

**NAME**

config – *HylaFAX* configuration database

**DESCRIPTION**

*HylaFAX* uses a configuration file for the central queuing 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 <sup>1</sup> are used only by the *HylaFAX* scheduler process and should appear only in the file named **config**. Parameters marked with a <sup>2</sup> 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.

<b>Tag</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
AdaptiveAnswer	boolean	No	enable adaptive answer of inbound calls
AdminGroup	string	faxadmin	System user group for administration (if PAM enabled)
AnswerRotary	string	Any	alternatives for answering calls
AnswerBias	integer	–	bias to apply to successful rotary answer
AreaCode <sup>2</sup>	string	–	local area code
AuditHook <sup>1</sup>	string	–	command to run for faxq event auditing
BadPageHandlingMethod	string	RTN-SAVE	bad page received handling method
BatchLogs <sup>1</sup>	boolean	Yes	keep all session logs of a batch in a single log
CallIDAnswerLength	integer	–	answer call when CallIDPattern received
CallIDPattern	string	–	call identification pattern string
CIDName	string	–	equivalent to CallIDPattern (2)
CIDNameAnswerLength	integer	0	equivalent to CallIDAnswerLength (2)
CIDNumber	string	–	equivalent to CallIDPattern (1)
CIDNumberAnswerLength	integer	0	equivalent to CallIDAnswerLength (1)
ClocalAsRoot	boolean	No	set CLOCAL using root UID
ContCoverCmd <sup>1</sup>	string	bin/mkcover	continuation cover page generation script
ContCoverPage <sup>1</sup>	string	–	continuation cover page template filename
CountryCode <sup>2</sup>	string	–	local country code
CountSkippedPages	boolean	Yes	Count skipped pages in the page counts/totals when sending
DestControls <sup>1</sup>	obsolete	–	Obsoleted by JobControl
DeviceMode	octal	0600	protection mode to use for modem device
DialStringRules <sup>2</sup>	string	–	dial string rules file

DistinctiveRings	string	–	configuration for distinctive ring cadences
DRingOff	string	–	distinctive ring “off” cadence indicator
DRingOn	string	–	distinctive ring “on” cadence indicator
DynamicConfig	string	–	script for dynamic receive configuration
EGettyArgs	string	–	arguments passed to external getty program
FAXNumber	string	–	facsimile modem phone number
FaxRcvdCmd	string	bin/faxrcvd	notification script for received facsimile
GettyArgs	string	–	arguments passed to getty program
Include <sup>2</sup>	string	–	include another file
InternationalPrefix <sup>2</sup>	string	–	dialing prefix for international calls
JobControlCmd <sup>1</sup>	string	–	job control command
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 <sup>2</sup>	integer	300	requeue interval for “other” problems
JobReqProto	integer	60	requeue interval after fax protocol error
JobRetryBusy	integer	–	number of retries for BUSY dial result
JobRetryDataConn	integer	–	number of retries for data connection dial result
JobRetryNoAnswer	integer	–	number of retries for NO ANSWER dial result
JobRetryNoCarrier	integer	1	number of retries for NO CARRIER dial result
JobRetryOther	integer	–	number of retries for “other” problems
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
LogCalls	boolean	Yes	Log all incoming calls as CALL records in xferfaxlog
LogFacility <sup>2</sup>	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
MaxBatchJobs <sup>1</sup>	integer	64	max jobs in a batch
MaxConcurrentCalls <sup>1</sup>	integer	1	max concurrent jobs to process for a destination
MaxConsecutiveBadLines	integer	5	max consecutive bad rows for OK page quality
MaxDials <sup>1</sup>	integer	12	max phone calls to make to transmit a job
MaxRecvPages	integer	<i>unlimited</i>	max pages to permit in a receive
MaxSendPages <sup>1</sup>	integer	<i>unlimited</i>	max pages to permit in a send
MaxSetupAttempts	integer	2	max attempts to initialize a modem
MaxTries <sup>1</sup>	integer	3	max attempts to transmit a job
MinAcceptedLineCount	integer	10	min number of rows for OK page quality
ModemGroup <sup>1</sup>	string	–	define a name for a set of modems
ModemPriority	integer	255	scheduling priority for outbound jobs
ModemReadyState	string	R	“ready state” sent by <i>faxgetty</i>
ModemRingResponse	string	–	command to respond after hearing RING
ModemRingsBeforeResponse	integer	0	the number of rings before ModemRingResponse
NoAnswerVoice	boolean	false	disable the answering of voice-indicated calls
NoCarrierRetrys	integer	–	Deprecated - See JobRetryNoCarrier
NotifyCmd <sup>1</sup>	string	bin/notify	user notification command script
PageChop <sup>1</sup>	string	last	control automatic truncation of trailing whitespace
PageChopThreshold <sup>1</sup>	float	3.0	whitespace truncation threshold (inches)
PCL2FaxCmd <sup>1</sup>	string	bin/pcl2fax	PCL RIP command script
PercentGoodLines	integer	95	percentage of good rows for OK page quality
PollLockPokeModem	boolean	false	check on modem every time the lock is polled
PollLockWait <sup>2</sup>	integer	30	polling interval for lockfile presence/removal (secs)
PollModemWait	integer	30	polling interval when in “modem wait” state (secs)

PollRcvdCmd	string	bin/pollrcvd	delivery script for facsimile received by polling
PostScriptTimeout <sup>1</sup>	integer	300	timeout on POSTSCRIPT interpreter runs (secs)
PriorityScheduling	boolean	<i>see below</i>	use available priority job scheduling mechanism
PS2FaxCmd <sup>1</sup>	string	bin/ps2fax	POSTSCRIPT RIP command script
QualifyCID	obsolete	–	See DynamicConfig and RejectCall for rejecting calls
QualifyPWD	string	–	file of PWD patterns for qualifying senders
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
RejectCall	boolean	false	Reject the current call
RingData	string	–	distinctive ring data call identifier
RingExtended	string	–	extended ring message identifier
RingFax	string	–	distinctive ring fax call identifier
RingsBeforeAnswer	integer	0	rings to wait before answering phone
RingTimeout	integer	6000	timeout in ms after RING before reset
RingVoice	string	–	distinctive ring voice call identifier
RTNHandlingMethod	string	Retransmit-Ignore	RTN signal handling method
SaveUnconfirmedPages	boolean	true	save or delete unconfirmed pages
SendFaxCmd <sup>1</sup>	string	bin/faxsend	fax transmit command script
SendPageCmd <sup>1</sup>	string	bin/pagesend	pager transmit command script
SendUUCPCmd <sup>1</sup>	string	bin/uucpsend	UUCP transmit command script
ServerTracing <sup>2</sup>	integer	1	non-session server tracing
SessionTracing <sup>2</sup>	integer	1	send and receive session tracing
SpeakerVolume	string	Quiet	volume level for modem speaker
TagLineCoverNumString	string	–	String substitution when not counting cover pages
TagLineFont	string	–	tag line font filename
TagLineFormat	string	<i>see below</i>	tag line format string
TIFF2FaxCmd <sup>1</sup>	string	bin/tiff2fax	TIFF converter command script
TimeOfDay <sup>1</sup>	string	Any	default time-of-day restrictions
Use2D <sup>1</sup>	boolean	Yes	restrict/permit use of 2D-encoded fax data
UseJobTSI	boolean	No	use job-specified TSI instead of LocalIdentifier
UseJobTagLine	boolean	Yes	Allow use of job-specified Tagline instead of TagLineFormat
UUCPLockMode <sup>2</sup>	octal	0600	protection mode for UUCP lock files
UUCPLockDir <sup>2</sup>	string	<i>see below</i>	UUCP lockfile directory
UUCPLockTimeout <sup>2</sup>	integer	30	time before removing stale UUCP lockfile
UUCPLockType <sup>2</sup>	string	<i>see below</i>	UUCP lockfile type
VGettyArgs	string	–	arguments passed to voice getty program
WedgedCmd <sup>1</sup>	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
ModemAnswerDialCmd	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	10	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)
ModemDoPhaseCDebug	boolean	No	query modem responses during Phase C transmit

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
ModemNoAutoAnswerCmdDelay	integer	0	time, in ms, to pause after a disabling auto-answer
ModemNoFlowCmd	string	–	command for disabling hardware flow control between DTE and
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)
ModemRaiseATCommands	boolean	Yes	raise AT commands in configuration to upper-case
ModemRate	integer	19200	baud rate to use for DCE-DTE communication
ModemReadyCmds	string	–	additional final commands when resetting modem
ModemRecvFillOrder	string	<i>see below</i>	bit order for received facsimile data
ModemRecvSuccessCmd	string	–	command to send after a successful reception
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	<i>see below</i>	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	<i>see below</i>	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
ModemSoftResetCmdDelay	integer	3000	time, in ms, to pause after a soft reset
ModemSoftRTFCC	boolean	Yes	enable software-driven real-time fax compression conversion
ModemType	string	<i>see below</i>	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
Class1Cmd	string	AT+FCLASS=1.0	Class 1.0: command to enter class 1
Class1AdaptRecvCmd	string	–	Class 1/1.0: command for adaptive reception support
Class1ColorJPEGSupport	boolean	No	Class 1/1.0: to enable color JPEG fax support
Class1EnableV34Cmd	string	–	Class 1/1.0: command to enable V.34-fax support
Class1ECMSupport	boolean	Yes	Class 1/1.0: enable T.30-A ECM support
Class1PersistentECM	boolean	Yes	Class 1/1.0: to continue to correct while in ECM
Class1ECMFrameSize	integer	256	Class 1/1.0: image frame size in ECM protocol
Class1ExtendedRes	boolean	–	Class 1/1.0: enable extended resolution support
Class1HasRHConnectBug	boolean	No	Class 1/1.0: modem can report CONNECT incorrectly
Class1HFLOCmd	string	–	Class 1/1.0: command to set hardware flow control
Class1FrameOverhead	integer	4	Class 1/1.0: extra bytes in a received HDLC frame

Class1GreyJPEGSuport	boolean	No	Class 1/1.0: to enable grey JPEG fax support
Class1HookSensitivity	integer	0	Class 1/1.0: times to ignore on-hook detection
Class1JBIGSupport	string	<i>see below</i>	Class 1/1.0: to enable monochrome JBIG fax support
Class1MRSuport	boolean	Yes	Class 1/1.0: enable 2-D MR support
Class1MMRSuport	boolean	Yes	Class 1/1.0: enable 2-D MMR support
Class1NFLOCmd	string	–	Class 1/1.0: command to set no flow control
Class1PageLengthSupport	integer	7	Class 1/1.0: coded value for modem page length support
Class1PageWidthSupport	integer	7	Class 1/1.0: coded value for modem page width support
Class1RecvAbortOK	integer	200	Class 1/1.0: max wait (ms) for “OK” after recv abort
Class1RecvIdentTimer	integer	40000	Class 1/1.0: max wait (ms) for initial ident frame
Class1RMPersistence	integer	2	Class 1/1.0: times to attempt high-speed carrier recv
Class1SFLOCmd	string	–	Class 1/1.0: command to set software flow control
Class1PPMWaitCmd	string	AT+FTS=7	Class 1/1.0: command to stop and wait before PPM
Class1ResponseWaitCmd	string	–	Class 1/1.0: command to wait before TCF response
Class1Resolutions	integer	0x7F	Class 1/1.0: bitmap of supported resolutions
Class1RMQueryCmd	string	AT+FRM=?	Class 1/1.0: command to query modem data reception rates
Class1TCFWaitCmd	string	AT+FTS=7	Class 1/1.0: command to stop and wait before TCF
Class1TMQueryCmd	string	AT+FTM=?	Class 1/1.0: command to query modem data transmission rate
Class1EOPWaitCmd	string	AT+FTS=9	Class 1/1.0: command to stop and wait before EOP
Class1ModemHasDLEBug	boolean	No	Class 1/1.0: modem fails to double DLE in V.21
Class1MsgRecvHackCmd	string	""	Class 1/1.0: command to avoid +FCERROR before image data
Class1TCFRecvHackCmd	string	""	Class 1/1.0: command to avoid +FCERROR before TCF
Class1TCFMaxNonZero	integer	10	Class 1/1.0: max% of non-zero data in good TCF
Class1TCFMinRun	integer	1000	Class 1/1.0: minimum zero run in good TCF
Class1TCFMinRunECMMod	integer	2	Class 1/1.0: modify minimum zero run in ECM by this factor
Class1TCFRecvTimeout	integer	4500	Class 1/1.0: max wait (ms) for TCF
Class1TMConnectDelay	integer	0	Class 1/1.0: delay between +FTM CONNECT and data trans
Class1SwitchingCmd	string	AT+FRS=7	Class 1/1.0: command to ensure silence after HDLC reception
Class1ValidateV21Frames	boolean	No	Class 1/1.0: check FCS against received frames
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
Class2DCCCcmd	string	AT+FDCC	Class 2: command to set modem capabilities
Class2DCCQueryCmd	string	AT+FDCC=?	Class 2: command to query modem capabilities
Class2DisableV17Cmd	string	–	Class 2: command to disable V.17 support
Class2DISCmd	string	AT+FDIS	Class 2: command to set session parameters
Class2DDISCmd	string	–	Class 2: command to set session parameters before dialing
Class2ECMType	string	“2”	Class 2: ECM specification type to follow
Class2HexNSF	boolean	Yes	Class 2: parse NSF strings as hex values
Class2HFLOCmd	string	–	Class 2: command to set hardware flow control
Class2JPEGSuport	boolean	No	Class 2: use modem JPEG support
Class2LIDCmd	string	AT+FLID	Class 2: command to set local identifier string
Class2MINSPCmd	string	AT+FMINS	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+FP	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
Class2UseLineCount	boolean	No	Class 2: use the line count from the firmware decoder
Class2UseHex	boolean	No	Class 2: parse capabilities strings as hex values
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, SEP
Class2APQueryCmd	string	AT+FAP=?	Class 2.0: query capabilities for sending and receiving SUB, SEP
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
Class2DisableV17Cmd	string	–	Class 2.0: command to disable V.17 support
Class2DISCmd	string	AT+FIS	Class 2.0: command to set session parameters
Class2ECMType	string	“2.0”	Class 2.0: ECM specification type to follow
Class2HexNSF	boolean	Yes	Class 2.0: parse NSF strings as hex values
Class2HFLOCmd	string	AT+FLO=2	Class 2.0: command to set hardware flow control
Class2JPEGSupport	boolean	No	Class 2.0: use modem JPEG support
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+FNRR=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
Class2PTSQueryCmd	string	AT+FPS?	Class 2.0: command to query 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
Class2RTFCC	boolean	No	Class 2.0: enable real-time fax compression conversion
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+FSPL	Class 2.0: command to set polling request
Class2TBCCmd	string	AT+FPP=0	Class 2.0: command to enable stream mode
Class2UseLineCount	boolean	No	Class 2.0: use the line count from the firmware decoder
Class2UseHex	boolean	No	Class 2.0: parse capabilities strings as hex values
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 **ModemAnswerCmd** 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\*Cmds**.

### AdminGroup

Tells PAM what user group is allowed to administer the fax server. Only useful if the server is compiled with PAM enabled.

### 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 <i>vgetty</i> application to answer a call
any	answer a call of any/unknown type

When a call is not answered successfully, 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<sup>2</sup>**

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.)

**AuditHook<sup>1</sup>**

Specifies a program command and an interest specification indicating which events as described in *faxwatch*(8C) should trigger the execution of the program. For example, this configuration:

```
AuditHook: "bin/hook: J*S*M*R*"
```

would cause bin/hook to be executed upon every event. The arguments passed to the hook program are respectively *<class>*, *<mask>*, *<id>*, and optionally *[info]* where...

```
<class> = JOB | SEND | MODEM | RECV
<mask>  = the specific event
<id>    = id of the job/modem with which the event is associated
[info]  = additional information
```

**BadPageHandlingMethod**

(Class 1/1.0 only) Specifies how to react to a bad page received from the remote sender: one of “RTN”, “DCN”, or “RTN-SAVE”.

If a page is received in non-ECM mode with unacceptable quality according to **PercentGoodLines** or **MaxConsecutiveBadLines** then it can be somewhat difficult to inform the sender of the problem. Historically, *HylaFAX* has assumed that signalling RTN to the sender will accomplish this. However, some senders are incapable of retransmitting pages, and to reduce burden they treat an RTN signal as a receipt confirmation and proceed to the next page without notifying the sending user of the potential problem in readability on the receive-end. (The assumption there being that the receiving user will notify the sending user if there actually is a readability problem.)

A setting of “RTN” is the historic behavior and assumes that an RTN signal will be enough to get the sender to retransmit or be otherwise informed of a potential readability problem on the receive-end. The previously-received page data is marked to be overwritten by the next page data received from the sender.

A setting of “DCN” tells *HylaFAX* to transmit a DCN signal in response to the post-page message and should trigger a call abortion by the sender. This should clearly indicate a problem in page readability to the sender, although the receipt of any following pages in a later call cannot be guaranteed.

A setting of “RTN-SAVE” more closely approximates the behavior of other fax receivers (especially fax machines). It causes *HylaFAX* to send the RTN signal but it saves the previously received page data and places the next transmitted page data in another page. This is the default setting. However, this could result in multiple copies of the same page image being saved in the same file - if the sender does indeed retransmit the unacceptable pages during the same call.

**BatchLogs<sup>1</sup>**

When sending or receiving multiple documents (denoted by EOM), this value determines if the session logs span the entire batch or, if set to no, only contain a single document.

**CallIDPattern**

A string that identifies the caller’s identity in any call identification messages provided by the modem (such as Caller\*ID or DNIS/DID). The specified string is compared against any unrecognized status messages received from the modem before **ModemAnswerCmd** is sent to the modem. If there is a match, then the remainder of the message is returned as **CallIDn** where n



corresponds to the n'th instance that this **CallIDPattern** is from the top in the modem configuration file. For example, for the ZyXEL U-1496 this parameter would be set to CallIDPattern: "CALLER NAME: " (note the trailing space). If more than one message matches between instances of "RING" messages, then the message remainders are concatenated.

The special **CallID** value of "SHIELDED\_DTMF" is used in conjunction with **ModemRingResponse** and **CallIDAnswerLength**. After executing **ModemRingResponse**, HylaFAX will expect DTMF data (usually DLE-shielded) for **CallIDn** until **CallIDAnswerLength** is reached. Note that if the modem is in voice mode to hear these DTMF digits, then it must be returned to fax mode in the **ModemAnswerCmd**. In order to prevent this feature from losing incoming fax calls, if fax sender CNG is detected while waiting for DTMF or if ten seconds elapse without sufficient DTMF then the remaining expected digits will be populated by whitespace, triggering fax answering.

For example, the following settings will cause HylaFAX to enter voice mode and go off-hook when a RING is detected. It will then expect four DTMF digits (presumably for routing), and then it will pause for 100 ms before returning to fax mode and answering the call.

```
ModemRingResponse: AT+FCLASS=8;H1
CallIDPattern:     SHIELDED_DTMF
CallIDAnswerLength: 4
ModemAnswerCmd:   <delay:100>AT+FCLASS=1;A
```

For modems that do not support AT+FCLASS=8 (such as some USB modems) this may be an appropriate equivalent:

```
ModemRingResponse: "AT#CLS=8\nAT#VLS=4\nATA"
CallIDPattern:     SHIELDED_DTMF
CallIDAnswerLength: 4
ModemAnswerCmd:   "<delay:100>AT+FCLASS=1\nATA"
```

Multiple entries of **CallIDPattern** are used together in one modem configuration file in order to capture multiple **CallID** responses. For example, the following settings would capture "NDID" responses from the modem as CallID1, "NMBR" responses from the modem as CallID2, "NAME" responses as CallID3, and it would trigger **ModemAnswerCmd** whenever CallID1 or CallID2 were longer than 7 or 10 characters, respectively.

```
CallIDPattern:     "NDID="
CallIDAnswerLength: 7
CallIDPattern:     "NMBR="
CallIDAnswerLength: 10
CallIDPattern:     "NAME="
```

Note that this example is only given as an example, and probably would cause unexpected results. Because pattern-matching ends when **ModemAnswerCmd** is sent to the modem, if "NDID", "NMBR", and "NAME" responses came from the modem in that order, and the "NDID" response was at least 7 characters long, then **ModemAnswerCmd** would be sent to the modem immediately following the "NDID" response, and the "NMBR" and "NAME" responses would be ignored. Generally only one **CallIDAnswerLength** item should be in a modem configuration file.

### **CallIDAnswerLength**

An integer indicating the minimum number of characters in a **CallID** matching the previous **CallIDPattern** to be received when **ModemAnswerCmd** is sent to the modem irrespective of any **RingsBeforeAnswer** value greater than zero. This allows the answering of calls which deliver

CID/DID data but not RINGs. For example, `CallIDAnswerLength: "7"` would cause the call to be answered when **CallID2** reached a length of seven digits. A value of zero for **CallIDAnswerLength** disables this feature.

### **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<sup>1</sup>**

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

### **ContCoverPage<sup>1</sup>**

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<sup>2</sup>**

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.)

### **CountSkippedPages**

If a page range has been specified in the job, and pages are being skipped during the send, this controls if the pages skipped are counted in the pages counted and totaled. If skipped pages are to be counted, it will appear as if the pages were instantly sent in 0:00 with no data. The tagline counts will be incremented on skipped pages, and faxq will notify clients via triggers of them as well. If they are not to be counted, then tagline page counts will act as if the pages never existed, and not faxq trigger notifications will be sent.

### **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<sup>2</sup>**

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**.

### **DistinctiveRings**

Modern distinctive ring support on most modems indicates the ring cadence rather than the older style of "RING1", "RING2", etc. To indicate the ring cadence, **DRingOn** and **DRingOff** values are presented by the modem to the *faxgetty* process. The modem indicates the entire cadence between "RING" indications Like this:

```
RING
DROF=40
DRON=8
DROF=4
DRON=8
RING
```

The corresponding **DistinctiveRings** parameter for this ring cadence would be:

DistinctiveRings: F-8-4-8

where “F” tells the *faxgetty* process that the ring cadence is for a facsimile (“V” for voice and “D” for data), and the other values describe the ring cadence with leading and ending **DRingOff** values ignored. Multiple ring cadences are indicated by delimiting them with commas in this fashion:

DistinctiveRings: V-20,F-8-4-8,D-4-2-4-8

### **DRingOff**

A string that identifies the “off” value in any distinctive ring cadence, for example “DROF=”. See also **DRingOn** and **DistinctiveRings**.

### **DRingOn**

A string that identifies the “on” value in any distinctive ring cadence, for example “DRON=”. See also **DRingOff** and **DistinctiveRings**.

### **DynamicConfig**

The pathname of the optional program, e.g. “etc/localid”, that makes dynamic configuration changes, i.e., to **LocalIdentifier**, based on device ID and call identification. The program is passed those values as the parameters (\$1 = device id, \$2 = CallID1, \$3 = CallID2, \$4 = CallID3, ...), when answering an incoming call. The program can then perform local processing as desired to send on standard output the configuration items to change in a “parameter:value” format, i.e. “LocalIdentifier: +1.800.555.1212”. If there are multiple parameters to be changed, then each item must be on its own line. This is commonly used to dynamically alter the local identification of systems which use DID/DNIS, but it can also be used to allow different modem configurations for different senders. This program can also set the **RejectCall** options to cause the current call to be rejected instead of answered. Note that this file must be marked as executable by the *faxgetty* process.

### **EGettyArgs**

A string that indicates whether or not the server should use an external getty application to deduce and possibly handle an incoming call. 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 “%1” is replaced with the tty name and any appearance of “%s” is replaced with the serial port speed (in decimal). Any appearance of escaped numbers 1 through 9 (“%1” through “%9”) are replaced by the match to the corresponding **CallIDPattern**, if present. The “%” character can be specified with “%%”. If the **EGettyArgs** parameter is not specified in the configuration file or if the string value is null, then “extern” connections will be rejected. Note that in addition to the specified command line arguments, the external getty program is invoked with its standard input, output, and error descriptors directed to the controlling tty device.

When the external getty application completes, its exit status is evaluated and is interpreted to indicate what, if anything, should be done with the call. An exit status of “0” indicates an unknown call type and that the call should be handled as if the external getty program had not been used. An exit status of “1” indicates a data connection and that the *getty(8C)* program should be used to handle the call (see **GettyArgs**) after being answered with **ModemAnswerDataBeginCmd**. An exit status of “2” indicates a fax connection that should be handled after being answered with **ModemAnswerFaxBeginCmd**. An exit status of “3” indicates a voice call and that the *vgetty* program should be used to handle the call (see **VGettyArgs**) after being answered with **BR ModemAnswerVoiceBeginCmd**. An exit status of “4” is considered to be an error condition. The session will be terminated. An exit status of “5” is used to indicate that the external getty program handled the call entirely, is not an error condition, and that the session is to be considered

terminated.

**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.

**Include<sup>2</sup>**

Include the specified file and parse it as a config file

**InternationalPrefix<sup>2</sup>**

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.

**JobControlCmd<sup>1</sup>**

The command to invoke to which provides Job Control information for a job. See *JobControl*(8C).

**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<sup>2</sup>**

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

**JobReqProto**

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

**JobRetryBusy**

The number of times to redial a phone number after receiving a “BUSY” result code when the number has not been successfully dialed before.

**JobRetryDataConn**

The number of times to redial a phone number after the dialing attempt failed because a data modem answered the phone when the number has not been successfully dialed before.

**JobRetryNoAnswer**

The number of times to redial a phone number after the dialing attempt failed because of NO ANSWER when the number has not been successfully dialed before.

**JobRetryNoCarrier**

The number of times to redial a phone number after the dialing attempt failed because of NO CARRIER when the number has not been successfully dialed before. 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.

**JobRetryOther**

The number of times to redial a phone number after the dialing attempt failed for a reason not already covered by one of the other JobRetry\* parameters when the number has not been successfully dialed before.

**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). Any appearance of escaped numbers 1 through 9 (“%1” through “%9”) are replaced by the match to the corresponding **CallIDPattern**, if present. 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<sup>2</sup>**

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.

**LogCalls**

Log all incoming calls into *xferfaxlog* as CALL records. This allows a quick check of all incoming calls, regardless of fax completion or not in a unified manner with other fax logs.

**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<sup>2</sup>**

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

**MaxConcurrentCalls<sup>1</sup>**

The default value to use for the maximum number of jobs for the same destination that are processed concurrently. 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 **MaxConcurrentCalls** is probably best left as the default of 1 and exceptions should be accommodated by the **JobControl** mechanism.

**MaxBatchJobs<sup>1</sup>**

The maximum number of jobs to batch together in a single call.

**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. Facsimile received with error-correction do not have copy quality checking performed. See also **PercentGoodLines**.

**MaxDials<sup>1</sup>**

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 **JobControl** mechanism. See also **MaxTries**.

**MaxRecvPages**

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

**MaxSendPages<sup>1</sup>**

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 **JobControl** mechanism.

**MaxSetupAttempts**

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

**MaxTries<sup>1</sup>**

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 **JobControl** mechanism. See also **MaxDials**.

**MinAcceptedLineCount**

The minimum number of received scanlines allowed on an “OK” page. The use of this setting prevents confirmation from being sent for very short pages.

**ModemGroup<sup>1</sup>**

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.

### **NoAnswerVoice**

Whether or not to disable the answering of calls indicated as voice (e.g. by **DistinctiveRings**).

### **NoCarrierRetrys**

This is mapped onto JobRetryNoCarrier

### **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.

### **ModemRingResponse**

This can be used to cause the command defined by **ModemRingResponse** to be sent to the modem after hearing at least **ModemRingsBeforeResponse** RINGS in order to receive DID-DTMF data, for example.

### **ModemRingsBeforeResponse**

This defines the minimum number of RINGS that must be heard before **ModemRingResponse** is first sent to the modem.

### **NotifyCmd<sup>2</sup>**

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

### **PageChop<sup>1</sup>**

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<sup>1</sup>**

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<sup>1</sup>**

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

### **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. Facsimile received with error-correction do not have copy quality checking performed. See also **MaxConsecutiveBadLines**.

### **PollLockPokeModem**

When polling for the presence/removal of a UUCP lockfile, also test to make sure the modem is still reachable. This is useful for scenarios where the modem is apt to become disconnected while the modem is idle, such as an external modem that may lose power, where the administrator wants notification of this (as a "wedged" condition).

**PollLockWait**

When polling for the presence/removal of a UUCP lockfile, the time (in seconds) to wait between checks. Lockfile polling occurs to ensure the fax server doesn't collide with another process using the modem.

**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, and HP-UX systems where the configure script detects one of the functions `schedctl()`, `priocntl()` or `rtprio()` the default is true. The only other supported priority mechanism is POSIX `sched_setscheduler()`, which is used on Linux, OpenServer and possibly other systems. On such systems, as well as on systems where no priority scheduling mechanism exists, the default is false."

**PS2FaxCmd<sup>1</sup>**

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

**QualifyPWD**

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 **QualifyPWD** is non-null, then only messages from facsimile machines identified in the file specified by the string (typically `etc/passwd`) will be accepted; similar to **QualifyTSI**.

**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 performed on the host. (When copy quality checking is not done by the server, for example when error correction protocol is used, the received facsimile data is written exactly as it is received from the modem.) The format may be one: "1-D MH", "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)*.

**RejectCall**

If this is set to true, the current call will not be answered, and will be rejected. This option has no effect if set in the configuration file, but when set by the **DynamicConfig** program, controls the current call. This options supersedes the obsolete **QualifyCID** option.



**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**.

**RingExtended**

An identifier which matches the initial portion of an extended RING message sent by the modem to relay CID and/or DNIS data instead of sending it formatted on separate lines.

**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)*.

**RingTimeout**

If a ring is heard and the call is not answered, the time in ms to wait before reinitializing the modem for the next call.

**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 an RTN signal received from the remote: one of “Retransmit”, “Giveup”, “Ignore”, or “Retransmit-Ignore”.

“Retransmit” assumes that the page is not received successfully. HylaFAX will make up to two additional attempts to send the page, decreasing signalling rate and retraining. If RTN continues, up to 2 additional calls will be placed. The downside is that if the remote always responds with RTN, the page will be sent 9 times and no following pages will be sent. Many fax machines will not behave this way, although T.30 specification seems clear that this is the intent of the RTN signal.

Many fax machines will interpret RTN as meaning to not send the same data again. That is because RTN may indicate problems with flow control, incorrectly encoded T.4 data, or some incompatibility other than line noise. Fax machines that interpret RTN this way will disconnect and require a manual retransmission. This “over and out” behaviour can be activated by the “Giveup” value. The advantage to this behavior is that the same page of image data will not be sent multiple times, but the downside is that the following pages will not be sent.

“Ignore” is similar to “Giveup” in that it makes the assumption the data cannot be automatically accepted by the receiver. However, rather than disconnecting the page of image data is abandoned and processing moves on to the next. The remote is left to decide what to do with the unacceptable page of image data. The downside to this behavior, is that the remote may or may not have kept the page, depending on its interpretation of the RTN signal - but it does allow us to continue on to the next page.

“Retransmit-Ignore” is a combination of “Retransmit” and “Ignore”. The page of image data is retransmitted up to two additional times, but rather than disconnecting after a third RTN signal for the same page, processing then continues on to the next page. This approach is an effort to satisfy both interpretations of an RTN signal. It allows the receiver to hopefully receive a better copy of the image data while not failing to send subsequent pages. If the receiver saves or prints a copy of pages for which it transmits RTN, then it could save or print up to three copies of every page.

**SaveUnconfirmedPages**

Whether or not to save a received facsimile image page if the sender disconnects without sending the post-page message, without hearing our message confirmation, without retraining as requested, or otherwise breaks fax protocol following such pages for which the sender should not have a “receipt confirmation”.

**SendFaxCmd<sup>1</sup>**

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

**SendPageCmd<sup>1</sup>**

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

**SendUUCPCmd<sup>1</sup>**

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

**ServerTracing<sup>2</sup>**

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	commands 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
262144 (0x40000)	TIFF library	any messages produced by the TIFF library
524288 (0x80000)	ECM Frames	binary T.30-A HDLC ECM frames

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<sup>2</sup>**

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:

<b>Value</b>	<b>Description</b>
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**.

**TagLineCoverNumString**

Setting **TagLineCoverNumString** instructs faxsend to not include the cover pages in the normal page numbering mechanism. The page numbers used in the tagline generation will adjusted to not count the cover pages, and the **TagLineCoverNumString** value will be used instead of a page number on the tagline of the cover page.

**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:

<b>Escape</b>	<b>Description</b>
%%a	destination subaddress
%%c	destination company name
%%C	sender's company name
%%d	destination phone number
%%g	destination geographic location
%%G	sender's geographic location
%%i	job identifier
%%I	group identifier
%%j	user-specified job tag
%%l	<b>LocalIdentifier</b> or canonicalized <b>FAXNumber</b>
%%m	sender's electronic mail address
%%n	canonicalized <b>FAXNumber</b>
%%p	current page number of session
%%P	current page number of job
%%r	receiver's name
%%s	sender's name
%%S	subject
%%t	total pages in session
%%T	total pages in job
%%v	destination voice number
%%V	sender's voice number
%%%	“%”

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<sup>1</sup>**

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

#### **TimeOfDay<sup>1</sup>**

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 **JobControl** mechanism.

**Use2D<sup>1</sup>** Control the use of 2D-encoded data for transmitted facsimile.

#### **UseJobTagLine**

Allow the use of the job-specified tagline instead of the configured **TaglineFormat**

#### **UUCPLockMode<sup>2</sup>**

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

#### **UUCPLockDir<sup>2</sup>**

The pathname of the directory in which UUCP lockfiles are to be created.

#### **UUCPLockTimeout<sup>2</sup>**

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<sup>2</sup>**

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..ttaa01”. 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). Any appearance of escaped numbers 1 through 9 (“%1” through “%9”) are replaced by the match to the corresponding **CallIDPattern**, if present. 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<sup>1</sup>

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)
<play:filename>	play the specified audio file (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
VCON	voice connection indicator
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).

The “<play:filename>” escape can be used to play a raw audio file with a voice modem. This feature can be used, for example, to play a brief audio message after picking up the line but before answering. A configuration example may be:

```
ModemRingResponse: "AT+FCLASS=8;H1\nAT+VSM=131\nAT+VLS=1\nAT+VTX\n<waitfor:CONNECT><play:/sound
ModemAnswerCmd: "<delay:100>AT+FCLASS=1;A"
CallIDPattern: SHIELDED_DTMF
CallIDAnswerLength: 4
```

In this example using an IS-101 voice-compliant modem, a RING indication from the modem will cause the modem to be placed in voice mode, set ulaw audio compression, and via the connected phone line play back the etc/play1.raw audio file, which may say, "After the tone enter a four-digit extension, then start the fax." Following the message a tone is played.

#### 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**.

#### **ModemAnswerDialCmd**

The command to explicitly answer the phone for a dialed 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.

#### **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 is only needed for hosts and modems (such as USRs) where 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 through 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 **ModemWait-TimeCmd** parameter. This additional server-based timeout is provided to guard against modems that can “lock up” when dialing the telephone.

**ModemDoPhaseCDebug**

Whether or not to query the modem for responses during image data transmission. Normally the modem should not produce any responses during Phase C data transmission. However, in some debugging scenarios (i.e. some Class 2.1 modems may show debugging information) it may be appropriate to query the modem for responses during the data transmission.

**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 “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.

**ModemNoAutoAnswerCmdDelay**

The time, in milliseconds, to pause after receiving the OK following **ModemNoAutoAnswerCmd** before any further commands are sent to the modem. All input from the modem is flushed after pausing.

**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).

**ModemRaiseATCommands**

Indicates whether or not all configuration AT commands for the modem should be raised to upper-case automatically. The standard is for all upper-case to be used in AT commands, and indeed, some rare modems have been known to malfunction with lower-case commands. However, sometimes an administrator may want to use lower-case commands in a rare condition such as to work around modem blacklisting of dialed numbers.

**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.

**ModemReadyCmds**

A string of commands to issue to the modem during reception initialization. This string is sent to the modem *after* the standard set of configuration commands required by the fax server. This is done, for example, to un-busy a DID line so that calls can come through.

**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.

#### **ModemRecvSuccessCmd**

A string of commands to issue to the modem after a successful receive session before the call is disconnected.

#### **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.

**ModemSoftResetCmdDelay**

The time, in milliseconds, to pause after receiving the OK following **ModemSoftResetCmd** before any further commands are sent to the modem.

**ModemSoftRTFCC**

Whether or not to enable software-driven Real-Time Fax Compression Conversion. RTFCC allows HylaFAX to convert the image compression between MH MR and MMR formats regardless of how faxq formatted the image file. Note that when using RTFCC, the compression format of the file will be ignored, thus the “-1”, “-2”, and “-3” options for sendfax, ps2fax, and others will only influence how the document is prepared by faxq and will not influence the actual negotiated session parameters. **Class2RTFCC** takes precedence over **ModemSoftRTFCC** and if both are enabled, then software-driven RTFCC will not be performed in favor of the firmware-driven RTFCC.

**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.

**Class1AdaptRecvCmd**

The command used to enable adaptive reception support (usually “AT+FAR=1”). This feature is new in T.31, and many modems will not support it. This feature may reduce the number of reception failures due to errors cascading from +FCERROR messages.

**Class1ColorJPEGSupport**

Whether or not to enable support for T.30-E full-color facsimile with JPEG compression. Enabling this automatically enables **Class1GreyJPEGSupport**.

**Class1EnableV34Cmd**

The command to enable V.34-fax support with at least the desired maximum primary channel rate.

**Class1ECMSupport**

Whether or not to support T.30-A error correction protocol. Use of ECM will require 64 kilobytes of free memory per modem in active use.

**Class1PersistentECM**

Whether or not to continue to retransmit and allow to continue to receive image data in ECM protocol which is not accepted as valid after four successive attempts.

**Class1ECMFrameSize**

The size in bytes of image frames to transmit during ECM protocol. This setting will also indicate a preference in receive sessions. The only acceptable values are 64 and 256. A setting of 64 may be useful on high-load systems and possibly environments with extremely poor line quality.

**Class1ExtendedRes**

Whether or not to support resolutions other than normal and fine. This option has been deprecated by **Class1Resolutions**.

**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).

**Class1GreyJPEGSupport**

Whether or not to enable support for T.30-E greyscale facsimile with JPEG compression. This is always enabled if **Class1ColorJPEGSupport** is enabled.

**Class1HookSensitivity**

The number of times to ignore on-hook detections and merely treat them as command or modem errors.

**Class1JBIGSupport**

Whether or not to enable support for T.85 monochrome facsimile with JBIG compression. Options are “true” for support in both sending and receiving, “false” for no support, “send” for support only in sending, and “receive” for support only in receiving. If, during the build process a compatible JBIG library was found then send support is enabled by default. If, during the build process the TIFF tools are found to support JBIG then receive support is enabled by default.

**Class1HasRHConnectBug**

A Class 1 modem should only report CONNECT after AT+FRH=3 when V.21 HDLC data is detected. However, some modems will incorrectly report CONNECT after AT+FRH=3 whenever any carrier is present. In such cases **Class1HasRHConnectBug** should be set to “true”.

**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.

**Class1MRSupport**

Whether or not to enable support for two-dimensional Modified Read (MR) image data format compression.

**Class1MMRSupport**

Whether or not to enable support for two-dimensional Modified Modified Read (MMR) image data format compression. Note that MMR support requires also ECM support to be enabled.

**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.

**Class1PageLengthSupport**

The coded value for page lengths supported by the modem. The only correct values and meanings are these: "1", A4 page length; "3", both A4 and B4 page length; "7", unlimited page length.

**Class1PageWidthSupport**

The coded value for page widths supported by the modem. The only correct values and meanings are these: "1", A4 page width; "3", both A4 and B4 page width; "7", all of A4, B4, and A3 page widths.

**Class1PPMWaitCmd**

The command used to stop and wait before sending the post page message, except before sending EOP, when **Class1EOPWaitCmd** is used instead. We must ensure that the high-speed carrier has stopped completely. According to T.30, Chapter 5, Note 4, this delay should be 75 +/- 20 ms.

**Class1ResponseWaitCmd**

The command used to stop and wait after sending TCF, before attempting to receive a training response from the remote. Set this to "AT+FRS=1" if you experience "DIS/DTC received 3 times" errors.

**Class1Resolutions**

A bitmapped (bit-or'd) value indicating the resolutions to be supported during facsimile operation. Individual resolutions follow Table 21/T.32 and are defined as follows:

Bit	Description
0 (0x00)	R8 x 3.85 l/mm, Normal
1 (0x01)	R8 x 7.7 l/mm, Fine
2 (0x02)	R8 x 15.4 l/mm, Superfine
4 (0x04)	R16 x 15.4 l/mm, Hyperfine
8 (0x08)	200 dpi x 100 l/inch
16 (0x10)	200 dpi x 200 l/inch
32 (0x20)	200 dpi x 400 l/inch
64 (0x40)	300 dpi x 300 l/inch

Thus, a value of 3 would indicate support for normal, fine, and superfine resolutions.

**Class1RMQueryCmd**

The command to send to the modem to get the list of supported reception bit-rates. If the parameter begins with a "!", then the remainder of the string is taken to be the modem response and no command is issued to the modem.

**Class1TCFWaitCmd**

The command used to stop and wait before sending TCF, similar to **Class1PPMWaitCmd**. According to T.30, Chapter 5, Note 3, this delay should be 75 +/- 20 ms.

**Class1TMQueryCmd**

The command to send to the modem to get the list of supported transmission bit-rates. If the parameter begins with a "!", then the remainder of the string is taken to be the modem response and no command is issued to the modem.

**Class1EOPWaitCmd**

The command used to stop and wait before sending the post page message similar to **Class1PPMWaitCmd**. We allow a different setting in the case of EOP, however, because

empirically some machines may need more time.

**Class1ModemHasDLEBug**

Used to indicate that the modem does not correctly duplicate DLE characters in the V.21 communication to the DTE.

**Class1MsgRecvHackCmd**

If receive failures occur due to +FCERROR just prior to image data reception, setting **Class1MsgRecvHackCmd** to AT+FRS=1 may help.

**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.

**Class1RMPersistence**

The number of times that an attempt to receive the high-speed data carrier should be made, resulting in +FCERROR, before the low-speed message carrier reception is attempted. Some modems are quick to (perhaps incorrectly) return +FCERROR, and for those modems a value of 2 or 3 should be used. For modems that are not quick to return +FCERROR, a value of 1 should be used. Proper tuning of this can provide a type of "adaptive reception control" for modems that accurately return +FCERROR without supporting **Class1AdaptRecvCmd**. For those modems that support **Class1AdaptRecvCmd**, **Class1RMPersistence** should probably not be set at 1, although +FCERROR should almost never occur with such modems.

If the modem does not support the reporting of +FCERROR or adaptive reception control, then **Class1RMPersistence** should probably be set at 0, which causes the timeout looking for the high-speed data carrier to be shortened, thus increasing the likelihood of recovery from any dissynchronization. When **Class1RMPersistence** is other than 0 it is assumed that the +FCERROR reporting or **Class1AdaptRecvCmd** is functional and therefore the timeout looking for the high-speed data carrier is lengthened.

**Class1SwitchingCmd**

The command used to ensure that the sending facsimile device has turned off its modulator (i.e. loss-of-carrier) as recommended by T.31: Appendix II.1.

**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** and **Class1TCFMinRunECMMod**.

**Class1TCFMinRunECMMod**

The factor by which **Class1TCFMinRun** should be modified in the case of an ECM session. As ECM protocol allows retransmissions it is commonly faster to accept a lesser-quality data stream

and the subsequent retransmissions than it is to allow the communication speed to slow down (where the demodulation may still not produce an ideal data stream, anyway).

#### **Class1TCFRecvHackCmd**

If receive failures occur due to +FCERROR just prior to TCF data reception, setting **Class1TCFRecvHackCmd** to AT+FRS=1 may help.

#### **Class1TCFRecvTimeout**

The maximum time to wait, in milliseconds, for the first byte and again for the entirety of the Training Check (TCF) message data that is received during the training phase of the facsimile reception protocol.

#### **Class1TMConnectDelay**

The time, in milliseconds, to delay after receiving CONNECT following +FTM before sending image data. T.31 8.3.3 requires the modem to respond with CONNECT before the modulation training pattern. If transmission begins before the remote has successfully completed its own modulation training pattern then data, especially during TCF, could be lost. Many modems do not follow T.31 in this regard, and thus the default is zero; however its use with such modems would likely have a negligible effect.

#### **Class1ValidateV21Frames**

Whether or not to use the FCS bits of received V.21 HDLC frames to check the validity of the frame itself. Most Class 1 modems perform this check independently (per T.31 7.4) and do not require this feature to be enabled.

### **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>".

**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”. **Class2CQCmd** should be configured to set-up all available copy quality services available per **Class2CQQueryCmd**. To disable features that are available, configure **Class2CQQueryCmd** with a “!”, and then set **Class2CQCmd** accordingly.

**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. If copy quality checking is configured to be done by the modem then it is not done by the server.

**Class2CRCmd**

The command to use to enable the reception of facsimile.

**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”.

**Class2ECMType**

The interpretation of the EC parameter in the modem DCC response varies between the Class 2, “2”, and Class 2.0, “2.0”, specifications. This configuration parameter allows the administrator to specify which type to use. The corresponding specification type is used by default.

**Class2HexNSF**

Whether or not to parse the NSF strings reported by the modem using hexadecimal values. By default, they are parsed as hexadecimal values.

**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”.

**Class2JPEGSupport**

Whether or not to enable and utilize the JPEG support found in the modem.

**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.

**Class2PTSQueryCmd**

The command to use to query the received page status code. This command may be used to determine the post-page response returned from the receiver. Beware that some modems do not properly implement this command in which case this setting should be set to “none”.

**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.

**Class2RTFCC**

Whether or not to enable MultiTech’s Real-Time Fax Compression Conversion which is available in later firmware revisions for the MT5634ZBA-V92, MT5634ZPX-PCI-V92, and other models. RTFCC allows HylaFAX to convert the image compression between MH MR and MMR formats regardless of how faxq formatted the image file. If RTFCC is available with your firmware, then the response to “AT+FFC=?” is non-zero. Note that when using RTFCC, the compression format of the file will be ignored, thus the “-1”, “-2”, and “-3” options for sendfax, ps2fax, and others will not influence the actual negotiated session parameters.

**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**.

**Class2UseLineCount**

Whether or not to use the line count reported to HylaFAX by the modem firmware decoder.

**Class2UseHex**

Whether or not to parse the capabilities strings reported by the modem using hexadecimal values. By default, they are parsed as decimal values.

**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) integer 30 IXO: 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 command “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)*