

**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[NB: The remainder of this manual page is out of date.]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 parameter 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 succeed.

**%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 *prioctl* system call to set the server process priority failed.

**prioctl(%s): %m.** (*Only on SVR4-style systems.*) A *prioctl* 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 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 1DMH 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.

**Bad HDLC terminating flag received..** During Class 1 ECM; a terminating flag for an HDLC frame was corrupt.

**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 length invalid.** An ECM HDLC protocol message was excessively long.

**HDLC frame not byte-oriented. Trailing byte: %#x.** An ECM HDLC protocol message was received with a number of bits incompatible with necessary byte-orientation.

**HDLC frame too short (%u bytes).** An HDLC protocol message was not at least 3 bytes long outside of ECM protocol or at least 5 bytes long within ECM protocol.

**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.

**RCV: bad code word 0x%x, len %d, row %lu.**

**RCV: begin page.** During a receive operation a new page of data is about to be received on the high speed message carrier.

**RCV: end page.** During a receive operation, receipt of page of data on the high speed message carrier just completed.

**RCV frame FCS check failed.** During ECM protocol, a received HDLC frame was received which fails the CRC check and the frame was marked as corrupt.

**RCV: 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.

**RCV: 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.

**RCV: %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.

**RCV: %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.

**RCV received %u frames of block %u of page %u.** When the server does Class 1 ECM; a complete block has been received with a given number of frames for a named page.

**RCV received assumed RCP frame with block end.** When the server does Class 1 ECM; the block-end indicator was received without a proper RCP frame and the receipt of an RCP frame was therefore assumed or feigned.

**RCV received frame number %u.** When the server does Class 1 ECM; the numbered image frame has been received.

**RCV received RCP frame.** When the server does Class 1 ECM; the first RCP frame was received, which signifies the end of the image block.

**RCV: 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.

**RCV: 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.

**RCV sent fourth PPR.** During Class 1 ECM; the server has rejected a received image block consecutively four times at which point the sender is allowed to either continue to retransmit the block or to end retransmissions.

**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 1DMH 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	during 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 <i>getty</i> 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)*