-led* (American Ledger), *us-exe* (American Executive), *jp-let* (Japanese Letter), and *jp-leg* (Japanese Legal). Comparisons are case-insensitive and any match of a substring of the full page-size name is sufficient; e.g. ‘‘legal’’ would match ‘‘American Legal’’.

Note that it may not be permissible to image into the full page area; the guaranteed reproducible area for a page is typically inset. Also, note that while arbitrary page sizes can be specified through the page size database, only a limited set of page dimensions are supported by the Group 3 facsimile protocol. Thus if an odd-size facsimile is submitted for transmission it may not be possible to determine if it can be sent until the fax server establishes communication with the remote facsimile machine.

−t *tries* Make no more than *tries* attempts to send the facsimile. By default, *HylaFAX* will terminate a job if: 3 consecutive attempts to send a particular page fail, or it appears the receiver is not a facsimile machine. Otherwise *HylaFAX* places no limit on the *number* of attempts to send a facsimile, instead terminating a job if it is not completed within a *kill time* timeout interval. Note that a try is a call in which carrier is established and the facsimile protocol is commenced; this is contrasted with a call attempt that might have failed because the line was busy.

−T *maxdials* Make no more than *maxdials* calls in attempting to send the facsimile. By default, HylaFAX will redial 12 times before killing a job.

−V *voice-no* Pass *voice-no* to the *faxcover*(1) program as the destination person's voice number.

−w Wait for the job to complete before ending the sendfax session. If this option is used, the sendfax process will not terminate until the job is completed.

−x *company* Pass *company* to the *faxcover*(1) program as the destination company name.

−y *location* Pass *location* to the *faxcover*(1) program as the destination company's location.

−v Print information on the standard output about each conversion and cover sheet generation operation it does. If **−v** is specified twice, the protocol between *sendfax* and the *hfaxd* server process that does the submission work on the remote machine is also displayed.

CONFIGURATION PARAMETERS

sendfax reads configuration information from the files **/usr/share/fax/hfaxd.conf**, **/usr/share/fax/hyla.conf**, **/usr/share/fax/sendfax.conf**, and **~/.hylarc**; in that order, with later-read settings superceding any that may have been previously read. Configuration files follow the conventions described in *hylafax-client*(1). The following configuration parameters are recognized:

Tag	**Type**	**Default**	**Description**
AutoCoverPage	boolean	Yes	automatically generate cover page
ChopThreshold	float	3.0	page chopping threshold
CoverCmd	string	*see below*	pathname of cover sheet program
Cover-Comments	string	–	cover page comments string
Cover-Company	string	–	cover page company name string
Cover-Location	string	–	cover page company location string
Cover-Regarding	string	–	cover page regarding string
Cover-Template	string	–	cover page template filename
Cover-Voice	string	–	cover page voice number string
DateFormat	string	–	cover page date format string
DesiredEC	boolean	–	desired use of ECM
DesiredMST	string	–	desired minimum scanling time to use
DesiredSpeed	integer	–	desired signalling rate to use
DialRules	string	*see below*	file containing dialstring rules
From	string	–	sender's identity
HRes	float	204.	horizontal resolution
Host	string	localhost	host to contact for service
KillTime	string	–	time to expire job
MailAddr	string	–	mail address for notification messages
MaxDials	integer	12	times to retry dialing
MaxTries	integer	3	times to retry transmission
MinSpeed	integer	–	minimum acceptable signalling rate
Modem	string	–	modem to use on server
Notify	string	none	control email notification
PageChop	string	default	control page chop handling
PageLength	float	–	page length in millimeters
PageSize	string	default	page size by name
PageWidth	float	–	page width in millimeters
Port	integer	4559	port to use in contacting server
Priority	string	default	job scheduling priority
Protocol	string	tcp	protocol to use in contacting server
RetryTime	string	–	delay between failed attempts to send
SendTime	string	–	time to send job
TagLine	string	–	tagline format string
TypeRules	string	*see below*	file containing file typing rules
VRes	float	98.	vertical resolution
Verbose	boolean	No	whether or not to enable protocol tracing

The configuration parameters are explained below. Certain configuration parameters, notably those for setting strings on automatically generated cover pages, are defined mainly for use by other *HylaFAX* programs that share common code used by *sendfax*.

AutoCoverPage Control whether or not a cover page is automatically generated for each job. (Equivalent to the **–n** option.)

ChopThreshold The amount of white space, in inches, that must be present at the bottom of a page before *HylaFAX* will attempt to truncate the page transmission.

CoverCmd The absolute pathname of the program to use to generate cover pages. The default cover sheet program is **/usr/bin/faxcover**.

Cover-Comments The comments string to pass to the cover sheet program when auto-generating cover pages.

Cover-Company The company name string to pass to the cover sheet program when auto-generating cover pages.

Cover-Location The company location string to pass to the cover sheet program when auto-generating cover pages.

Cover-Regarding The regarding string to pass to the cover sheet program when auto-generating cover pages.

Cover-Template The absolute pathname of a cover sheet template file to use in auto-generating cover pages.

Cover-Voice The voice number string to pass to the cover sheet program when auto-generating cover pages.

DateFormat The *strftime*(3) format string to pass to the cover page command when auto-generating cover pages.

DesiredEC Whether or not to request the use of the optional Error Correct Mode (ECM) when transmitting facsimile. (Equivalent to the **−E** option.)

DesiredMST The desired minimum scanline time to use when transmitting facsimile. (Equivalent to the **−M** option.)

DesiredSpeed The desired signalling rate to the use when transmitting facsimile. (Equivalent to the **−B** option.)

DialRules The name of the file containing the dialstring processing rules to when preparing an externally visible representation of the dialstring (e.g. to strip credit card numbers). By default this is /usr/share/fax/dialrules.

From The sender's identity. (Equivalent to the **−f** option.)

Host The host to contact for service. (Equivalent to the **−h** option.)

HRes The horizontal resolution to use when transmitting facsimile. (NB: *this currently has no effect*.)

KillTime The expiration time to assign to each job. (Equivalent to the **−k** option.)

MailAddr The electronic mail address to direct notification messages from the server. If this string is specified without an ''@hostname'' part then the local hostname will automatically be appended. (Equivalent to the **−f** option.)

MaxDials The maximum number of times to dial the phone for each job. (Equivalent to the **−T** option.)

MaxTries The maximum number of times to retry sending a job. (Equivalent to the **−t** option.)

MinSpeed The minimum acceptable signalling rate to use when transmitting facsimile. (Equivalent to the **−b** option.)

Modem The modem to use on the server. (Equivalent to the **−h** option.)

Notify Control the email notification messages from the server. This string may be one of ''done'', ''none'', ''requeued'' or ''default'' with an optionally preceding ''when '' (e.g. ''when done''). Note that ''when requeued'' implies ''when done''. (Equivalent to the **−D**, **−R**, and **−N** options.)

PageChop Control page chop handling when transmitting facsimile. This string may be one of ''none'', ''all'', or ''last' to have no pages chopped, all pages chopped, or only the last page chopped; respectively. Page chopping is controlled by the **ChopThreshold** parameter.

PageLength	Set the transmitted page length in millimeters.
PageSize	Set the page dimensions according to an entry in the *pagesizes*(5F) database. (Equivalent to the **−s** option.)
PageWidth	Set the transmitted page width in millimeters.
Port	The network port to contact for service. (Eqvuialent to the **−h** option.)
Priority	The scheduling priority to assign to each job. (Equivalent to the **−P** option.)
Protocol	The name of the communication protocol to use when contacting a server. (Equivalent to the **FAXSERVICE** environment variable.)
RetryTime	The time to delay between job retries due to a communication failure. (Equivalent to the **−I** option.)
SendTime	The time to send jobs. (Equivalent to the **−a** option.)
TagLine	The tagline format string to use when transmitting facsimile. (Equivalent to the **−F** option.)
TypeRules	The name of the file containing file type conversion rules. By default this is /usr/share/fax/typerules.
Verbose	Control protocol tracing. (Equivalent to the **−vv** option.)
VRes	Set the vertical resolution in lines/inch to use when transmitting facsimile. (Equivalent to the **−m** and **−l** options.)

DIAGNOSTICS

The following messages are generated because of problems encountered on the local machine.

Could not call server. The facsimile server on the remote machine did not respond.

No input data; tranmission aborted. No data was supplied (typically on the standard input). No facsimile will be sent.

Error converting data; command was %s." A data conversion operation failed; the shell command that was used is printed.

Can not convert %s. Data needed to be converted for transmission, but the required conversion rule was not located. This may indicate that the *HylaFAX* software was incorrectly installed; or perhaps that an NFS-mounted directory is temporarily unavailable.

%s: Can not determine file type. *sendfax* was unable to deduce the type of the file.

%s: Not a regular file. The file is something other than a regular file; for example, a directory.

The following messages may be printed by *sendfax*. These messages describe problems encountered by the server process on the remote machine. This list is incomplete; other messages may be generated by the server under unusual circumstances.

FILES

~/.hylarc	per-user configuration file
/usr/share/fax/hyla.conf	system-wide configuration file
/usr/share/fax/sendfax.conf	system-wide configuration file
/usr/share/fax/typerules	file type and conversion rules
/usr/share/fax/pagesizes	page size database
/usr/share/fax/dialrules	optional client dialstring rules
/usr/bin/faxcover	for generating cover sheets
/usr/sbin/textfmt	for converting text to PostScript
/usr/sbin/sgi2fax	for converting SGI RGB images
/usr/tmp/sndfaxXXXXXX	temporary files

SEE ALSO

at(1), *hylafax-client*(1), *faxalter*(1), *faxcover*(1), *faxmail*(1), *faxrm*(1), *faxstat*(1), *sgi2fax*(1), *textfmt*(1), *hfaxd*(8C), *hylafax-server*(5F), *typerules*(5F)

NAME

sendpage − submit a pager job for transmission

SYNOPSIS

sendpage [*options*] [*message...*]

DESCRIPTION

sendpage submits a pager transmission request to a server using the Simple Network Paging Protocol (SNPP) described in RFC 1861. Pager requests specified with *sendpage* are normally processed immediately, although they may also be queued for transmission at a later time using a syntax identical to the *at*(1) command. For each job that is queued, *sendpage* prints a *job identifier* on the standard output. A job identifier is the token by which jobs are identified within *HylaFAX*. Job identifiers can be supplied to the *faxrm*(1) command to remove the jobs or to the *faxalter*(1) command to alter job parameters.

Pages may be sent to one or more destination paging terminals and, optionally, include a text message (depending on the capability of the receiving device). Each destination is identified by a Pager Identification Number (PIN) specified with the **−p** option; for example, sendpage -p 12345 ‘‘Help, the fax machine is on fire!’’. A PIN is typically a numeric string though it may also be an alpha-numeric alias depending on the capabilities of the SNPP server.

Text messages can be specified on the command line or taken from standard input. The **−n** option must be used if no text message is to be supplied with the page request—as for a numeric-only pager. Text supplied to the paging system is sent exactly as submitted including any newline, carriage return, or non-ASCII data. Message preparation including filtering and formatting must be done prior to submitting the data to *sendpage*. Note that text supplied on the command line does not include a trailing newline character.

The order of options on the command line is significant. Each page to be transmitted is assigned the current value of parameters specified on the command line; e.g. whether or not to queue the request. This means that options that specify parameters for a particular destination must precede the pager ID specified with the **−p** option. For example,

    sendpage -l 2 -p sam Your network is hosed

submits a page to ‘‘sam’’ that has service level 2 while

    sendpage -p sam -l 2 Your network is hosed

submits the same page but with the default service level (1). Note also that parameter values persist across **−p** options so the following

    sendpage -l 0 -p sam -p 5551212 Your network is hosed

submits two pages, both of which have service level 0.

OPTIONS

−a *time* Schedule transmission at the indicated *time*. Times are specified using the syntax of the *at*(1) command; e.g. ‘‘now + 30 minutes’’ to schedule transmission 30 minutes in the future. By default, *sendpage* schedules pages for ‘‘now’’. (In SNPP parlance this option specifies the ‘‘hold time’’ for the page; and may result in the page being delivered to the paging terminal but not delivered to the user until the specified time.)

−D Enable notification by electronic mail when the page has been delivered to the service provider. This option is meaningful only when communicating with a *HylaFAX* server.

−f *from* Use *from* as the identity of the message sender. Otherwise the sender is the user that invoked *sendpage*. The sender is an account name to which the *HylaFAX* software should direct email notification messages. The *from* identity may be a fullblown ‘‘From:’’ line, e.g. e.g. ‘‘Joe Blow <joe@foo.com>’’ and ‘‘joe@foo.com (Joe Blow)’’ are acceptable. If no hostpart is specified in the *from* string the local hostname is used. This option is useful when *sendpage* is invoked on behalf of another user, such as in the implementation of an electronic mail to pager gateway.

–h [*modem*@]*host*[*:port*]
Force the jobs to be processed on a specific *host* and, optionally, using a particular *modem*. The *host* may be either a symbolic name or a network address. If no **–h** option is supplied, *sendpage* uses the SNPPSERVER environment variable to identify the SNPP server to which the pages should be directed. If no server is specified then *sendpage* attempts to contact a server on the local host. If no modem is specified, the job will be submitted to any available modem.

–I time
If a job must be requeued because of a communication failure schedule the retry according to the specified *time*. Times are given in seconds. Minutes, hours, and days can be specified with ‘‘*min*’’, ‘‘*hour*’’, and ‘‘*day*’’ suffixes, respectively. By default *HylaFAX* reschedules jobs using retry times that depend on the manner in which the job failed; this interface permits a user to override this algorithm and specify a single fixed retry time for all communication failures. This option is meaningful only when communicating with a *HylaFAX* server.

–l level
Specify a particular service *level* to use in delivering a page. A service level defines both a scheduling priority and the time that a client is willing to wait to find out if delivery is successful or not. Normal delivery is obtained with level 1 (default). The specification identifies level 0 as the highest priority service level with levels 1-7 lower priority and levels 8-11 vendor/server-specific. For *HylaFAX* server machines the mapping between service level and scheduling priority is defined by the configuration of the SNPP server process; see the **PriorityMap** parameter in *hfaxd*(8C)

–n
Send to a numeric-only paging device; i.e. do not send a text message in the paging request.

–N
Do not notify the sender by electronic mail when the page has been delivered to the service provider (default). This option is meaningful only when communicating with a *HylaFAX* server.

–p *PIN*
The Pager Identification Number (PIN) for a terminal that is to receive the message. A message can be sent to multiple destinations by specifiying multiple **–p** options.

–q
Mark pages to be queued on the server and do not wait for their delivery. By default *sendpage* conforms to the SNPP spec by synchronously waiting for each page to be delivered. This option instructs *sendpage* to submit one or pages and then terminate immediately without waiting for their completion. When pages are queued it is advisable to enable email notification with the **–D** or **–R** options. Job queueing is a non-standard function of the SNPP implementation in *HylaFAX* and is not available when submitting pages to non-*HylaFAX* servers.

–R
Enable notification by electronic mail when the message has been delivered and when it is requeued for retransmission. *HylaFAX* will always notify the sender by electronic mail if there is problem delivering a page (irregardless of the email notification setting). This option is meaningful only when communicating with a *HylaFAX* server.

–s *subject*
Set the ‘‘subject’’ for the specified pages according to *subject*. For *HylaFAX* *subject* is used as the identifier string returned in any email notification messages (instead of the usual job identifier number). This option is useful for applications that cross reference notification messages against job submissions.

–t *tries*
Make no more than *tries* attempts to send the page. By default, *HylaFAX* will terminate a job if: 3 consecutive attempts to send a particular message fail, or it appears the receiver is not a service provider. Otherwise *HylaFAX* places no limit on the *number* of attempts to send a page, instead terminating a job if it is not completed within a fixed period of time specified by the service level. Note that a try is a phone call in which carrier is established and the IXO/TAP or UCP protocol is commenced; this is contrasted with a call attempt that might have failed because the line was busy. This option is meaningful only when communicating with a *HylaFAX* server.

−T *dials* Make no more than *dials* phone calls when attempting to send a page. By default, *HylaFAX* will terminate a job if: 12 attempts to reach a service provider fail, or it appears the receiver is not a service provider. Otherwise *HylaFAX* terminates a job if it is not completed within a fixed period of time specified by the *service level*. This option is meaningful only when communicating with a *HylaFAX* server.

−v Trace the protocol between *sendpage* and the SNPP server process that does the delivery work on the remote machine. This information is directed to the standard output.

CONFIGURATION PARAMETERS

sendpage reads configuration information from the files **/usr/share/fax/hyla.conf**, **/usr/share/fax/sendpage.conf**, and **~/.hylarc**; in that order. Configuration files follow the conventions described in *hylafax-client*(1). The following configuration parameters are recognized:

Tag	**Type**	**Default**	**Description**
HoldTime	string	−	hold time to assign to pages
Host	string	localhost	host to contact for service
MailAddr	string	−	mail address for notification messages
MaxDials	integer	12	times to retry dialing
MaxTries	integer	3	times to retry transmission
Notify	string	none	control email notification
Port	integer	444	port to use in contacting server
Protocol	string	tcp	protocol to use in contacting server
QueueSend	boolean	No	whether or not to queue pages
RetryTime	string	−	delay between failed attempts to send
ServiceLevel	integer		SNPP service level for pages
Verbose	boolean	No	whether or not to enable protocol tracing

The configuration parameters are explained below:

HoldTime The hold time to assign to pages. (Equivalent to the **−a** option.)

Host The host to contact for service. (Equivalent to the **−h** option.)

MailAddr The electronic mail address to direct notification messages from the server. If this string is specified without an ‘‘@hostname’’ part then the local hostname will automatically be appended. (Equivalent to the **−f** option.)

MaxDials The maximum number of times to dial the phone for each job. (Equivalent to the **−T** option.)

MaxTries The maximum number of times to retry sending a job. (Equivalent to the **−t** option.)

Notify Control the email notification messages from the server. This string may be one of ‘‘done’’, ‘‘none’’, ‘‘requeued’’ or ‘‘default’’ with an optionally preceding ‘‘when ’’ (e.g. ‘‘when done’’). Note that ‘‘when requeued’’ implies ‘‘when done’’. (Equivalent to the **−D**, **−R**, and **−N** options.)

Port The network port to contact for service. (Eqvuialent to the **−h** option.)

Protocol The name of the communication protocol to use when contacting a server. (Equivalent to the **SNPPSERVICE** environment variable.)

QueueSend Control whether or not to wait for the pages to be delivered. (Equivalent to the **−q** option.)

RetryTime The time to delay between job retries due to a communication failure. (Equivalent to the **−I** option.)

ServiceLevel The SNPP service level to assign to each page. (Equivalent to the **−l** option.)

Verbose Control protocol tracing. (Equivalent to the **−v** option.)

ENVIRONMENT

The following environment variables are used by *sendpage*:

SNPPSERVER The identity of the SNPP server to contact for service. This value is specified as ‘‘[*modem@*]*host*[:*port*]’’ where *host* is either a host name or the Internet ‘‘.’’ (dot) address of the host; *port* is the TCP port number or service name to use (default is ‘‘snpp’’ or 444); and *modem* is an optional modem name or class on the server machine to use in processing the requests (this is meaningful only when the server is running *HylaFAX*).

SNPPSERVICE The communication service specification for communicating with the SNPP server. This value is specified as ‘‘*port*[/*protocol*]’’; by default ‘‘444/tcp’’.

FILES

/usr/tmp/sndpageXXXXXX temporary files

NOTES

The phone number of the pager service provider is stored on the server machine and is not specified by the user; this is a departure from pager support that was supplied with *HylaFAX* version 3.0.

HylaFAX version 3.0 used to recognize a null text message and assume the destination paging device was numeric-only. This is no longer true; if a null text message is submitted then a null message will be included in the paging request. This may cause paging requests submitted to ‘‘real SNPP’’ servers to be rejected if the paging device is in fact numeric-only. When sending to a numeric-only paging device the **−n** option should be used.

BUGS

It is not possible to page text- and non-text devices together; *sendpage* must be run twice, once to send to text-capable devices and once to send to non-text devices.

SEE ALSO

at(1), *hylafax-client*(1), *faxalter*(1), *faxrm*(1), *faxstat*(1), *faxq*(8C), *hfaxd*(8C), *hylafax-server(5F)*

NAME

sendq/q* − *HylaFAX* outbound job description

DESCRIPTION

Files in the **sendq** directory specify transmission job requests. These files are created by *hfaxd*(8C), when submitting a job on behalf of *sendfax*(1) or *sendpage*(1). Job description files also reside in the **doneq** directory; they are moved there by *faxq*(8C) when a job completes.

Job description files are ASCII files with lines of the form

tag: *value*

where a *tag* is one of the identifiers described below and a *value* is either a string or number. Values start at the first non-blank character after the ‘‘:’’ and continue to the end of the line. Lines that end with a backslash character ‘‘\’’ are continued to the next line. Numeric values are decimal integers, except for **chophreshold** which is expressed as a floating point value.

The following table lists the possible tags and the required type of the associated value.

Tag	**Type**	**Description**
chopthreshold	float	page chopping threshold (inches)
commid	string	communication identifier for last call
company	string	receiver’s company
client	string	host that submitted the job
cover	string	continuation coverpage file
data	string	unprocessed document of indeterminate type
!data	string	processed document of indeterminate type
dataformat	string	data format used to transmit a facsimile
desiredbr	integer	desired transmit speed
desireddf	integer	desired page data format
desiredec	integer	desired use of Error Correction Mode (ECM)
desiredst	integer	desired minimum scanline time
desiredtl	integer	whether or not to use **tagline** parameter
donop	string	operation to perform when job is reaped
external	string	external form of dialstring
fax	string	document ready for transmission
groupid	integer	*HylaFAX* job group identifier
jobid	integer	*HylaFAX* job identifier
jobtag	string	user-specified job identifier
jobtype	string	job type identification string
killtime	integer	time to give up trying to send job
location	string	receiver’s location
mailaddr	string	email address of sender
maxdials	integer	max number of times to dial
maxtries	integer	max number of attempts to send job
minsp	integer	minimum required transmit speed
modem	string	outgoing modem to use
ndials	integer	number of consecutive failed attempts to place call
notify	string	email notification specification
npages	integer	number of pages transmitted
ntries	integer	number of attempts to send current page
number	string	dialstring to use in placing call
owner	string	user that owns the job
page	string	PIN in a page operation
pagechop	string	whitespace truncation handling
pagehandling	string	page analysis information
pagelength	integer	facsimile page length (millimeters)
pagewidth	integer	facsimile page width (millimeters)

passwd	string	info to send in HDLC PWD frame
pcl	string	unprocessed PCL document
!pcl	string	processed PCL document
poll	string	polling request
postscript	string	unprocessed POSTSCRIPT document
!postscript	string	processed POSTSCRIPT document
priority	integer	user-specified scheduling priority
receiver	string	receiver's identity
resolution	integer	vertical resolution of facsimile
retrytime	integer	time to use between job retries
schedpri	integer	current scheduling priority
sender	string	identity of sender
signalrate	string	signalling rate at which a facsimile was sent
state	integer	job scheduling state
status	string	job status
subaddr	string	info to send in HDLC SUB frame
tagline	string	tagline format string
tiff	string	unprocessed TIFF/F document
!tiff	string	processed TIFF/F document
totdials	integer	total number of phone calls
totpages	integer	total pages to transmit
tottries	integer	total number of attempts to send job
tts	integer	time to send job
useccover	integer	whether or not to use a continuation cover page

PARAMETERS

Note that all files must be owned by the fax user. Pathnames for document files must be relative to the top of the *HylaFAX* spooling directory and not include "..".

chopthreshold The minimum trailing whitespace (in inches) that must appear on a page for it to be considered for "page chopping". If this parameter is not specified then the value specified by the **PageChopThreshold** configuration parameter is used; see *hylafax-config*(5F).

commid The communication identifier for the last outbound call placed for this job. This string is of the form "XXXXXXXX" where XXXXXXXX is a decimal sequence number in the range 1-99999999.

client The fully qualified hostname for the machine from which the job was submitted. If the hostname is unknown at the time the job is submitted, then the host address in Internet standard "." (dot) notation is supplied instead.

company The receiver's company name as specified with the **−x** option to *sendfax*(1). This value is used only when creating continuation cover pages.

cover The pathname of a *continuation cover page* to send. The file must be TIFF/F (i.e. it must already be prepared for transmission). This document will be transmitted before any other documents listed in the file. See **ContCoverPage** and **ContCoverCmd** in *hylafax-config*(5F).

data The pathname of a document to send. Once a document has been processed, it is marked as **!data** so that subsequent retries do not reprocess the document.

dataformat The format used to transmit facsimile data. This item is recorded by *faxsend*(8C) so that *notify*(8C) can return it in a notification message.

desiredbr The desired signalling rate to use for page data transmissions: 0 for 2400 bps, 1 for 4800 bps, 2 for 7200 bps, 3 for 9600 bps, 4 for 12000 bps, and 5 for 14400 bps (default).

desireddf The desired data format to use for page data transmissions: 0 for 1-D MR, 1 for 2-D MR, 3 for 2-D MR with uncompressed data, and 4 for 2-D MMR (default).

desiredec Whether or not to negotiate the use of Error Correction Mode (ECM): 0 to disable, 1 to enable (default).

desiredst The desired minimum scanline time to use for page data transmissions: 0 for 0 ms (default), 1 for 5 ms, 2 for 10ms/5ms, 3 for 10ms, 4 for 20ms/10ms, 5 for 20ms, 6 for 40ms/20ms, and 7 for 40ms. This value is used in selecting the negotiated session parameters.

desiredtl Whether or not to use the format string specified by the **tagline** job parameter when imaging taglines on document pages. If this value is non-zero then per-job format string is used; otherwise the tagline format string specified in the server configuration file is used; see **TagLineFormat** in *hylafax-config*(5F).

doneop The treatment the job should be given when it is processed by the *faxqclean*(8C) job cleaner program. The may be one of *remove* (to expunge the job description file and all associated documents that are not referenced by another job) or *archive* (to request that the job and associated documents be archived on the server machine). Consult *faxqclean* more details on other information that may be included in this string.

external The external form of the dialstring. This string is used instead of the dialstring in any status and log messages. Clients can protect private information such as access codes that must be placed in dialstrings by supplying a ‘‘cleansed’’ version of the dialstring.

fax The pathname of a TIFF/F document that is ready for transmission. If the pathname string has a leading ‘‘xx:’’, then ‘‘xx’’ is interpreted to be the index of the first directory in the file to transmit (directories are numbered starting at zero).

groupid The job group identifier, an integer number in the range 1 to 32000 assigned to each group of jobs by the *HylaFAX* software. Each job in a group of jobs submitted at the same time is assigned the same value by the *HylaFAX* software.

jobid The job identifier, a unique integer number in the range 1 to 32000 assigned to each job by the *HylaFAX* software.

jobtag A user-specified string that is intended for identification purposes. If this string is defined, it is returned to the user in notification messages instead of the job identifier.

killtime The time to kill this job, expressed in seconds since 00:00:00 GMT, January 1, 1970.

location The receiver’s location as specified with the **−y** option to *sendfax*(1). This value is used only when creating continuation cover pages.

mailaddr The electronic mail address to which notification messages should be directed.

maxdials The maximum number of times the server should dial the phone. If the job cannot be completed in this number of calls, it is terminated.

maxtries The maximum number of times the server should attempt to send a job; where an attempt is defined to be a session that makes it past Phase A of the CCITT T.30 protocol (for facsimile), or similar. If the job cannot be completed in this number of tries, it is terminated.

minsp The minimum signalling rate to use for transmitting page data: 0 for 2400 bps (default), 1 for 4800 bps, 2 for 7200 bps, 3 for 9600 bps, 4 for 12000 bps, and 5 for 14400 bps.

modem The outgoing modem to use to send the job, or ‘‘any’’ if any available modem is acceptable.

owner The identity of the user that owns the job. Typically the owner is the user that created the job, though clients with administrative privileges can change the ownership of jobs. The value is the string specified in the USER command passed to *hfaxd*(8C) when a client logins in to the server.

notify A specification of whether the sender wants notification: whenever the job is requeued after a failed attempt to transmit (‘‘*when requeued*’’), when the job is completed (‘‘*when done*’’), either of the above (‘‘*when done+requeued*’’), or not at all (‘‘*none*’’).

npages The number of pages in the facsimile, including the cover page, that have been successfully transmitted.

ntries The number of unsucessful attempts that have been made to transmit the current page.

ndials The number of consecutive unsucessful attempts to establish carrier with the the remote device.

number The dialstring to use to call the destination device.

page a string with PIN and to use in processing a page transmit request.

pagechop Whether or not to automatically truncate trailing whitespace on pages of outbound facsimile; see **PageChop** in *hylafax-config*(5F). The possible values are: ''*default*'' to use the setting of the **PageChop** configuration parameter, ''*none*'' to disable page chopping, ''*all*'' to enable chopping of all pages, and ''*last*'' to enable chopping of the last page of each document.

pagehandling A string that contains per-page information used during facsimile transmission. Most of the information is used in the negotiation of session capabilities with optional information used in the implementation of page chopping. The string has three characters, say *xxc*, for each page to be transmitted. The *xx* characters are a hex-encoded specification of the session capabilities required to transmit the page. The *c* character is one of ''M'', ''S'', and ''P''; where: ''M'' indicates there is another page to follow and that the page requires different T.30 session parameters be negotiated (an EOM code is to be transmitted). ''S'' indicates there is another page to follow, but the page uses the same session parameters (an MPS. code is to be transmitted). ''P'' indicates this is the last page to be transmitted (an EOP code is to be transmitted). In addition, if page chopping has been enabled and the page has enough trailing whitespace to be a candidate, then each three character string will be followed by a string of the form ''*ZXXXX*'', where ''XXXX'' is a 4 digit hex-encoded count of the number of bytes of data to send for a chopped page.

pagelength The desired page length in millimeters.

pagewidth The desired page width in millimeters.

pcl The pathname of a PCL document to send. Once a PCL document has been processed by *faxsend*, it is marked as **!pcl** so that subsequent retries do not reprocess the document.

poll The value is a Caller Identification String (CIS) to use in making a polling request. If no CIS is specified then *faxsend* will form one from the **FAXNumber** configuration parameter.

postscript The pathname of a POSTSCRIPT® document to send. Once a POSTSCRIPT document has been processed by *faxsend*, it is marked as **!postscript** so that subsequent retries do not reprocess the document.

receiver The receiver's name. This value is used only when creating continuation cover pages.

resolution The desired vertical resolution in lines per inch. This value should be either 98 or 196.

retrytime The time, in seconds, to delay between each attempt to transmit a job that fails because of a communication-oriented problem. Normally jobs are scheduled according to a collection of configuration parameters that are tuned for the specific reason of each failure; e.g. see **JobReqProto** in *hylafax-config*(5F).

sender The ''full name'' of the sender of the facsimile.

signalrate The signalling rate used to transmit facsimile data. This item is recorded by *faxsend*(8C) so that *notify*(8C) can return it in a notification message.

state The job scheduling state. Recognized values are: 1 (*suspended*, not being scheduled), 2 (*pending*, waiting for the time to send), 3 (*sleeping*, waiting for a scheduled timeout), 4 (*blocked*, waiting for concurrent activity to the same destination to complete), 5 (*ready*, ready to be processed except for available resources), 6 (*active*, actively being processed by *HylaFAX*), 7 (*done*, processing completed).

status Status information about the current state of the job. Typically this is the reason why the last attempt to send the facsimile failed.

subaddr The SubAddress information to transmit when sending a facsimile. This string should be used by the receiver to route received facsimile documents.

tagline The format string to use in imaging taglines on outgoing documents. Note that this parameter is only used if the **desiredtl** parameter is non-zero.

tiff The pathname of a TIFF/F document to send. Once a TIFF document has been processed by *faxsend*, it is marked as **!tiff** so that subsequent retries do not reprocess the document.

totdials The total number of times the server has placed a call to the receiving phone number.

totpages The total number of pages to be transmitted.

tottries The total number of attempts to transmit a job; see **maxtries** above.

tts The time to send the job. If this value is non-zero, it is number of seconds since 00:00:00 GMT, January 1, 1970. Otherwise, a value of 0 is interpreted to mean process as soon as possible.

useccover 1 if the job should use a continuation cover page during retransmission, 0 otherwise. See *hylafax-server*(5F) for a description of continuation cover pages.

SEE ALSO

sendfax(1), *faxq*(8C), *faxqclean*(8C), *faxsend*(8C), *pagesend*(8C), *hfaxd*(8C), *hylafax-config*(5F)

NAME

sgi2fax − convert a Silicon Graphics image file for facsimile transmission

SYNOPSIS

/usr/sbin/sgi2fax [*options*] *files*...

DESCRIPTION

sgi2fax converts Silicon Graphics format images to bilevel TIFF Class F images in preparation for transmission as facsimile. *sgi2fax* is not normally invoked directly; it is usually run by *sendfax*(1) when an SGI image file is submitted for transmission.

Images are converted by scaling them to fill the guaranteed reproducible area on a default page. Each RGB pixel *p* is converted to greyscale according to:

$$p_r * \%r + p_g * \%g + p_b * \%b$$

where %r, %g, and %b default to 30, 59, and 11, respectively. Greyscale images are then process with a high pass filter and dithered to a bilevel image. The resultant image are written as a TIFF Class F document.

By default, the output image is created with 204 dots/inch horizontal resolution and 98 lines/inch vertical resolution. The image is written as Group 3 1D-encoded data.

OPTIONS

−1 (**−2**) Write the image as Group 3 1D- (2D-) encoded data.

−b *%b* Set the *%b* parameter used to convert color pixel values to greyscale pixel values.

−g *%g* Set the *%g* parameter used to convert color pixel values to greyscale pixel values.

−l Generate an image at 98 lines/inch vertical resolution. know as *low resolution*.

−m Generate an image at 196 lines/inch vertical resolution.

−o *file* Write output to *file* instead of the default file named **sgi.fax**.

−r *%r* Set the *%r* parameter used to convert color pixel values to greyscale pixel values.

−s *size* Set the output size according to the named page size. Images are normally created with a system-default page size (usually letter-size pages, 8.5" by 11", for sites in North America). Alternate page sizes are specified symbolically using either the name or abbreviation of an entry in the *pagesizes*(5F) database; e.g. *a3* (ISO A3), *a4* (ISO A4), *a5* (ISO A5), *a6* (ISO A6), *b4* (ISO B4), *na-let* (North American Letter), *us-leg* (American Legal), *us-led* (American Ledger), *us-exe* (American Executive), *jp-let* (Japanese Letter), and *jp-leg* (Japanese Legal). Comparisons are case-insensitive and any match of a substring of the full page-size name is sufficient; e.g. ‘‘legal’’ would match ‘‘American Legal’’.

FILES

/usr/share/fax/pagesizes page size database

SEE ALSO

hylafax-client(1), *sendfax*(1), *hylafax-server(5F)*

NAME

status − server status database

DESCRIPTION

Files in the **status** directory in the *HylaFAX* spooling area contain status information about each running server. Files are named according to the modem being serviced. Files without a suffix contain a single line of ASCII text that indicates what the server is currently doing: ‘‘Running and idle’’, ‘‘Sending to ...’’, ‘‘Answering the phone’’, etc. The contents of the file **any.info** is returned by the *faxstat*(1) program when given the **−i** option; it should contain information about the server system including an administrative contact, its location, usage conventions, etc.

SEE ALSO

faxq(8C), *faxsend*(8C), *pagesend*(8C), *faxgetty*(8C), *hfaxd*(8C), *faxstat*(1)

NAME

tagtest − *HylaFAX* tag line testing program

SYNOPSIS

/usr/sbin/tagtest [*options*] input.tif

DESCRIPTION

tagtest is a program for testing the ‘‘tag line’’ support in the *HylaFAX* server. *tagtest* takes a TIFF/F (TIFF Class F) file and generates a new TIFF/F file that is a copy of the input file, but with sample tag lines imaged across the top of each page. Options are provided for specifying a tag line format string and a tag line font file. If neither are specified, then *tagtest* will use builtin defaults. *tagtest* is designed for testing tag line format strings and fonts before they are configured for use by the facsimile server.

OPTIONS

−f *file* Image the tag line format string using the Portable Compiled Font (PCF) in *file*. By default *tagtest* looks for the font file **fixed.pcf**.

−m *format* Use *format* when imaging tag lines. If this option is not specified, then *tagtest* uses the format string ‘‘From %%n|%c|Page %%p of %%t’’.

−o *file* Write output to *file*. By default output is written to the file **t.tif**.

SEE ALSO

faxsend(8C), *hylafax-config*(5F)

NAME

textfmt – convert text to POSTSCRIPT® for facsimile transmission

SYNOPSIS

/usr/sbin/textfmt [*options*] [*files...*]

DESCRIPTION

textfmt takes ASCII text and generates formatted POSTSCRIPT® suitable for facsimile transmission. It is usually invoked as a filter program by the *sendfax*(1) program. Input files may be specified on the command line. If none are given, then *textfmt* will read from standard input.

By default, *textfmt* will generate an output document that is formatted with a single vertical column of text, set in a 10 point fixed width font, and with a simple page header that consists of the file name, last modification date, and the page number. Pages are generated according to the default page dimensions (see *pagesizes*(5F)). When multi-column output is requested, text is set by default with a 7 point fixed width font.

OPTIONS

Many of the options to *textfmt* have been chosen for compatibility with the Adobe Systems' *enscript* program; other options are for features that are not provided by *enscript*.

–1 Set text in one column.

–2 Set text two columns.

–B Suppress page headers.

–c Clip long text lines instead of wrapping them at page margins.

–D Do not use ISO 8859-1 encoding in the generated POSTSCRIPT.

–f *font* Set text using the POSTSCRIPT font named *font*.

–F *fontdir(s)*
Path(s) to afm font files.

–G Use a gaudy-style page header.

–m *cols*
Set text in *cols* columns.

–Ml=*#***,r=***#***,t=***#***,b=***#*
Set the page margins; the default margins are: left and right .25 inch, top .85 inch, bottom .5 inch.

–o Draw an outline around each column of text; the specified number defines the width of the line that is drawn.

–p *pts* Set text using fonts with *pts* point size.

–r Set pages rotated by 90 degrees (in ''Landscape mode'').

–R Set pages unrotated (in ''Portrait mode'').

–s *size* Set the output size according to the named page size. Images are normally created with a system-default page size (usually letter-size pages, 8.5" by 11", for sites in North America). Alternate page sizes are specified symbolically using either the name or abbreviation of an entry in the *pagesizes*(5F) database; e.g. *a3* (ISO A3), *a4* (ISO A4), *a5* (ISO A5), *a6* (ISO A6), *b4* (ISO B4), *na-let* (North American Letter), *us-leg* (American Legal), *us-led* (American Ledger), *us-exe* (American Executive), *jp-let* (Japanese Letter), and *jp-leg* (Japanese Legal). Comparisons are case-insensitive and any match of a substring of the full page-size name is sufficient; e.g. ''legal'' would match ''American Legal''.

–U Reverse collate pages; that is the last page will appear first in the resultant POSTSCRIPT, then the next-to-last page, etc.

−V *spacing*
Set the line height; the default height is 1.2 x point size.

CONFIGURATION PARAMETERS

textfmt reads configuration information from the files **/usr/share/fax/hyla.conf**, and **˜/.hylarc**; in that order. Configuration files follow the conventions described in *hylafax-client*(1). The following configuration parameters are recognized:

Tag	Type	Default	Description
Columns	integer	1	number of columns in formatted output
GaudyHeaders	boolean	No	enable/disable guady page headers
FontMap	string	*see below*	location of ghostscript Fontmap file
FontPath	string	*see below*	directory for font metric files
ISO8859	boolean	Yes	use ISO 8859-1 character encoding
LineWrap	boolean	Yes	wrap/truncate text lines
Orientation	string	portrait	orientation of text on page
OutlineMargin	inches	0	width of outline line
PageCollation	string	forward	collate pages in forward or reverse direction
PageHeaders	boolean	Yes	enable/disable page headers
PageHeight	float	−	output page height
PageMargins	string	*see below*	margins for formatted page
PageSize	string	default	output page size from database
PageWidth	float	−	output page width
TabStop	integer	8	inter-stop setting in characters
TextFont	string	Courier	name of font for setting text
TextLineHeight	inches	−	text formatting line height control
TextPointSize	inches	*see below*	size to use in setting text

The configuration parameters are explained below. Most parameters correspond to a command line option. Parameter values identified above as *inches* are converted according to the following conventions (where “#.##” is a decimal number):

#.##bp	big point (1in = 72bp)
#.##cc	cicero (1cc = 12dd)
#.##cm	centimeter
#.##dd	didot point (1157dd = 1238pt)
#.##in	inch
#.##mm	millimeter (10mm = 1cm)
#.##pc	pica (1pc = 12pt)
#.##pt	point (72.27pt = 1in)
#.##sp	scaled point (65536sp = 1pt)

Unit names can be upper or lower case but no white space is permitted between the number and the unit. Values specified with no unit are interpreted as points.

Columns The number of columns to set text in. (Equivalent to the **−m** option.)

FontMap The directory or directories where the ghostscript Fontmap file(s) are located; multiple paths are seperated by a colon (":"). The Fontmap table allows the translation of the descriptive name of the font (as used by the *TextFont* parameter) to the ghostscript font filenames. Setting the value to "" (zero length string) disables this translation and provides functionality with RIPs other than ghostscript.

FontPath The path where Adobe Font Metric (AFM) files are located; by default /usr/share/ghostscript/5.50:/usr/share/ghostscript/5.50/vflib:/usr/share/ghostscript/fonts:/usr/share/fonts/defa font/pfb. (Equivalent to the **−F** option.)

GaudyHeaders Control whether or not to use a gaudy-style page header. (Equivalent to the **−G** option.)

ISO8859 Control the use of ISO 8859-1 encoding in the generated POSTSCRIPT (Equivalent to the **−D** option.)

LineWrap Control whether long text lines are wrapper or truncated at the right hand margin. (Equivalent to the **−c** option.)

Orientation Control whether pages are oriented horizontally (‘‘landscape’’) or vertically (‘‘portrait’’). (Equivalent to the **−r** and **−R** options.)

OutlineMargin Control whether columns of text have a line drawn around them and the width of the line. Setting this parameter to 0 disables outlines. (Equivalent to the **−o** option.)

PageCollation Control whether the output file has pages collated in the same order as the input file (‘‘forward’’) or in reverse order (‘‘reverse). (Equivalent to the **−U** option.)

PageHeaders Control whether page headers are generated. (Equivalent to the **−B** option.)

PageHeight Set the output page height in inches (in inches).

PageMargins Set the output page dimensions. Dimensions are specified as string of the form: ‘‘**l=#**,**r=#**,**t=#**,**b=#**’’ where **l** indicates the left margin, **r** indicates the right margin, **t** indicates the top margin, **b** indicates the bottom margin, and numbers are interpreted as *inches*. (Equvalent to the **−M** option.)

PageSize Set the output page dimensions by name. (Equivalent to the **−s** option.)

PageWidth Set the output page width in inches (in inches).

TabStop Set the tab stop distance; in characters.

TextFont Set the descriptive name of font to use for setting text. (Equivalent to the **−f** option.)

TextLineHeight Set the vertical text line height and spacing. (Equvalent to the **−V** option.)

TextPointSize Set the point size to use in setting text. (Equvalent to the **−p** option.)

NOTES

If *textfmt* is unable to locate font metric information for a font, then it will use a fixed-width metric that is 60% of the text point size; this metric will almost certainly result in incorrectly formatted POSTSCRIPT.

textfmt is distantly related to the *lptops* program written by Don Beebe.

FILES

/usr/share/fax/hyla.conf system-wide configuration file
~/.hylarc per-user configuration file
/usr/share/fax/pagesizes page size database
/usr/share/ghostscript/5.50:/usr/share/ghostscript/5.50/vflib:/usr/share/ghostscript/fonts:/usr/share/fonts/default/ghostscript:/us

SEE ALSO

hylafax-client(1), *sendfax*(1), *pagesizes*(5F)

NAME

tiff2fax − convert TIFF for facsimile transmission by *HylaFAX*

SYNOPSIS

/var/spool/hylafax/bin/tiff2fax [*options*] *file*

DESCRIPTION

bin/tiff2fax takes TIFF input and generates a TIFF Class F image file suitable for facsimile transmission. It is usually invoked by the *HylaFAX* scheduler process, *faxq*(8C). The default implementation uses the *tiffcheck*(8C) program and tools from the freely available TIFF software distribution to implement the conversion operations.

OPTIONS

The following operations are recognized:

−1 Write Group 3 1D-encoded data to the output file.

−2 Write Group 3 2D-encoded data to the output file.

−m *pages* Set the maximum number of pages that are permitted in the output file. If more pages are present in the input document then abort processing.

−o *file* Write output to the specified *file* instead of the default **foo.tif** filename.

−w *width* Set the output page *width*, in pixels. By default, pages are 1728 pixels wide.

−l *length* Set the output page *length* in millimeters. By default, pages are 297 millimeters long.

−r *vres* Set the vertical resolution, in lines/inch, at which to image the POSTSCRIPT.

FILES

/usr/sbin/tiffcheck	TIFF file conversion examiner
/var/spool/hylafax/bin/ps2fax	POSTSCRIPT RIP for doing complicated conversions
/usr/bin/tiffcp	for converting compression schemes
/usr/bin/tiff2ps	for colorspace and/or image size conversions

BUGS

Conversions that require re-imaging and resizing may result in images that are poorly placed on the page.

There is no way to control how images are treated when they are resized; e.g. the original aspect ratio is not always maintained.

SEE ALSO

sendfax(1), *faxq*(8C), *hylafax-server(5F)*

NAME

tiffcheck − check TIFF for *HylaFAX* conversion requirements

SYNOPSIS

/usr/sbin/tiffcheck [*options*] *file.tif*

DESCRIPTION

tiffcheck takes TIFF input and writes a multi-line text message to the standard output describing any work needed to prepare the images for facsimile transmission. *tiffcheck* is usually invoked by *tiff2fax*(8C) script that is used by the *HylaFAX* scheduler to prepare TIFF documents submitted for transmission.

The first line of output is a series of ‘‘+’’-separated words that indicate if the input file should be:

- rejected (‘‘REJECT’’),
- reformatted with a different data-encoding scheme (‘‘REFORMAT’’),
- scaled to different page dimensions (‘‘RESIZE’’),
- reimaged in a different vertical resolution (‘‘REVRES’’), or
- reimaged for other reasons such as conversion from color to black&white (‘‘REIMAGE’’).

The conversion requirements string will always have keywords in the above order; e.g. ‘‘RESIZE+REVRES’’ if the input file needs to be scaled and converted to a different resolution. Information following the first line is descriptive text that explains the reason(s) for the file conversion(s); this text is suitable for return to a client (should document conversion fail or not be attempted). The descriptive text is ordered identically to the keywords in the conversion string; this makes it possible to match up conversion requirements to reasons.

OPTIONS

The following operations are recognized:

−1 Set output to be Group 3 1D-encoded.

−2 Set output to be Group 3 2D-encoded.

−w *width* Set the output page *width*, in pixels. By default, pages are 1728 pixels wide.

−l *length* Set the output page *length* in millimeters. By default, pages are 297 millimeters long.

−r *vres* Set the vertical resolution, in lines/inch, at which to image the .

EXAMPLES

The following color image needs to be converted to black&white, resized to the output page dimensions, and reformatted to use Group 3 1D-encoded data.

```
hyla% /usr/sbin/tiffcheck jello.tif
REFORMAT+RESIZE+REIMAGE
Document is not a bilevel image (bits/sample 8).
Document requires reformatting, not in Group 3 format.
Document requires resizing to adjust page width (convert to 1728, document is 256).
```

This image is a facsimile received at high resolution (196 lpi); it only needs to be converted to 98 lpi.

```
hyla% /usr/sbin/tiffcheck faxix.tif
REVRES
Document requires reformatting to adjust vertical resolution (convert to 98, document i
```

SEE ALSO

tiff2fax(8C), *faxq*(8C), *hylafax-server*(5F)

NAME

tsi − Transmission Subscriber Identification (TSI) access control list

DESCRIPTION

The *HylaFAX* configuration parameter **QualifyTSI** specifies whether or not the identity of a calling facsimile machine should be checked against an access control list before receiving facsimile. If **QualifyTSI** is non-null, then only messages from facsimile machines identified in the file specified by the string (typically **etc/tsi**) will be accepted.

Patterns are specified one per line and must conform to the regular expressions syntax specified by POSIX 1003.2; see *re_format*(7). Comments may be included; they are introduced with the "#" character and extend to the end of the line. Any trailing white space on a line is ignored (for convenience when comments are used).

If a line begins with "!", then the regular expression identifies clients that should be *rejected*; otherwise regular expressions identify clients whose transmissions should be *accepted*. The order of patterns in a TSI file is important. When a facsimile is to be received, the fax server will compare the client's TSI against the patterns in the access control list in the order in which they appear in the file. The first pattern that matches the client TSI is used to decide whether to accept or reject the facsimile. If no patterns match the client TSI then the facsimile is rejected. Thus if you want to accept all but a restricted set of TSI the last line in the file should be "^.*$".

Note that regular expression patterns should be written to match a TSI exactly. That is, patterns should be of the form:

```
^<pattern>$
```

where the "^" and "$" characters are used to specify the start and end of the matching TSI. Additionally, regular expression patterns should handle white space that may appear in known locations. For example,

```
^([+]1){1}[ .−]*415[ .−]*555[ .−]*1212.*$
```

matches the following TSI strings:

```
+1.415.555.1212
    415  555  1212
1−415−555−1212
```

Finally, note that regular expressions can be used to specify many TSI with one pattern.

NOTES

It would be nice if TSI that were to be matched against were placed in some canonical form (e.g. remove white space and white space-like characters). This is, however, problematic, because some facsimile machines permit any printable ASCII string to be sent as a TSI.

SEE ALSO

faxgetty(8C), *re_format*(7), *hylafax-config*(5F)

NAME

tsitest − *HylaFAX* TSI access control list checking program

SYNOPSIS

/usr/sbin/tsitest [**−q**] *ACLfile*

DESCRIPTION

tsitest is an interactive program for the testing of the TSI access control list (ACL) used by the *faxgetty*(8C) server processes to screen incoming facsimile. The *ACLfile* specified on the command line is expected to contain TSI control expressions as described in *tsi*(5F). Strings typed at the prompt are then checked against the list and *tsitest* prints out whether or not the string is an acceptable TSI.

Note that *tsitest* reads the control list file only when it is initially started up. This means that *tsitest* must be restarted for it to recognize changes to *ACLfile*.

OPTIONS

−q Operate *quietly*; *tsitest* will print only whether an input string is accepted or rejected.

SEE ALSO

faxgetty(8C), *hylafax-server*(5F), *hylafax-config*(5F), *tsi*(5F)

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 two 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/share/fax/typerules**, an ASCII file that is patterned after the **/etc/magic** file used by the System V *file*(1) program.

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 wth 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), 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):

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

If no operation is specified then ''='' is used.

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 mm
%W	page width in pixels
%l	page length in mm
%L	page length in pixels
%s	page size by name
%F	the directry 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)

NAME

wedged − *HylaFAX* modem wedged script

SYNOPSIS

/var/spool/hylafax/bin/wedged *devid device*

DESCRIPTION

bin/wedged is the command script invoked by the facsimile server when a modem is deemed to be in an irreparable state and requires operator intervention. This decision is based on the setting of the **MaxSetupAttempts** configuration parameter described in *hylafax-config*(5F). The default *wedged* script sends electronic mail to the FaxMaster alias and attempts to disable any getty processes spawned by the *init*(8C) process. In addition the *faxq*(8C) process automatically disables use of the modem for servicing outbound jobs.

The *devid* argument is the identifier derived from the name of the modem's device file by substituting "_" characters for "/" characters; e.g. *term_10* for */dev/term/10*. The *device* argument is the name of the character special device associated with the modem.

FILES

/usr/sbin/sendmail for delivering mail

SEE ALSO

faxq(8C), *hylafax-config*(5F)

NAME

xferfaxlog − *HylaFAX* activity log

DESCRIPTION

The file **etc/xferfaxlog** contains information about inbound and outbound activities. The file contains one line per inbound or outbound call (except for facsimile documents retrieved by polling in which case multiple entries may be present for a single call). Lines are fixed-format, *tab-separated* ASCII strings. Each record of a facsimile transmission is of the form:

date SEND commid modem jobid jobtag sender ‘‘dest-number’’ ‘‘CSI’’ params #pages jobtime conntime ‘‘reason’’

A facsimile reception record is of the form:

date RECV commid modem <*null*> <*null*> fax ‘‘local-number’’ ‘‘TSI’’ params #pages jobtime conntime ‘‘reason’’

Each facsimile document retrieved by polling has a record of the form:

date POLL commid modem jobid jobtag sender ‘‘dest-number’’ ‘‘TSI’’ params #pages jobtime conntime ‘‘reason’’

An alphanumeric pager request has a record of the form:

date PAGE commid modem jobid jobtag sender ‘‘dest-number’’ ‘‘<*null*>’’ 0 0 jobtime conntime ‘‘reason’’

The following describes the fields in the above records:

date The date and time of the transaction in the format MM/DD/YY HH:MM , where MM is the numeric month, DD the numeric day, YY the last two digits of the year, and HH:MM is the time in 24-hour format.

commid The communication identifier for the call.

modem The device identifier for the modem that was used to do the send or receive.

jobid The job number for outbound calls.

jobtag The client-specified job tag for outbound calls.

sender The sender/receiver’s electronic mailing address (facsimile receptions are always attributed to the ‘‘fax’’ user).

dest-number The phone number dialed for outgoing calls.

TSI The Transmitter Subscriber Identification string (as received) for incoming calls.

CSI The Caller Subscriber Identification string of the remote machine (as reported) for outgoing calls.

local-number The local phone number on which the data was received.

params The negotiatated facsimile session parameters used for transferring data encoded as described below.

#pages The total number of pages transferred.

jobtime The duration of the session; in the format HH:MM:SS. This time includes setup overhead and any time spent cleaning up after a call.

conntime The time spent on the phone; in the format HH:MM:SS. This should be the time used by the PTT to calculate usage charges.

reason A string that indicates if any problem occured during the session.

Note that fields may have embedded blanks. Session parameters are encoded as a decimal number that contains a packed set of Class 2 session parameter values. The format is: 2 bits for vertical resolution, 3 bits for signalling rate, 3 bits for page width, 2 bits for page length, 2 bits for data format, 1 bit for ECM use, 1 bit of BF use, and 3 bits for the min scanline time. The page counts are always integral numbers. Partial pages that are to be retransmitted are not included in the page count.

The simple ASCII format was designed to be easy to process with programs like *awk*(1).

NOTES

The sender field does not necessarily represent the submitter's actual identity. For example, it reflects the value given by the *sendfax*(8C) ‘‘-f’’ option. Be cautious that this field is not utilized for auditing if the fax user base is not trusted.

BUGS

The date format will ‘‘break’’ in the year 2000. Information should be recorded on a per-page basis for facsimile communications.

SEE ALSO

faxq(8C) *faxsend*(8C) *faxgetty*(8C)

NAME

xferfaxstats − summarize *HylaFAX* transmission accounting

SYNOPSIS

/usr/sbin/xferfaxstats [*options*] [*files*]

DESCRIPTION

xferfaxstats is a command script that prints a summary of the *HylaFAX* accounting statistics for transmitted jobs. The statistics can be sorted using one of several keys: the sender's account name (default), the job tag (useful when used for accounting purposes), the remote party's CSI, the called phone number, the output device used for the call, the communication speed, and the communication data format. Statistics are read from the file **/var/spool/hylafax/etc/xferfaxlog** in the spooling area, unless filenames are specified on the command line.

The following table is an example of the output from *xferfaxstats* (using the default sort key):

Destination	**Pages**	**Time**	**Pg/min**	**Errs**	**TypRate**	**TypData**
cadsmith@esd.sgi.com	34	24:05	1.4	0	9600	1-D MR
ctb@corp.sgi.com	32	20:43	1.5	2	9600	2-D MR
edmark@esd.sgi.com	42	29:30	1.4	3	9600	1-D MR
fredc@mfg.sgi.com	157	1:37:52	1.6	0	9600	2-D MR
jlw@engr.sgi.com	81	57:29	1.4	1	9600	1-D MR
jlw@wpd.sgi.com	346	3:57:57	1.5	100	9600	1-D MR
karlton@asd.sgi.com	68	36:57	1.8	0	9600	1-D MR
kerrie@mfg.sgi.com	48	32:29	1.5	1	9600	2-D MR
kgraf@mfg.sgi.com	128	1:48:21	1.2	2	9600	1-D MR
kubey@wpd.sgi.com	264	3:22:13	1.3	6	9600	1-D MR
loki@wpd.sgi.com	205	2:56:45	1.2	5	9600	2-D MR
moondog@esd.sgi.com	112	1:23:01	1.3	4	9600	2-D MR
mpolen@mti.sgi.com	122	1:39:32	1.2	1	9600	1-D MR
nancyf@mfg.sgi.com	174	2:08:44	1.4	2	9600	2-D MR
paulh@mfg.sgi.com	56	1:29:24	0.6	0	9600	2-D MR
richardn@mfg.sgi.com	34	24:42	1.4	1	9600	1-D MR
rosa@corp.sgi.com	32	28:00	1.1	2	9600	1-D MR
sam@asd.sgi.com	65	46:25	1.4	5	9600	1-D MR
seth@asd.sgi.com	34	25:04	1.4	0	9600	1-D MR
--						
Total	2034	25:49:13	1.3	135		

The first column column shows the sort key value used (the sender's identity here, but otherwise the destination phone number, remote CSI, etc.). The *Pages* column is the total number of good pages transmitted. The *time* column shows the total amount of time spent transmitting. The *Pg/min* column displays the average transfer rate. The *Errs* column indicates how many errors were encountered while transmitting facsimile. (Note that this statistic does not include problems encountered while placing a call; e.g. busy signals.) The *TypRate* column displays the most common communication speed with which facsimile were transmitted. The *TypData* column shows the most common data format in which facsimile data were transmitted.

By default, account names other than ''root'' and ''guest'' are automatically mapped to a canonical format in which the host name is removed and identical addresses in the same domain are merged together. For example, ''sam@flake.asd.sgi.com'' and ''sam@foobar.asd.sgi.com'' are both mapped to ''sam@asd.sgi.com'' with the associated statistics accumulated. Identical account names in different domains generate a warning; e.g. ''loki@wpd.sgi.com'' and ''loki@corp.sgi.com''. If the **−nomap** option is specified, then the above mapping is not done and account names are not altered.

OPTIONS

–csi* Sort output by the remote party's CSI.

–dest* Sort output by the phone number called.

–dev* Sort output by the output device (i.e. modem) used.

–format Sort output by the negotiated data format.

–job* Sort output by client-specified job tag identifiers.

–nomap Supress the normal account name mapping by which statistics for the ''root'' and ''guest'' acounts are merged and leading hostname part of domain names are stripped.

–send* Sort output by sender's account name (default).

–since *time* Show information only for the facsimile transmitted since the specified *time*; where *time* is of the form ''MM/DD/YY HH:MM'' (the *date*(1) format string %D %H:%M).

–speed Sort output by the signalling rate.

DIAGNOSTICS

''**Warning, address clash, "%s" and "%s".**'' During address mapping two mapped addresses were found for the same account name.

''**Warning, weird user address/name "%s".**'' During address mapping an address was encountered that multiple ''@'' symbols in it.

NOTES

The sender field does not necessarily represent the submitter's actual identity. For example, it reflects the value given by the *sendfax*(8C) ''-f'' option. Be cautious that this field is not utilized for auditing if the fax user base is not trusted.

FILES

/var/spool/hylafax	spooling area
etc/xferfaxlog	default file from which to read statistics

SEE ALSO

faxq(8C), *faxsend*(8C), *recvstats*(8C), *hylafax-log*(5F)