HylaFAX Home Page

Latest Release: *None*
Latest Software: v3.0beta112
Master FTP Site: sgi.com (192.48.153.1), directory sgi/fax
Home Page: http://www.vix.com/hylafax/
FAQ Page: http://www.vix.com/hylafax/FAQ/

This is the home page for the HylaFAX facsimile software system.

- Check here for news about the name change and major new distribution.
- *If you want to build the software and are impatient*, try this synopsis.
- If you have a question, consult the HylaFAQ or search the mailing list archives.
- If you are looking for a source or binary distribution, read these instructions.
- If you want to find out what HylaFAX is all about, begin with the overview.
- If you want a list of all online documentation, check out the table of contents.

If this is your first encounter with this software you should start with the overview section and follow the button at the bottom of each page to get a top to bottom reading of the most important information. You will also find that each part of this document includes a button that can be used to take you to the table of contents for the HylaFAX documentation. Note that a number of pages are still under construction; these are expected to change soon.

HylaFAX overview (*first-time users start here*).
HylaFAX table of contents.

HylaFAX Table of Contents

This is the table of contents for the documentation included with the HylaFAX software distribution. To get a single document with all the on-line HylaFAX documentation except for the supporting materials, click on the ***Collect Pages*** button below.

Create a single document (suitable for printing or saving).

Alternatively, a PostScript copy of a snapshot of this documentation is available by public ftp in the directory ftp://ftp.sgi.com/sgi/fax/doc.

If you are looking for ancillary materials such as the FAQ or mailing list archives, consult the section on supporting materials

BASIC INFORMATION

- HylaFAX overview
- How to tell which version of HylaFAX you have
- Which modems can be used with HylaFAX
- About the Class 1 modem support
- About the Class 2 modem support
- About the Class 2.0 modem support
- About the binary distribution
- About the source distribution

BUILDING AND SETUP INFORMATION

- How to build the software from source code
- How to build Ghostscript with the TIFF driver
- How to install from the binary distribution
- How to setup a server: basic configuration
- How to setup a server: advanced configuration
- How to setup clients
- About server operation
- Troubleshooting problems
- How to setup a mail to fax gateway

SUPPORTING MATERIALS

- Where to get the HylaFAX distribution
- HylaFAX Frequently Asked Questions (HylaFAQ)
- HylaFAX mailing list archives
- HylaFAX user survey form

- FlexFAX Modem Bakeoff report
- FlexFAX/HylaFAX modem information
- ZyXEL modem voice support information
- Supra/Rockwell Class 1 documentation
- Supra/Rockwell Class 2 documentation
- Telebit T3000/WorldBlazer FAX commands
- Hayes technical reference for modem users
- Peter Chen's "What you need to know about modems"
- FAX modem command scorecard

END MATTER

- About contributed software
- Acknowledgements
- Other documentation
- HylaFAX mailing list and other final words
- Use and copyright
- InterViews information
- TIFF information

■ HylaFAX overview.

HylaFAX Overview

HylaFAX is a facsimile system for UNIX systems. It supports:

- sending facsimile
- receiving facsimile
- polled retrieval of facsimile
- transparent shared data use of the modem

Facsimile can be any size (e.g. A4, B4), either 98 or 196 lpi, and transmitted/received as either 1D-encoded or 2D-encoded facsimile data (2D-encoded data is frequently more compact and hence takes a shorter time to communicate). Any modem that supports one of the standard interfaces for facsimile operation can be used; i.e. any Class 1, Class 2, or Class 2.0 modem.

Outgoing documents can be any format; the sendfax program uses a rule-based definition file similar to the System V **/etc/magic** file to deduce document types and to decide how to convert each document to a form suitable for transmission (either PostScript or TIFF/F). Automatic cover page generation is supported and users can easily tailor cover pages to their environment. A simple text-based phonebook database is supported by *sendfax*. Information is also provided on how to trivially setup an email to fax gateway service.

Incoming facsimile are stored in a receiving area as TIFF/F (read ‘‘*TIFF Class F*’’) files and may be automatically delivered by mail and/or printed. A fax server status program, faxstat, can be used to monitor the send and receive queues, as well as the state of facsimile servers.

Fax modems may be shared with outgoing data communication applications that honor the "UUCP locking protocol". These applications typically include: *cu*, *tip*, *kermit*, *uucp*, *slip*, and *ppp*. The system can also be configured so that incoming data calls cause the system to invoke the standard system getty program.

The software is structured around a client-server architecture. Fax modems may reside on a single machine on a network and clients can submit an outbound job from any machine that can communicate with the machine on which the modems reside. Client software is designed to be lightweight and easy to port; imaging can be offloaded to the server or done on the client. (Imaging is, however, typically done on the server because it simplifies administration.) An access control mechanism is included to control which users on which machines may submit jobs.

Multiple modems on a single machine are effectively scheduled for high throughput. Broadcast faxing is well-supported through optimal imaging of transmitted documents and the effective scheduling of modem resources. Support is provided for scheduling jobs during off-peak hours based on the destination phone numbers (e.g. long distance calls may be scheduled for off-peak phone rates). An access control mechanism can be used to restrict the class of phone numbers called so that, for example, calls to emergency services such as 911 can be rejected out of hand.

The server requires a PostScript to facsimile imaging utility for useful operation (otherwise, only pre-imaged facsimile may be transmitted.) A Display PostScript-based imager is provided for IRIX 4.x- and 5.x-based systems. For other systems, a Ghostscript-based version can be built from the GNU sources.

HylaFAX is freely available under copyright in complete source form. It may be used in commercial applications in part or in whole without charge.

Which modems can be used with HylaFAX.
HylaFAX table of contents.

HOW TO TELL WHICH HYLAFAX VERSION YOU HAVE

If you have need to refer to this specific software, you should identify it as:

```
HylaFAX v<version><n>
```

where <*version*> is found in the file **VERSION** and <*n*> is the number recorded in **dist/hylfax.alpha**. This string is also prominently displayed when you run the configure script to setup the software for compilation and each time the faxq scheduler process is started (look in the file where syslog messages are recorded).

If you have a binary distribution for a Silicon Graphics machine, <*version*> and <*n*> are displayed in `"versions -n hylafax.sw"`; for example:

```
I = Installed, R = Removed

   Name                  Version     Description

I  hylafax               1006004075  HylaFAX Facsimile Software, Version 3.0beta
I  hylafax.sw            1006004075  HylaFAX Software
I  hylafax.sw.client     1006004075  HylaFAX Client Software
I  hylafax.sw.server     1006004075  HylaFAX Server Software
```

Here <*version*> = 3.0beta and <*n*> = 075 (the right three digits in the Version column), so this is HylaFAX v3.0beta075.

HylaFAX table of contents.

WHICH MODEMS CAN BE USED WITH HYLAFAX

HylaFAX is intended to be used with fax modems. Fax modems are not the same as data modems though most contemporary data modems also include support for fax communication. HylaFAX should work with any Class 1, Class 2, or Class 2.0 fax modem. Wherever possible HylaFAX works around known modem problems or restricts modem usage in order to provide a functioning system.

The following is a list of modems that have been used with HylaFAX. There are likely others modems that have been tried and/or are being used with HylaFAX; this list shows only those that have some form of support in the software or that have been publicly reported as *functional* (though beware that what is functional to one person may not be functional to you). Consult the Modems section of this documentation for known modem bugs and gotchas or follow the links in the table to find information for a particular modem.

Manufacturer	**Model**	**Notes/Class**	
AT&T Paradyne	DataPort 14.4	$	1,2
Boca Research	M1440E		1,2
Boca Research	M1440E/RC32ACL	@	1,2
CPI	ViVa 14.4/FAX		2
Digicom	Scout+	+	1
Dynalink	Dynalink 1414VE		2
E-Tech, Inc.	P1496MX 5.06-SWE		2
Everex	EverFax 24/96D, 24/96E		2
Hayes	Optima 144, Optima 28800		1
Hayes	Optima 2400+Fax96		2
Intel	SatisFAXtion 400e		1
Logicode	Quicktel Xeba 14.4		2
Multi-Tech	MT1432BA, MT224BA, MT2834BA	+,*	2
Multi-Tech	MT1432BG	+	2
Multi-Tech	MT1932ZDX, MT2834ZDX	+	2
Nuvo	Voyager 96424PFX		1
Olitec France	Olicom Poche Fax Modem 2400		2
PPI	PM14400FXMT, PM14400FXSA		2
Supra	SupraFAX v.32bis		1,2
Supra	SupraFAX v.32bis/RC32ACL	@	1,2
Telebit	T3000, WorldBlazer, QBlazer		2
Tricom	Tornado28/42		1
Twincom	144/DF		1,2
UDS	FasTalk Fax32		1
USRobotics	Courier		1,2.0
USRobotics	Sportster		1,2.0
Yocom	1414E		2
Zoom	VFX		1,2
Zero One Net.	ZyXEL U1496	+	2,2.0

Legend:

* There are apparently many variants of the MT1432BA, the following models are known to work: MT1432BA, MT1432BA/A, MT1432MK, MT1432PCS.

@ /RC32ACL refers to second-generation products made with the Rockwell RC32ACL part and different firmware.

+ *Modem is recommended for use with this software.*
$ *Modem is recommended for use with this software* ***only when used in Class 1***.
1 Modem supports the TIA/EIA-578 "Class 1" specification (see Class 1 Support for **important information on system requirements for this support**).
2 Modem supports the TIA/EIA-592 draft SP-2388-A of August 30, 1991 (see Class 2 Support for caveats).
2.0 Modem supports the TIA/EIA-592 "Class 2.0" specification (see Class 2.0 Support).

Source distributions.
Binary distributions.
HylaFAX table of contents.

CLASS 1 MODEM SUPPORT (CAVEAT EMPTOR)

HylaFAX includes a driver for modems that support the EIA-578 "Class 1" programming interface. These modems export a very low level interface that requires that the CCITT Recommendation T.30 facsimile communication protocol be implemented almost entirely in the host. Robust support for this protocol places two requirements on the host system:

- low latency for serial line input
- near realtime response

In a UNIX environment both these requirements can be problematic. In particular, many UNIX systems increase the latency for data received on a serial port in order to reduce system overhead. That is, input data are typically held in a low level device driver for some period of time before they are passed to the user so that bursts of input data can be delivered to the user together. This behaviour is known to occur under the Silicon Graphics IRIX operating system; it may also occur under other systems. It is important for the proper operation of the Class 1 driver that input data be delivered to the facsimile server as quickly as possible. This may require making a non-standard call of some sort to the operating system. For IRIX systems this call is automatically done.

The response time requirements are important for insuring that T.30 protocol messages are received in a timely fashion. On a loaded system the protocol process may be preempted by other activities and not be scheduled fast enough to satisfy the protocol timing requirements. This can usually be guarded against by assigning the facsimile process a high scheduling priority. Unfortunately most UNIX systems do not support such facilities and so even if it is possible to receive serial line input with the minimum of delay, protocol timing requirements may not be met because of delays in scheduling the execution of the fax server process. For this reason the facsimile server processes attempt to raise their scheduling priority while actively sending or receiving facsimile. At other times, such as when doing queue management operations, processes run at a normal priority. On Silicon Graphics systems the ‘‘high priority’’ is a nondegrading scheduling priority that is above the priorities of the normal system processes. On SVR4-derived systems the ‘‘high priority’’ is a real-time scheduling priority. On other systems the server currently always runs at the same (normal) scheduling priority. For more details consult the `setProcessPriority` routine in **faxd/ModemServer.c++**.

In summary, If you want to use a Class 1 modem with this software and your system does not provide support for low latency serial line input you are likely to have troubles. If your system does not provide a mechanism for raising process scheduling priority (note that this is not the same as the UNIX ‘‘nice’’ parameter), then you may see problems when the server is under load. Exactly how much load will cause trouble is dependent on the system configuration and processing power.

HylaFAX table of contents.

CLASS 2 MODEM SUPPORT

The system includes a driver for modems that conform to TIA/EIA-592 draft SP-2388-A of August 30, 1991. Unfortunately, because this was only a draft, there are many modems that claim to conform to this document, but which diverge in sometimes subtle ways.

The Class 2 driver has been extensively tested with a wide variety of Class 2 modems. The software attempts to work around many incompatibilities and mistakes in modem firmware. A common problem that requires special attention is that some modems are incapable of accepting an `AT+FDIS` command to set the session parameters between the time a call is placed and a page is transmitted. Modems with this sort of problem may ignore the `AT+FDIS` command which causes incorrect session parameters being used to interpret the page data (often resulting in ‘‘squished pages’’), or abort the session, returning a `+FHNG:` status message to the host. If you encounter this problem use the `Class2DDISCmd` configuration parameter to enable workaround support in the fax server.

HylaFAX table of contents.

CLASS 2.0 MODEM SUPPORT

There is a driver for Class 2.0 modems that was developed initially using a USR Courier modem and later using a ZyXEL 1496E modem. Since this standard is very young do not be surprised if there are firmware problems and/or incompatibilities between the driver and modems with Class 2.0 firmware.

HylaFAX table of contents.

HYLAFAX SOURCE DISTRIBUTIONS

The source distribution for HylaFAX contains *all the code* in the system except a small bit of code used to build the Display PostScript-based imager for Silicon Graphics machines (this code is useless unless you have a **developers agreement** with Adobe for Display PostScript).

The system is written almost entirely in contemporary C++. It uses nested types and in several places assumes the compiler generates code according to the ANSI C++ Reference Manual (ARM). The normal development environment for the software is based on compiler products from Silicon Graphics. Every effort is made to insure the code builds correctly under all contemporary C++ compilers, including the GNU gcc compiler distribution. Beware however that not all compilers will compile this software.

Versions of gcc prior to 2.6.1 will not correctly build this software; gcc 2.6.3 and libg++ 2.6.2 are the current recommended versions of the GNU tools to use.

To build this software your system must have the following functionality or be capable of emulating those items that are missing:

- ANSI C compiler and runtime support (e.g.strftime).
- C++ compiler and runtime support. The C++ compiler must have support for nested types and ARM-style handling of temporary variables.
- ANSI C system include files with proper function prototype declarations (though some missing function declarations will be automatically handled).
- A make program that supports include files.
- Support for BSD-style sockets and TCP/IP for network communication, including the select interface for synchronous multiplexing of I/O streams.
- Support for FIFO special files.
- Support for BSD-style file locking (flock) or equivalent functionality from which it can be emulated (fcntl, lockf).
- POSIX 1003.1-style interfaces, including termios for manipulating tty devices.

When the software is configured for building any function that is not located will be emulated if it is known how. If a required facility is not found and no emulation or workaround is known, then the configuration procedure will abort.

The Class 1 modem driver requires sub-second timer facilities and a minimum latency interface to serial input. The server uses the BSD-style setitimer calls for the timer support and system-specific calls to enable low latency input from serial ports. If your system does not support the BSD timer calls, then the server will fall back to using the alarm system call that only has 1 second granularity (and the driver is unlikely to work reliably). If your system buffers serial input and does not provide a mechanism for defeating this, then the Class 1 driver will not work reliably. See the section above for specific requirements of the Class 1 support.

There is support in the fax server for handing inbound data calls to the system getty program. This support must be configured according to the requirements of the target platform; either System V-style or BSD-style. Only one of these two styles may be configured for use; the one that is appropriate to the target system should automatically be selected when the software is configured for building.

The system depends on the InterViews and TIFF software distributions. Only the *parts* of each system that are used by this software are included in the source distribution. Both distributions in their entirety are freely available by public FTP on the Internet.

Building the HylaFAX distribution from source code.

HylaFAX table of contents.

HylaFAX Binary Distributions

There are a number of pre-built binary distributions of the HylaFAX source code that are blessed as *safe for installation*. These distributions are available by public FTP from various sites on the Internet.

Beware of taking binary distributions from sites that are not on this list. HylaFAX includes several programs that are either started by the super-user or are installed as setuid root; thus installing untrusted distributions can be risky.

Many binary distributions are available on the master FTP site:

```
ftp.sgi.com                  sgi/fax/binary
    (192.48.153.1)
```

These currently include:

- Silicon Graphics *inst'able images*
- SVR4 *pkg images* for x86 systems
- HPUX-9.01 for HP9000/7xx systems
- Ultrix 4.4 for DecStation-5000 systems
- NetBSD-1.0 on x86 systems
- SunOS 4.1.x for Sparc

[*This list may be out of date, check the FTP directory.*]

Alternate sites that either mirror the master FTP site or that have specific binary distributions are:

These sites had binary distributions of FlexFAX, they will presumably have HylaFAX distributions soon.

SCO binary distributions are available from:

```
ftp.celestial.com            /pub/sco-ports/unix
    (192.136.111.2)
```

FreeBSD binary distributions are available from:

```
freebsd.cdrom.com            /pub/FreeBSD/packages
    (192.216.191.11)
```

NetBSD binary distributions are available from:

```
ftp.netbsd.org               /pub/NetBSD/packages/binaries/NetBSD-1.0/i386
    (128.32.138.88)
```

The binary distributions come with their own instructions because their format varies. The following guidelines describe how to interpret the filenames to understand the format and contents of each distribution.

Distribution filenames are of the form: `hylafax-`<*target*>-<*version*>-<*format*>; where <*target*> describes the target system, <*version*> specifies the HylaFAX version the distribution was built from, and <*format*> specifies the format of the distribution as follows:

- a `.Z` suffix indicates the file has been compressed with the compress(1) program
- a `.gz` suffix indicates the file has been compressed with the GNU gzip(1) program
- a `.tar` component indicates the file was created with the tar(1) program
- a `-inst` component indicates the file contains images for the IRIX inst(1) program
- a `-pkg` component indicates the file contains images for the SVR4 pkgadd(1M) program

So, for example, `hylafax-irix5-v3.0beta099-inst.tar` is a binary distribution for machines running IRIX 5.x and the distribution is provided as IRIX inst images collected together in a single archive created by the tar program.

In addition to the binary distributions, there may also be two other files present for each target system. Files of the form `INSTALL-*-*` are installation instructions for a specific binary distribution. Files of the form `gs*-*-tar.gz` are binary distributions of Ghostscript, the Ghostscript fonts, and any associated .ps files; configured for use with HylaFAX.

Installing a binary distribution of the HylaFAX software on an SGI machine .

HylaFAX table of contents.

Building HylaFAX from Source Ccode

This chapter contains the information needed to configure and build HylaFAX from the source distribution. The following sections are available here:

- Before You Start
- The Build Procedure
- Build Trees
- Configuration Files
- Configuring Optional Packages
- Configuring the HTML Documentation
- Upgrading From FlexFAX
- A Sample Configuration Session
- Configuration Parameters

Before You Start

Before you build the software you **may need to obtain certain other software distributions**. All the software packages described here are available by public ftp from a variety of Internet hosts; consult the archie resource location service to find copies close to your system.

gcc

You need a contemporary C++ compiler to build this system. *gcc version 2.6.3 is the current recommended version to use.* Versions of gcc prior to 2.6.1 will not work. When installing gcc beware of the make install step. On some systems and/or with some versions of gcc, the fax software may not compile properly if the include files are wrong (function prototypes that would normally cause casts to be done may be missing).

: Click here to use Archie to locate gcc 2.6.3

C++ runtime libraries

cfront-based compilers include everything that you need in the libC.a runtime library. When using GNU gcc you may also need the libg++ distribution. *libg++2.6.2 is the current recommended version to use.* Older versions may also work though it is recommended that you use the version appropriate for the compiler.

: Click here to use Archie to locate libg++ 2.6.2

ghostscript

If you are not on a Silicon Graphics machine, then you will need the Ghostscript PostScript interpreter software to build a PostScript imaging engine for use by the facsimile server. Version 2.6.1 and later are known to work. If you use 2.6.1 however, be certain to apply patches 1-4. If you do not apply the patches you will encounter a bug in the clipping code that is tickled by the default cover page distributed with this software. Also beware that if you use a version prior to 3.12 (inclusive) that there is a tiffg32d driver that writes incorrect 2D-encoded facsimile data; either do not configure this driver for use or disable its use by editing the **ps2fax.gs.sh** script that invokes Ghostscript to image PostScript for outbound facsimile jobs.

: Click here to use Archie to locate Ghostscript 2.6.1

gmake

The make files are extensive and work untouched with the system make under many systems. If your make does not understand them, then you should be able to use the GNU make (gmake) instead. If you decide to use gmake, be sure to get version 3.63 or newer; otherwise you may encounter problems (especially with the rules that automatically generate source code dependency information).

: Click here to use Archie to locate make

gawk

Several of the shell scripts included in this software make use of awk. These uses are reasonably simple, but will not work if your awk is old enough that it does not support functions or the `-v` command line option for setting variable values before the `BEGIN` action is executed. If you encounter problems using the standard awk on your system, try the GNU awk: gawk.

: Click here to use Archie to locate gawk

sed

Many of the shell scripts included in this software make use of sed. Certain systems are known to have versions of sed that do not handle the shell scripts. If you encounter problems, the latest GNU sed should be substituted.

/bin/test

The software configuration shell script (configure) and the modem configuration shell script (faxaddmodem) make heavy use of the test program. On some systems the **/bin/test** program does not support options such as `-c` (test if a file is a character special device). Source for a contemporary, public domain, test program is available by public ftp from ftp.uu.net.

ps2fax binary for IRIX systems

The Display PostScript-based imager for Silicon Graphics systems is included in the binary distribution images provided on sgi.com. If you choose to work from the source code on an IRIX system you may want a copy of the ps2fax program: it is available separately on sgi.com in the source distribution area. Note that this is a binary executable for IRIX 4.x systems and requires that the IRIX **dps_eoe** package be installed for proper use.

The Build Procedure

To build the software you need to first run the configure shell script that is located in the top level of the source directory. This script probes the target system for necessary tools and functions and constructs a build environment in which the software may be compiled. Once configuration is done, you simply run `make` to build the software and then `make install` to do the installation; for example:

```
% cd hylafax-v3.0beta099
% ./configure
    <...lots of messages...>
% make
    <...lots of messages...>
% su            # NB: installation must be done by the super-user
# make install
```

In general, the software is designed such that the following should be ‘‘*make-able*’’ in each directory:

```
make [all]      build stuff
make depend     build dependency information
make install    build&install stuff
make clean      remove .o files and cruft, but not executables
make clobber    remove everything that can be recreated
make distclean  remove absolutely everything that can be recreated
```

Note that after running "`make clobber`" or "`make distclean`" the configure script must be run again to create the Makefiles and other make-related files.

OSF/1 V1.3 and V3.0: *If you use the standard make you must type `make all` at the top level to get things to build (if you type `make` only the first directory will be processed).*

Build Trees

There are two schemes for configuring and building the software. If you intend to build the software for only one target system, you can configure the software so that it is built in the same directories as the source code.

```
% cd hylafax-v3.0beta099
% ls
hyla% ls
Distfile      config        distrules     faxmail       rules.in
HOWTO         config.guess  doc           faxrm         sendfax
Makefile.in   config.h.in   dps           faxstat       sendpage
RCS           config.site   etc           html          sgi2fax
README        config.sub    fax2ps        libtiff       util
TODO          configure     faxalter      man
VERSION       defs.in       faxcover      port
afm           dist          faxd          recvfax
% ./configure
```

Otherwise, you can configure a build tree that is parallel to the source tree hierarchy but which contains only configured files and files created during the build procedure.

```
% cd hylafax-v3.0beta099
% mkdir obj obj/mycpu
% cd obj/mycpu
% ../../configure
```

This second scheme is useful for:

- building multiple targets from a single source tree
- building from a read-only source tree (e.g. if you receive the distribution on CD-ROM)

Beware that if you choose to use the second scheme for configuring the software you must not use an absolute pathname when you run configure (i.e. a pathname that begins with ‘‘/’’) and the make that you use to build the software must correctly support the VPATH facility.

HPUX 9.05: *The standard make incorrectly processes VPATH; either use gmake or configure builds in the source tree.*

Solaris 2.3: *The standard make does VPATH processing incorrectly for files passed to the make dependency generator script (the last file in the list is not converted to a pathname relative to the source directory); this causes lots of messages that can be ignored.*

Configuration Files

The configuration process is critical to the proper compilation, installation, and operation of the software. The configure script runs a series of tests to decide whether or not the target system supports required functionality and, if it does not, whether it can emulate or workaround the missing functions. This procedure is fairly complicated and, due to the nonstandard nature of most UNIX systems, prone to error. The first time that you configure the software for use you should check the output from the configure script and look for anything that does not make sense for your system. A sample configure run is shown below together with an explanation of some of the work that is done.

A second function of the configure script is to set the default configuration parameters for the software. Many of these parameters can be changed after configuration through files that programs read at runtime, but some parameters cannot be changed without rerunning the configure script. Of particular note are the directories where the software is to be installed. By default the software is installed in the **/usr/local** hierarchy. To change this behaviour the appropriate parameters can be specified either:

- on the command line to configure,
- in a site-wide configuration file, or
- in a target-specific configuration file.

configure reads any site-wide configure file first, and then any target-specific configuration file. This permits target-specific definitions to override site-wide definitions.

Site-wide configuration files are named **config.site** and are automatically searched for first in any directory specified on the command line to configure (using the `-site` option), or if that fails, in the directory in in which the configure script is located.

Target-specific configuration files are named **config.local** and are looked for first in the top-level configuration directory, or, if that fails, in the directory in which the configure script is located.

Configuration files are just shell scripts that define shell variables that control the configuration process. For example, the following file might be used on a BSDI system to configure the software for installation in the **/usr/contrib** area and to make use of the Adobe Font Metric files that are already distributed as part of the BSDI 1.1 distribution:

```
#
# Parameters suitable for BSDI 1.1
#
DIR_AFM="/usr/contrib/lib/flexfax/afm"  # reuse existing files
DIR_BIN="/usr/contrib/bin"              # directory for client apps
DIR_SBIN="/usr/contrib/bin"             # directory for system apps
DIR_LIBEXEC="/usr/contrib/bin"          # directory for libraries&hidden apps
DIR_LIBDATA="/usr/contrib/lib/hylafax"  # directory for client data
```

```
DIR_MAN="/usr/contrib/man"              # directory for manual pages
DIR_SPOOL="/var/spool/hylafax"          # directory for spooling area
DIR_HTML="/usr/contrib/html/hylafax"    # directory for HTML documentation
DIR_CGI="/usr/contrib/html/cgi-bin"     # directory for CGI scripts
```

For a complete list of the possible configuration parameters, look at the sample **config.site** file provided in the distribution or check the section below.

Configuring Optional Packages

HylaFAX comes with several packages that are installed only as needed, or only if specifically configured at the time the configure script is run. Packages can be configured in a **config.site** or **config.local** file, or by using a `-with-<PACKAGE>` option when invoking configure; e.g. `configure -with-AFM`.

AFM Package
: The `AFM` package contains sample Adobe Font Metric files that came from the public dvips distribution. AFM files are required by the textfmt(1) and faxmail(1) programs for proper formatting of ASCII text. The AFM files used should reflect the characteristics of the fonts that are used for imaging. However, if the PostScript RIP used to image text does not come with AFM files for its native fonts, this package can be installed with only minimal degradation in the quality of the imaged text. By default this package is configured for installation only if font metric files do not appear to be present (`auto`).

DSO Package
: The `DSO` package controls whether or not to configure the software to build Dynamic Shared Objects for the TIFF library, for utility routines used by both client and server applications, and for a nucleus of common code used by server applications. Use of DSOs can significantly reduce the disk space needed for the HylaFAX software. If DSOs are not used then the code is statically linked into each application that uses it. By default this package is configured only if the system appears to suport DSOs in a way that fits into the normal build scheme (`auto`). If DSO support is *explicitly enabled* and there is no support for using DSOs in the expected way then DSOs are not used.

GETTY Package
: The `GETTY` package controls whether or not to configure BSD- or System V-style getty support for handling incoming data connections. By default this package is configured according to the target environment.

HTML Package
: The `HTML` package contains this HTML documentation about HylaFAX. By default this package is not configured for installation. Some HTML documentation can be viewed directly from the source directory; otherwise all the documentation can be viewed from the main WWW site at http://www.vix.com/hylafax/. You may want to configure the installation of the HTML materials if there is poor connectivity to the main WWW site. See the next section for more information about configuring the HTML materials.

PS Package

The `PS` package selects whether to configure use of Ghostscript (`gs`) or Display PostScript (`dps`) for imaging PostScript on the server. By default the DPS support is selected only for IRIX systems; otherwise Ghostscript is used. IRIX users may choose to use this control to select Ghostscript for use on their SGI systems.

SGI2FAX Package

The `SGI2FAX` package controls whether or not to configure the sgi2fax program that converts SGI RGB images to TIFF/F (for direct transmission as facsimile). By default this package is configured for installation only if the target system appears to have support for the SGI Image library that sgi2fax uses to read RGB images (`auto`). If this package is not installed then SGI RGB images must be converted to a form suitable for transmission using some other mechanism.

SYSVINIT Package

The `SYSVINIT` package controls whether or not to configure the System V-style support for automatically starting the HylaFAX queuer process from init(1). By default this support is configured for installation only if the target system appears to use a System V-style init (`auto`); i.e. the **/etc/rc0.d** and **/etc/rc2.d** directories exist.

Configuring the HTML Documentation

HylaFAX comes with extensive documentation written in HTML, the language used to author many documents found on the World Wide Web (WWW). These materials can be viewed at the main WWW site http://www.vix.com/hylafax/ or they can be installed locally for better access. The HTML documentation can be configured for installation from the source distribution by specifying `-with-HTML` as an option to the configure script, or by setting `HTML=yes` in a **config.site** or **config.local** file.

When the HTML package is configured for installation there are four other parameters that should be setup according to local needs:

`DIR_HTML`
: The directory where the HTML materials should be installed.

`DIR_CGI`
: The directory where the supporting CGI scripts should be installed.

`CGIPATH`
: The virtual pathname to use in the HTML documents to invoke CGI scripts installed in `DIR_CGI`.

`HTMLPATH`
: The virtual pathname to use in the HTML documents to reference materials installed in the `DIR_HTML` directory.

These parameters default to suitable values for the NCSA HTTP server:

```
DIR_HTML="/var/httpd/hylafax"
HTMLPATH="/hylafax"
DIR_CGI="/var/httpd/cgi-bin"
CGIPATH="/cgi-bin"
```

If you are not running an HTTP server it may be better to configure the materials for local access; for example:

```
DIR_HTML="/usr/local/html/hylafax"
HTMLPATH="/hylafax"
DIR_CGI="/usr/local/html/cgi-bin"
CGIPATH="/cgi-bin"
```

Note however that if the HTML materials are setup for local access then references ‘‘*up the tree*’’, will not work and CGI scripts will not be functional. For most of the materials provided with this distribution this will usually be acceptable.

There are a few other issues to consider when installing the CGI scripts:

- The `manpage` script that provides access to the UNIX manual pages may require some tailoring if manual pages are installed in non-standard locations or if the normal system man(1) command does not support the expected options. If you have problems consult the script.
- The `cat-pages` and `cat-faq` scripts will optionally return a cached copy of concatenated documents; consult the scripts for more information.
- If local support is to be provided, it may be worthwhile to alter the electronic mail addresses embedded in several of the CGI scripts.

[*Note that the CGI scripts are evolving; comments are welcome.*]

Upgrading From FlexFAX

If you were previously using FlexFAX beware that several things have changed in incompatible ways. The following is a list of potential things you should be aware of when configuring the software:

- The default server operation parameters set in config.h have changed; for example the default kill time for a fax job is now 3 hours instead of 24 hours. Verify the parameters in config.h are what you want.
- Numerous pathnames may have changed. The source-based installation procedure does not remove previously installed files unless they are to be overwritten; this can result in multiple versions of applications such as faxaddmodem being installed. Verify the parameters used for installation; you may want to tailor them to your environment by creating a **config.local** file with local configuration parameters.
- HylaFAX includes an improved **install.sh** script that is used to install files on non-IRIX systems. Certain files in the distribution are treated as *configuration-specific files* and are not overwritten if they appear to have local modifications. Instead the **install.sh** script will install the new file with a **.N** suffix and print a warning message. Be sure to reconcile these conflicts by comparing the old and new file contents with a program like diff(1); in most cases you should incorporate any local modifications into the new file and install it for use.

[*Ed: this list is expected to grow with experience.*]

There are other issues with upgrading from FlexFAX; they are discussed in the chapter that describes setting up the server.

A Sample Configuration Session

This section shows a sample configuration session and describes the work done. The session is shown indented in a `fixed width font` with user-supplied input in a **`bold font`**. Comments are shown in a normal or *italic* font. This session was collected on a 486 machine running BSDI 1.1.

```
wullbrandt% mkdir fax
wullbrandt% cd fax
wullbrandt% ln -s /hosts/oxford/usr/people/sam/fax src
```

A build tree separate from the source tree is used here. In fact, in this case the distribution is accessed from a read-only NFS-mounted filesystem.

```
wullbrandt% src/configure

Configuring HylaFAX (tm) (aka FlexFAX) v3.0beta095.

Reading site-wide parameters from src/config.site.
Fee, fie, foe, this smells like a i386-unknown-bsdi1.1 system.
```

Note that configure announces the distribution version and the deduced target configuration (`i386-unknown-bsdi1.1` here).

```
Using /usr/local/bin/gcc for a C compiler (set CC to override).
Looks like /usr/local/bin/gcc supports the -g option.
Looks like an ANSI C preprocessor.
... but __ANSI_CPP__ is not automatically defined, will compensate.
Looks like /usr/local/bin/gcc supports the -M option for make dependencies.
```

configure checked the normal shell search path for potential ANSI C compilers. The compiler is selected according to it properly compiling a small ANSI C test program. A specific compiler may be requested by setting the `CC` environment variable to the appropriate pathname, by supplying the parameter on the command line, e.g. `-with-CC=gcc`, or by setting `CC` in a configuration file.

Note that an ANSI C compiler is required to build the software. If a C compiler requires options to enable ANSI C compilation, they can be specified with the `ENVOPTS` *parameter.*

Once a compiler is selected configure does several checks to see:

- if the compiler accepts a -g option to enable the generation of debugging symbols,
- if the compiler includes an ANSI C preprocessor, and
- if the compiler will generate make dependency rules with a `-M` option.

By default the software automatically creates and maintains make dependency rules for all source code.

```
Using /usr/local/bin/gcc for a C++ compiler (set CXX to override).
Looks like /usr/local/bin/gcc supports the -g option.
Using " -g" for C++ compiler options.
Looks like an ANSI C++ preprocessor.
... but __ANSI_CPP__ is not automatically defined, will compensate.
```

A C++ compiler is selected using a scheme similar to the one used to find a C compiler. Likewise there is a `CXX` parameter that can be set to explicitly select a C++ compiler.

```
Using /usr/bin/make to configure the software.
Using ".include " syntax for Makefiles.
```

The make selected must support include files. configure can deduce several different syntaxes for specifying include files. Note that a large number of configuration parameters can be changed by editing the top-level **defs** file that is included by each Makefile in the distribution. A `MAKE` parameter can be used to control which make to use.

```
Using /bin/bash to process command scripts.
```

All the shell scripts included with this software are written using the Bourne Shell syntax. The standard shell on many systems is incapable of processing these scripts. configure prefers **bash** to **ksh** to **sh**.

```
Looks like -lutil is needed for wtmp file logging.
Looks like -lm is the library for math functions.
```

Various system-specific libraries that may or may not be needed are checked for. If your system requires a library that is not automatically included it can be specified by setting the `MACHDEPLIBS` parameter.

Creating port.h. The **port.h** file is included by all the C and C++ code in the system. It includes definitions for functions and type definitions that are missing from system include files, `#defines` to enable or disable system-specific functionality, and other odds and ends.

```
Creating port.h with necessary definitions.
... open FIFO files read-only
... use (sig_t) for signal handler type
... use (sig_t) for sigvec handler type
... use (sig_t) for sigaction handler type
... configure use of mmap for memory-mapped files
... configure use of fchown
... add function prototype for cuserid
... configure use of fchmod
... add function prototype for setutent
... add function prototype for endutent
... add function prototype for pututline
... configure use of  (internationalization support)
... configure use of
... configure use of
... configure use of logwtmp (BSD-style wtmp logging)
... add function prototype for logwtmp
... configure use of logout (BSD-style utmp support)
... add function prototype for logout
```

```
Done creating port.h.
```

This file can take a long time to create so configure generates the file only when it is needed, either because the file does not exist or because a different target or compiler is to be used. Note that running "`make distclean`" in the top-level directory of the build tree will remove the **port.h** file (along with all the other files generated by configure).

Selecting emulated library functions. Certain functions used by the software are not present on all systems and can be emulated using other system functionality. configure checks for the presence of such functions and if they are missing, will configure emulation code from the **port** directory to use instead. Building HylaFAX on unsupported systems may require adding to the code to the **port** directory.

```
Checking system libraries for functionality to emulate.
... emulate cuserid
Done checking system libraries.
```

If a routine must be emulated and configure does not automatically check for it, the routine name can be specified using the `PORTFUNCS` parameter. To add emulation support for a new function `foo`, create a file **port/foo.c** that contains the emulation code and then set `PORTFUNCS=foo` in a configuration file or modify the configure script to automatically check for the missing function.

```
Setting up libtiff configuration parameters.
Using memory mapped file support in libtiff.
Done setting up libtiff.
```

HylaFAX uses the TIFF library extensively. The only thing that should need to be configured for this software is whether or not to use memory-mapped files for reading TIFF images.

```
Checking for Dynamic Shared Object (DSO) support.
Done checking for DSO support.
```

If the `DSO` package is enabled (`DSO=auto` or `DSO=yes`), then configure will verify the system and compiler are capable of constructing SVR4-style DSO's in the expected way. Note that while a system may support DSO's the compiler may not be capable of generating the required position-independent code and/or the compiler may not pass the needed options through to the loader.

Selecting utility programs. configure locates various system utility programs that are used by the software. Note that absolute pathnames are selected because several scripts and programs excute with super-user privileges.

```
Selecting programs used during installation and operation.
Looks like /usr/bin/awk should be used to process command scripts.
Looks like /usr/sbin/sendmail should be used to deliver mail.
Looks like /usr/bin/mkfifo creates FIFO special files.
Looks like /bin/mv supports the -f option to force a move.
Looks like /bin/ln supports the -s option to create a symbolic link.
Done selecting programs.
```

Several things to note: The awk program selected for use must support functions using the System

V-style syntax (i.e. using `func` and not the POSIX-style `function`). configure will not select an awk program that does not support the syntax used by the awk scripts included in the distribution. Mail delivery is done using sendmail; a different mailer cannot be substituted without changing certain scripts because a number of sendmail-specific command line options are used.

Selecting default configuration parameters. The remainder of the work done by configure involves setting up configuration parameters that are mainly used during the *operation* of the fax software. These parameters can be set during configuration or, in most cases, they can be set through runtime configuration files.

```
Using uid uucp and gid dialer for controlling access to fax stuff.
Using uid bin and gid bin for installing programs.
Using LSB2MSB bit order for your i386 cpu.
Looks like you need BSD getty support.
Looks like /usr/libexec/getty is the program to exec for a data call.
Looks like you use binary-style UUCP lock files.
Looks like UUCP lock files go in /var/spool/uucp.
Looks like font metric information goes in /usr/contrib/lib/flexfax/afm.
Looks like the gs imager package should be used.
Looks like /usr/contrib/bin/gs is the PostScript RIP to use.
Looks like manual pages go in /usr/local/man.
Looks like manual pages should be installed with bsd-nroff-gzip-0.gz.

HylaFAX configuration parameters are:

[ 1] Directory for applications:        /usr/local/bin
[ 2] Directory for lib data files:      /usr/local/lib/fax
[ 3] Directory for lib executables:     /usr/local/lib/fax
[ 4] Directory for system apps:         /usr/local/bin
[ 5] Directory for manual pages:        /usr/local/man
[ 6] Directory for HTML documentation:  /usr/local/doc/fax
[ 7] Directory for spooling:            /var/spool/fax
[ 8] Directory for font metrics:        /usr/contrib/lib/flexfax/afm
[ 9] Directory for uucp lock files:     /var/spool/uucp
[10] Uucp lock file scheme:             binary
[11] PostScript imager package:         gs
[12] PostScript imager program:         /usr/contrib/bin/gs
[13] Manual page installation scheme:   bsd-nroff-gzip-0.gz
[14] Default page size:                 North American Letter
[15] Default vertical res (lpi):        98
[16] Location of getty program:         /usr/libexec/getty
[17] Location of sendmail program:      /usr/sbin/sendmail

Are these ok [yes]?
```

At this point you can interactively modify any of the displayed parameters. Hitting a carriage return or typing `yes` will accept the current parameters. Typing one of the number displayed along the left hand side causes configure to prompt for a new value of the specified parameter. Typing anything else causes configure to prompt for a new value *for each parameter*. In general hitting carriage return will accept the current value and typing anything that is unacceptable will cause a help message to be displayed. A description of each of the configuration parameters is given below.

Once acceptable parameters are setup configure will generate all the files that depend on these parameters. Note that certain files may or may not be created based on the configuration of optional packages and/or the functions supported by target system.

```
Creating defs from src/defs.in
Creating rules from src/rules.in
Creating Makefile from src/Makefile.in
    ...stuff deleted...
Setting up make dependency files.
Done.
```

Configuration Parameters

This section gives a brief description of the less obvious configuration parameters. Consult the distributed **config.site** for a *complete* list of parameters. The list here is sorted alphabetically.

The default setting for most configurations parameters is usually correct. Be especially careful about enabling or overriding parameters that control workarounds for system bugs; certain problems can cause server processes to go into infinite loops!

Note that HylaFAX is composed of *client* and *server* applications. Client applications are programs that normal users invoke to send facsimile, query the status of facsimile servers, etc. Server applications are programs that reside *only on the machine where the fax modems are present*.

`AROPTS`
: The options to ar for creating an archive. Note that configure will automatically check to see if ar supports an `s` to create a symbol table instead of using ranlib.

`CONFIG_OPENFIFO`
: The mode to use when opening a FIFO special file in a server process. Normally this should be `O_RDONLY`, but on some systems this must be `O_RDWR` to avoid kernel bugs that cause select system calls to return prematurely after a process closes the ‘‘client side’’ of a FIFO. By default this parameter is set according to the target system.

`CONFIG_FIFOBUG`
: Whether or not to enable a workaround for a kernel bug that causes select system calls to return prematurely after a process closes the ‘‘client side’’ of a FIFO special file. Setting this parameter to `yes` causes server processes to close and reopen FIFO special files after each received message. Note that aside from the additional overhead incurred in the server processes, this problem can also cause client programs to be temporarily unable to reach a server process. To minimize this possibility, the client code tries to reach a server process up to 5 times before giving up; this can introduce noticeable delays in applications such as faxstat.

`CONFIG_SELECTBUG`
: On some systems where the FIFO select bug (see above) exists the select system call may return bits set in the read, write, and exception masks even if they were not requested; this in turn can cause the Dispatcher code to abort. Setting this parameter to `yes` causes the Dispatcher to ignore any bits set in bit vectors returned by select if the bits were not requested. Enabling this workaround adds additional overhead to each server process.

`CONFIG_WINSZHACK`
: On some systems the `TIOCWINSZ` ioctl is defined, but its use requires the inclusion of two non-standard files. Setting this paramter to `yes` causes the inclusion of these files.

`CONFIG_ABORTBUG`

On some systems the logic used by the sever processes to poll for messages on a FIFO special file while actively doing work can cause the process to go into an infinite loop. Setting this parameter to `yes` causes the server processes to **not** poll for messages. Enabling this workaround means that the faxgetty process will not recognize messages to abort a facsimile while it is being received.

`CONFIG_NOSTDINDUP`

On some systems, if the standard output is redirected to be the same as the standard input, then the stty program will emit a warning message. This is necessary for some older systems because it is the only way to force stty to change parameters on a tty device that is not the controlling tty. Setting this parameter to `yes` causes the ondelay program to not set stdout to stdin. Enabling this workaround can break the faxaddmodem program; it should be used only on systems where stty supports the `-f` option to select a tty device other than the controlling tty.

`CXXFILE`

The options to the C++ compiler required to get the compiler to process a file with a **.c++** suffix as C++ source code. configure automatically sets this parameter to `"-x c++"` for gcc and to `"-+"` for the IBM xlC compiler under AIX.

`DEFVRES`

The default vertical resolution in lines/inch that clients should use for submitted facsimile. Choices are 98 and 196.

`DIR_AFM`

The directory where client applications should find Adobe Font Metric (AFM) files. These files are used by various HylaFAX programs when converting text to PostScript.

`DIR_BIN`

The directory where client applications should be installed; by default this is **/usr/local/bin**.

`DIR_CGI`

The directory where CGI scripts used by the HTML-based HylaFAX documentation should be installed; by default this is **/var/httpd/cgi-bin** (usually appropriate for the NCSA HTTP server).

`DIR_HTML`

The directory where HTML-based HylaFAX documentation should be installed; by default this is **/var/httpd/htdocs/hylafax** (usually appropriate for the NCSA HTTP server).

`DIR_LIBDATA`

The directory to install data files that are required by client applications; by default this is **/usr/local/lib/fax**.

`DIR_LIBEXEC`

The directory to install executable programs that are invoked by other applications and not from the command line (e.g. the textfmt program invoked by sendfax to convert text to PostScript when submitting a fax job); by default this is **/usr/local/lib/fax**.

`DIR_LOCKS`

The directory in which to create UUCP lock files.

`DIR_MAN`

The top-most directory of the manual area where client/server manual pages should be installed.

`DIR_BIN`

The directory where server-specific applications should be installed; by default this is **DIR_SPOOL/bin**.

`DIR_SBIN`

The directory where system applications such as servers should be installed; by default this is **/usr/local/bin**.

`DIR_SPOOL`

The directory in which the HylaFAX server spooling area should be setup; by default this is

/var/spool/fax.

`DSODELAY`
: When DSO's are built, the option to specify to `CC` and/or `CXX` to note that a library should be loaded only as needed. This option is used to delay the loading of the TIFF library for applications that may use it infrequently.

`DSOOPTS`
: When DSO's are built, the options to specify to `CC` and/or `CXX` to create a DSO.

`DSOSUF`
: When DSO's are built, the filename suffix for a DSO. If this is set to `"a"` then statically linked archives are used.

`ENVOPTS`
: Options to pass to `CC` to force ANSI C compilation.

`FAXGID`
: The group ID to use for the fax user; by default this is selected according to the target system.

`FAXUID`
: The user ID to use for the fax user; by default `uucp`.

`FILLORDER`
: The order of bits in a byte on the server machine; either LSB2MSB or MSB2LSB. This is normally selected according to the target system.

`GCOPTS`
: Special options to pass the C compiler. If this parameter is set, then configure may append other options to this list.

`GCXXOPTS`
: Special options to pass the C++ compiler. If this parameter is set, then configure may append other options to this list.

`INSTALL`
: The pathname of the install program to use. Note that this program must emulate the command line interface used by the IRIX install program.

`LIBMALLOC`
: Whether or not to use `-lmalloc` in building the software. By default (`auto`) configure will probe the system to see if the library is present; if it is then the software will be configured to use it when building. Other possible values are: `yes` and `no` to enable and disable use, respectively. On systems that support DSOs it may be important to disable the use of `-lmalloc` to avoid conflicts with the normal memory alocation routines present in the standard C library.

`LIBPORT`
: The pathname of the library that holds code to emulate missing system functionality. Normally this parameter is set by configure based on whether or not emulation code is required for the target.

`LIBSUN`
: Whether or not to use `-lsun` in building the software. By default (`auto`) configure will probe the system to see if the library is present; if it is then the software will be configured to use it when building. Other possible values are: `yes` and `no` to enable and disable use, respectively.

`LLDOPTS`
: Extra command line options passed to `CC` and `CXX` when linking an executable. This option is usually set only when DSO support is enabled (to force the executable to search for the HylaFAX-specific DSO's in non-standard locations in the filesystem.)

`LOCKS`
: A specification of the default type of UUCP lock files to use. This parameter is usually one of

ascii, binary, -ascii or +ascii; consult the config(4F) manual page for a description of the **UUCPLockType** configuration parameter.

MACHDEPLIBS

Target-dependent libraries that should be used when linking applications. Note that if this parameter is specified configure will append to the list of libraries.

MAKECXXOVERRIDE

A special parameter used by configure when constructing Makefiles. This parameter is required when compiling the software with the SunPRO C++ compiler; it works around limitations in the SunPRO compilation environment.

MAKEDEPINCLUDE

The keyword used to specify the inclusion of a make dependency file when constructing Makefiles. This parameter is set to MAKEINCLUDE if make dependendcies are to be constructed by the software; otherwise it is set to # so that no make dependency rules are included (and used).

MAKEDSOINCLUDE

The keyword used to specify inclusion of a file containing rules for building DSO's. This parameter is set to MAKEINCLUDE if DSO's are to be built; otherwise it is set to # so that nothing is included.

MAKEINCLUDE, MAKELQUOTE, MAKERQUOTE

The syntax used to specify an include file for make; e.g. @<>MAKEINCLUDE<>@ @<>MAKELQUOTE<>@*file*@<>MAKERQUOTE<>@. These parameters are normally deduced by configure depending on the capabilities of the make program to use.

MANSCHEME

The scheme to use when preparing and installing manual pages. Schemes are constructed according to:

<*organization*>-<*formatting*>-<*compression*>[-<*suffix*>]

where: <*organization*> is either bsd for BSD-style section organization (e.g. file formats in section 5) or sysv for System V-style organization (e.g. file formats in section 4). <*formatting*> is either nroff to force installation of formatted materials (using nroff) or source to get the nroff source installed. <*compression*> is either the name of a program to compress the manual pages (gipz, compress, pack) or cat for uncompressed data. <*suffix*> is either the file suffix to convert installed pages to (e.g. 0.gz for gzip-compressed pages under BSD) or strip to force the normal ".4f" suffix to be converted to ".4" (or ".5" if using the BSD organization). If no -<suffix> is specified then filenames are not converted when they are installed.

PAGESIZE

The default page size that client applications will use for submitted facsimile. Page sizes are specified by name and checked against the pagesizes database installed in the DIR_LIBDATA directory. See also pagesizes(4F).

PATH_DPSRIP

The absolute pathname of the Display PostScript-based imager program.

PATH_GETTY

The absolute pathname of suitable getty program to exec to handle an inbound data call. Note that this parameter must be set correctly before the software is built; there is no mechanism for doing this through a runtime configuration file.

PATH_GSRIP

The absolute pathname of the Ghostscript-based imager program. Beware that this must be a version of the **gs** program that includes the tiffg3 device driver.

PATH_SENDMAIL

The absolute pathname of the sendmail program on the server machine; sendmail is used by

HylaFAX to post mail messages for a variety of reasons.

`PORTFUNCS`
: A list of non-standard functions that should be emulated. Normally this list is constructed by configure based on checks it does. If this parameter is set, configure will append to the specified list.

`PROTOTYPES`
: Options that should be passed to the C compiler to force it to check for function prototype mismatches. This parameter should not be needed because ANSI C compilers should do this automatically.

`SETMAKE`
: If make does not automatically set `$MAKE` to the name of the make program to invoke for subdirectories, then configure will create an explicit definition. If this parameter is set, then it will be used instead.

`SHDLIBC`
: A special library to use in place of the standard `-lc` implicitly provided by `CC` and/or `CXX`.

`SIGHANDLERTYPES`
: The type signatures to use when checking for the correct type for a signal handler. If this parameter is set then configure will check these types before it checks a builtin list of well-known values.

`SYSGID`
: The group ID to use for installing non-setgid programs; by default this is chosen according to the target system.

`SYSUID`
: The group ID to use for installing non-setuid programs; by default `bin` is used.

How to setup a HylaFAX server and prepare it for use.

HylaFAX table of contents.

HOW TO BUILD GHOSTSCRIPT WITH THE TIFF DRIVER

If you are not using the Display PostScript-based imager on a Silicon Graphics machine, then you need to build a version of Ghostscript that includes the tiffg3 driver provided with version 2.6.x and later. Ghostscript and its fonts can be found at many public ftp sites around the Internet. A good way to find the nearest public ftp site that has version 2.6.1 is with archie:

: Click here to use Archie to locate Ghostscript 2.6.1

To setup Ghostscript you need to:

1. Create a Makefile (follow the Ghostscript documentation).
2. Add "tiffg3.dev" to the list of configured devices.
3. Build a **gs** executable that includes the tiffg3 driver.
4. Verify that you have configured your **gs** executable correctly by invoking it with the -h option and looking for the tiffg3 driver in the list of available devices; e.g.

```
oxford% gs -h
Aladdin Ghostscript version 2.9.9 (6/23/1994)
Copyright (C) 1990-1994 Aladdin Enterprises, Menlo Park, CA.
Usage: gs [switches] [file1.ps file2.ps ...]
Available devices:
        x11 tiffg3 tiffg4
Search path:
    .
    /usr/local/lib/ghostscript/gs-2.9.7:/usr/local/lib/ghostscript/fonts
Most frequently used switches: (you can use # in place of =)
    -d[=]   define name as token, or true if no token given
    -dNOPAUSE            don't pause between pages
    -gx   set width and height ('geometry'), in pixels
    -q                   'quiet' mode, suppress most messages
    -r              set resolution, in pixels per inch
    -s=    define name as string
    -sDEVICE=   select initial device
    -sOutputFile=  select output file: embed 268501088 for page #,
                           - means stdout, use |command to pipe
    -                    read from stdin (e.g., a pipe) non-interactively
For more information, see the (plain text) file use.doc
  in the directory (null).
```

5. Install the **gs** executable and the associated runtime support code (e.g. ‘‘make install’’).
6. Install, if necessary, the Ghostscript fonts (these may come separately from the Ghostscript source distribution).

Beware that in recent versions of Ghostscript the PostScript Level II support requires the Independent JPEG Group (IJG) distribution; this software typically comes bundled in an archive separate from the Ghostscript distribution. This means that you may need to fetch three archives to setup a working Ghostscript:

- Ghostscript source code
- Ghostscript/IJG code
- Ghostscript fonts

Consult the Ghostscript documentation for specifics.

*Note that on machines with dynamic shared libraries (e.g. SunOS), if you link Ghostscript with the X11 device driver and use shared X11 libraries that are not in a standard location, then you may need to augment the HylaFAX **util/ps2fax.gs.sh** script with something of the form:*

```
LD_LIBRARY_PATH=/usr/local/R5/lib:/usr/openwin/lib
export LD_LIBRARY_PATH
```

Beware that while many versions of Ghostscript between 2.6.1 and 3.12 have tiffg3 drivers that purport to support 2D-encoded facsimile data, none do so correctly. Versions post-3.12 should generate correct 2D-encoded data.

HylaFAX table of contents.

INSTALLATION FROM A BINARY DISTRIBUTION

This chapter contains the information about installing HylaFAX from a binary distribution. There are multiple formats for binary distributions. The following sections are available here:

- Binary Distributions for IRIX

Binary Distributions for IRIX

The IRIX binary distribution comes as a tar file that contains images for use with the standard Silicon Graphics installation programs inst(1) and swmgr(1). To unpack the inst images do something like:

```
hyla% tar xf <somewhere>/hylafax-v3.0-inst-tar
```

Next, run inst or swmgr to install the appropriate packages that you want.

```
hyla# inst -f dist/hylafax

Default distribution to install from: dist/hylafax

For help on inst commands, type "help overview".


Inst Main Menu

 1. from [source]                 Specify location of software to be installed
 2. list [keywords] [names]       Display information about software subsystems
 3. go                            Perform software installation and removal now
 4. install [keywords] [names]    Select subsystems to be installed
 5. remove [keywords] [names]     Select subsystems to be removed
 6. keep [keywords] [names]       Do not install or remove these subsystems
 7. step [keywords] [names]       Interactive mode for install/remove/keep
 8. conflicts [choice ...]        List or resolve installation conflicts
 9. help [topic]                  Get help in general or on a specific word
10. view ...                      Go to the View Commands Menu
11. admin ...                     Go to the Administrative Commands Menu
12. quit                          Terminate software installation

Inst> list
Reading installation history .. 100% Done.
Reading distribution .. 100% Done.
Checking dependencies .. 100% Done.
Calculating sizes .. 100% Done.
Current View:
  Location:  distribution
  Status:    N=new,U=upgrade,P=patch upgrd,S=same,D=downgrade,' '=not installed
  Selection: i=install, r=remove, k=keep
  Level:     subsystem
  Name:      name

  Subsystem Type(s) [bdrp]: b=reBoot needed, d=Default, r=Required, p=Patch

  N  hylafax.html.docs                0   HylaFAX Online Documentation (in
                                            HTML)
  N  hylafax.html.faq                 0   HylaFAX Frequently Asked Questions
                                            (in HTML)
  N  hylafax.html.tools               0   HylaFAX CGI Scripts & Tools
i N  hylafax.man.client [d]         62+   HylaFAX Client Manual Pages
```

```
  N  hylafax.man.server                  0   HylaFAX Server Manual Pages
  N  hylafax.sw.afm                      0   HylaFAX Font Metric Files (for
                                               client)
i N  hylafax.sw.client [d]            1577+  HylaFAX Client Software
  N  hylafax.sw.dpsrip                   0   HylaFAX DPS-based PostScript RIP
                                               (for server)
  N  hylafax.sw.server                   0   HylaFAX Server Software

Disk space summary (Kbytes):             / /oxford/usr/people
Selections net change                    0              1638+
Space available                     442049            116269
Inst>
```

Note that there are separate images for *client* machines and for *server* machines: a server machine is a machine that has one or more fax modems on it; a client machine is any machine where facsimile are to be submitted. It is a good idea to install the client support on a server machine.

The **hyalfax.sw.afm** package should be installed on client machines that do not have **dps_eoe.sw.dpsfonts** installed; this package includes font metric files that are used in formatting ASCII text submitted for facsimile transmission. If you have access to **dps_eoe.sw.dpsfonts**, install that instead of **hyalfax.sw.afm**; you will get more accurate formatting of text because the font metric files will be correct for the fonts used to image text.

When installing software for a server machine you must also install a PostScript RIP, otherwise the fax server will not able to convert PostScript to facsimile for transmission. If you have **dps_eoe.sw.dps** and **dps_eoe.sw.dpsfonts** on the server machine then you can use the Display PostScript-based RIP in **hylafax.sw.dpsrip**; otherwise a Ghostscript-based RIP is provided in **hylafax.sw.gsrip**. [*Ed: **hylafax.sw.gsrip** is not yet ready for distribution.*]

```
Inst> install *.*.server *.*.dpsrip
Inst> list
Current View:
  Location:  distribution
  Status:    N=new,U=upgrade,P=patch upgrd,S=same,D=downgrade,' '=not installed
  Selection: i=install, r=remove, k=keep
  Level:     subsystem
  Name:      name

  Subsystem Type(s) [bdrp]: b=reBoot needed, d=Default, r=Required, p=Patch

  N  hylafax.html.docs                   0   HylaFAX Online Documentation (in
                                               HTML)
  N  hylafax.html.faq                    0   HylaFAX Frequently Asked Questions
                                               (in HTML)
  N  hylafax.html.tools                  0   HylaFAX CGI Scripts & Tools
i N  hylafax.man.client [d]             62+  HylaFAX Client Manual Pages
i N  hylafax.man.server                148+  HylaFAX Server Manual Pages
  N  hylafax.sw.afm                      0   HylaFAX Font Metric Files (for
                                               client)
i N  hylafax.sw.client [d]            1577+  HylaFAX Client Software
i N  hylafax.sw.dpsrip                1041+  HylaFAX DPS-based PostScript RIP
                                               (for server)
i N  hylafax.sw.server                1269+  HylaFAX Server Software

Disk space summary (Kbytes):             / /oxford/usr/people
Selections net change                 2309+             1786+
```

```
Space available                    442001          116261
```

The HTML documentation is in the **hylafax.html.*** images, configured for installation in the directories **/var/httpd/htdocs/hylafax** (documents) and **/var/httpd/cgi-bin** (CGI scripts); you may need to add symbolic links to these directories if your HTTP server is configured with a different document root (or add entries to the HTTP server's configuration files). Note that the HTML materials are also available on the WWW at http://www.vix.com/hylafax; the snapshot included with the distribution is provided for those sites that are not connected to the Internet.

If you do not have an HTTP server on the machine where you want to install the HTML materials you can view the documentation locally using your favorite World Wide Web browser; e.g.

```
hyla% netscape /var/httpd/htdocs/hylafax
```

Note however that if the materials are viewed without an intervening HTTP server then the CGI scripts will not be functional and you will not be able to follow links that reference *up* the document hierarchy.

How to setup a HylaFAX server and prepare it for use.
HylaFAX table of contents.

SERVER SETUP AND BASIC CONFIGURATION

HylaFAX is composed of *client* and *server* applications. Client applications are programs that normal users invoke to send facsimile, query the status of facsimile servers, etc. Server applications are programs that reside *only on the machine where the fax modems are present*. HylaFAX is distributed so that the normal `make install` step done after the software is built will install both client and server applications. Client-only systems require a slightly different procedure that is discussed in the next chapter on Client Setup and Operation. This chapter discusses setting up a server machine for use.

The basic steps required to setup a HylaFAX server machine are:

1. Install the HylaFAX software.
2. Select a facsimile modem for use.
3. Check your modem is functional.
4. Select a flow control scheme to use for facscimile communication.
5. Select a TTY device to use.
6. Configure the modem for outbound use by HylaFAX.
7. Start/restart the HylaFAX scheduler process.
8. Setup the modem for inbound use by HylaFAX [*optional*].

It is also possible to configure a modem for inbound use only; this is done simply by not running the HylaFAX scheduler process. Finally, note that many decisions are system-specific. Much system-specific guidance is sprinkled throughout these materials; the following sections also provide specific guidance for individual systems where HylaFAX has been run.

- BSDI Guidance

- IRIX Guidance
- Solaris Guidance
- SunOS Guidance
- Ultrix Guidance

This chapter also has a section on Modem Configuration Issues. Advanced configuration and setup issues are discussed in a separate chapter.

Installing HylaFAX

As noted previously, most installations of HylaFAX will be done from the source code distribution. In this case installation is done by running

```
# make install                    # NB: must be done by the super-user
```

from the top-level of the build tree.

If you are working from a binary distribution the above step is also common, though different distribution formats may require different techniques for installing the software. For Silicon Graphics users the installation technique is discussed in Binary Distributions for IRIX.

Selecting a Facsimile Modem

Selecting a modem usually means purchasing a modem that is capable of sending and receiving facsimile. Most any fax-capable modem can be used with HylaFAX but not all are equal. HylaFAX has drivers for Class 1, Class 2, and Class 2.0 facsimile modems, but there are caveats on the use of Class 1 modems. Class 2 modems are the most common, but due to the non-standard nature of the specification that they are implemented against compatibility can be a problem. Note also that the quality of Class 2 modems varies significantly. Class 2.0 modems follow the latest standard, a ratified version of the specification used in implementing Class 2 modems. There are significantly fewer Class 2.0 modems available, though the quality of these modems also varies significantly. There is a list of modems that have been tried with HylaFAX and this list includes several modems that have been found to be reliable for use in sending and receiving facsimile.

Checking Your Modem

Once you have a modem to use with HylaFAX first make sure that the modem works for data use. One can not say this enough. If you can not use *cu*, *tip*, *kermit*, or whatever with your modem, do not try to configure it for use with HylaFAX. This means in particular that you should verify that you have a working cable between your host and modem and that this cable is suitable for use. That is, that the cable has the relevant signals for doing hardware flow control if that is necessary and that it passes the

DCD and DTR signals appropriately.

Verify that the modem you are using is a fax modem. This can be done by communicating directly with the modem or by using the probemodem shell script that comes with HylaFAX.

To verify the modem can be used with HylaFAX use a communication program such as cu to talk to the modem directly, for example:

```
% cu -l ttyf2
Connected
at+fclass=?
0,1,2
OK
~[oxford].
Disconnected
```

The `at+fclass=?` command asks the modem to report which *classes* it is capable of supporting. Class 0 is for data use. Class 1, Class 2, and Class 2.0 are for facsimile use. Other classes may be reported, for example, for modems that provide digitized voice support. HylaFAX can be used with any modem that supports Class 1, Class 2, or Class 2.0.

The probemodem script can be used to obtain information about which facsimile capabilities a modem implements. It works by sending commands to the modem to query known capabilities and reporting the results of the queries. probemodem is normally located in the same location as the faxaddmodem script and is run in a similar manner:

```
% su                      # NB: probemodem must be run by the super-user
# /var/spool/fax/bin/probemodem ttym2

Now we are going to probe the tty port.  This takes a few seconds,
so be patient.  Note that if you do not have the modem cabled to
the port, or the modem is turned off, this may hang (just go and
cable up the modem or turn it on, or whatever).

Probing for best speed to talk to modem: 38400 OK.

This looks like a Class 1+2 modem.

ATI0    RESULT = "OK"   RESPONSE = "144"
ATI1    RESULT = "OK"   RESPONSE = "176"
ATI2    RESULT = "OK"   RESPONSE = "OK"
ATI3    RESULT = "OK"   RESPONSE = "03134479"

Class 1 stuff...

AT+FCLASS=?     RESULT = "OK"   RESPONSE = "0,1,2"
AT+FCLASS?      RESULT = "OK"   RESPONSE = "2"
AT+FCLASS=0     RESULT = "OK"   RESPONSE = "OK"

        <...stuff deleted...>
```

For the purpose of verifying that a modem is capable of use with HylaFAX, the most important information to check in the output of probemodem is the response to the `AT+FCLASS=?` command.

Selecting a Flow Control Scheme

The rules to use for selecting a flow control method for facsimile use are:

- If you have a Class 1 modem, then you can use either hardware or software flow control, but beware of using hardware flow control as the Class 1 specification only requires vendors to support software flow control (and many of the Class 1 modems tried so far do not properly support hardware flow control).
- If you have a Class 2 or Class 2.0 modem, then you can use either hardware or software flow control, but if you are going to communicate with the modem faster than 9600 baud then you should probably use hardware flow control.

Beware that although a modem may properly implement hardware flow control when doing data communication, it may not support hardware flow control during facsimile communication. Consult the modem information for specifics on some modems.

Choosing a TTY Device

There are two things to beware of in selecting a tty device file to use with your modem: flow control usage and port locking mechanisms.

On many systems different devices are used to select different flow control schemes and/or whether or not the system will monitor the DCD signal. For example, IRIX systems use different device names to identify devices that monitor DCD and/or support RTS/CTS flow control. Likewise the FAS driver for SCO uses a different names as does the standard HP-UX terminal driver.

On some systems inbound and outbound port use is interlocked by using a pair of devices, one for inbound use and another for outbound use. Typically this scheme works by stopping programs that use the inbound device until an inbound call is received (and DCD is raised by the modem). Outbound usage is also interlocked against applications waiting for the inbound device. HylaFAX provides no direct support for this because this scheme requires that a modem auto-answer incoming calls (something that does not work with virtually any multi-mode, i.e. fax and data, modem). When faced with a system that uses this you have several alternatives. Most people elect to avoid the inbound device and run both incoming and outgoing traffic on the outbound device, using the builtin interlocking mechanism provided by HylaFAX. In this case the appropriate device to use is typically named **/dev/cu***. Systems that have this style of device usage include BSDI and SunOS.

Configuring Outbound Use

Modems are configured for use with HylaFAX using the faxaddmodem script. This is an interactive script that walks you through the configuration and installation of a new or existing modem. Note that even if you have a previous version of this software installed you should run *faxaddmodem* to update the configuration information for your modems. faxaddmodem actually does a lot of work other than configuring a modem for use by HylaFAX. Specifically, before faxaddmodem even begins setting up a modem it:

- Verifies that a ''fax'' user is registered in the password file or YP/NIS database. If one is not setup, then faxaddmodem sets one up for you. The fax user is used in various places in HylaFAX and should normally be setup to have the same user ID as that of uucp so that UUCP lock files can be shared. (If you run IRIX the ''fax'' user will also be added to the **passwd.sgi** file so that it does not show up in the normal palette of users displayed during the login procedure.)
- Verifies that a suitable entry exists in the **/etc/services** file or in the equivalent YP/NIS database. If no entry is present, then one is setup.
- Verifies that the inetd(1M) configuration file includes an entry for the HylaFAX client-server protocol process faxd.recv(1M) and that the pathname corresponds to the one that was configured. If an entry is not setup for faxd.recv, then one is added.
- Verifies that a ''FaxMaster'' entry is present in the aliases database. This alias is equivalent to the normal PostMaster alias that is used to deliver mail-related problems; HylaFAX directs notices about problems and received facsimile to this alias. The FaxMaster alias should list those system administrators that will handle HylaFAX-specific problems. If an alias is not present, then faxaddmodem sets one up.

[**NB**: *faxaddmodem prompts for permission before doing any of the above actions*.]

The remainder of this section shows a sample configuration session and describes the work done. The session is shown indented in a `fixed width font` with user-supplied input in a **`bold font`**. Comments are shown in a normal or *italic* font. This session was collected on an SGI Indigo running IRIX 5.3 and communicating with an AT&T DataPort modem.

Note that if your modem is configured to communicate to the host at fixed baud rate, then you should use the `-s` option to lock the host-modem line rate; consult the *faxaddmodem* manual page for details.

```
oxford# faxaddmodem             # NB: must be run as the super-user
Verifying that your system is setup properly for fax service...

Done verifying system setup.

Serial port that modem is connected to []? ttym2

Ok, time to setup a configuration file for the modem.  The manual
page config(4F) may be useful during this process.  Also be aware
that at any time you can safely interrupt this procedure.

No existing configuration, let's do this from scratch.
```

If you run faxaddmodem and a configuration file already exists, the server-specific parameters will be carried over to the new configuration file, but the modem-specific parameters will not. This is because faxaddmodem is not intelligent enough to recognize when the old modem and the new modem are the same type. In this situation you may prefer to edit the configuration file using your favorite text editor instead of using faxaddmodem.

```
Country code [1]?
Area code [415]? 510
```

The area code may be set to a null string or omitted from the configuration file in countries where the notion of an area code does not exist.

```
Phone number of fax modem [+1.415.965.7824]? +1 510 526-8781
```

This is the phone number associated with the modem being configured. By default it is passed as an ‘‘identity’’ to peer fax machines (see below) and it may also appear on tag lines created by the fax server. The phone number should be a complete international dialing specification in the form +<country code> <area code> <local part>. Any other characters that are included for readability are automatically removed if they might cause problems.

```
Local identification string (for TSI/CIG) [+14159657824]? +1 510 526-8781
```

The local identification string is passed to peer fax machines during communication. If it is not specified, or set to a null string, then the canonical phone number for the fax modem is used instead. Beware that the CCITT T.30 protocol for facsimile communication specifies that this string should contain only numbers, blank, and the + symbol. In practice however, most facsimile machines will accept any ASCII string.

```
Long distance dialing prefix [1]?
International dialing prefix [011]?
Dial string rules file (relative to /var/spool/fax) [etc/dialrules]? etc/dialru
```

The dial string rules file holds rules used to convert user-supplied dialing strings (i.e. phone numbers) to a canonical format and to prepare strings for delivery to a modem. The default rules do very little. Specific dialing rules may be useful for your site or locale based on how your modems are connected to the PTT. Consult the dialrules(4F) manual page for a description of this file.

```
Tracing during normal server operation [1]?
Tracing during send and receive sessions [11]?
```

These parameters control the tracing/logging done by the server process. It is important that tracing during send and receive sessions include sufficient information to diagnose problems. For Class 1 modems this parameter is usually set to `0x4f` so that HDLC frames are included in the logs. For Class 2 and Class 2.0 modems the default setting of 11 is typically ok. Consult the description of the **ServerTracing** and **SessionTracing** configuration parameters in the config(4F) manual page for detailed information about these parameters.

```
Protection mode for received facsimile [0600]?
```

The default parameter selected here makes all received facsimile accessible only to the fax user. It may be desirable to set this parameter to `0644` so that anyone can look at received facsimile.

```
Protection mode for session logs [0600]?
```

Note that if sensitive information such as credit card access codes are supplied by users that they will appear in the session logs kept by the HylaFAX servers. For this reason the default protection mode for session logs makes them accessible only to the fax user. For ease of debugging however it may be preferrable to select a different mode for the session logs.

```
Protection mode for ttym2 [0600]?
Rings to wait before answering [1]?
```

This parameter is used during inbound call handling. It specifies the number of rings to wait before answering the phone. See the section on inbound call handling for a discussion of the different schemes that HylaFAX supports for handling inbound calls.

```
Modem speaker volume [off]?
Command line arguments to getty program ["-h %l dx_%s"]?
```

This parameter is also related to inbound call handling. It controls whether or not to enable support for inbound data calls and should not be set without first understanding how to setup your system for incoming data usage.

```
Pathname of TSI access control list file (relative to /var/spool/fax) [""]?
```

The TSI access control list file can be used to restrict inbound facsimile usage. The default parameter does not setup a file so any inbound facsimile will be accpted. This parameter is discussed more below.

```
Tag line font file (relative to /var/spool/fax) [etc/lutRS18.pcf]?
Tag line format string ["From %%l|%c|Page %%p of %%t"]?
```

Tag lines are an optional feature supported by HylaFAX whereby each page of an outbound facsimile is marked with a line of text that identifies the sender and possibly more. A proper description of the tag line support is provided in the Tagline Configuration chapter. Note that in the United States some form of identification of the sender of a facsimile is required by law, properly configured tag lines are an acceptable form of identification. Also note that if you specify a tag line font file that does not exist or that is not a proper font file nothing bad will happen; you just won't get tag lines on outbound facsimile. It is ok to initially accept the default parameters and then complete the setup of tag lines at a later time.

```
Time before purging a stale UUCP lock file (secs) [10800]? 30
```

This parameter controls the time that a HylaFAX server process will wait before usurping a UUCP lock file whose owner appears to be gone. The default parameter forces these stale lock files to be left around for 5 minutes; a shorter, more agressive, time can be used on systems where applications that use UUCP lock files are known to be well behaved.

```
Percent good lines to accept during copy quality checking [95]?
Max consecutive bad lines to accept during copy quality checking [5]?
Max number of pages to accept in a received facsimile [25]?
Syslog facility name for ServerTracing messages [""]? local5
```

More parameters associated with inbound facsimile jobs; consult the section below. The syslog facility controls where HylaFAX directs server tracing-related messages. The facility name identifies the syslog facility to direct the messages to. By setting this parameter to a non-standard value HylaFAX messages can easily be recorded in a file separate from the normal system messages. Consult the Troubleshooting chapter, config(4F) and syslogd(1M) manual pages for more information.

```
Set UID to 0 to manipulate CLOCAL [""]?
```

This parameter should only need to be set on Silicon Graphics systems. Consult the section on IRIX-specific guidance for a description of why this parameter might be used.

```
The non-default server configuration parameters are:

CountryCode:            1
AreaCode:               510
FAXNumber:              +1 510 526-8781
LongDistancePrefix:     1
InternationalPrefix:    011
DialStringRules:        etc/dialrules.sf-ba
SessionTracing:         11
RingsBeforeAnswer:      1
SpeakerVolume:          off
GettyArgs:              "-h %l dx_%s"
LocalIdentifier:        +1 510 526-8781
TagLineFont:            etc/lutRS18.pcf
TagLineFormat:          "From %%l|%c|Page %%p of %%t"
QualifyTSI:             etc/tsi
MaxRecvPages:           25
UUCPLockTimeout:        30
LogFacility:            local5

Are these ok [yes]?
```

This completes the collection of server-related parameters; the remaining steps identify and configure the modem for use. If the displayed parameters are unacceptable, typing something other than `yes` or a carriage return will cause faxaddmodem to prompt for new/changed parameter settings.

```
Now we are going to probe the tty port to figure out the type
of modem that is attached.  This takes a few seconds, so be patient.
Note that if you do not have the modem cabled to the port, or the
modem is turned off, this may hang (just go and cable up the modem
or turn it on, or whatever).

Probing for best speed to talk to modem: 38400 OK.

This modem looks to have support for both Class 1 and 2;
how should it be configured [2]?
```

Modems that support multiple classes can be configured to use any supported class. By default faxaddmodem prefers Class 2.0 over Class 2 over Class 1.

```
Hmm, this looks like a Class 2 modem.
Modem manufacturer is "AT&T Paradyne EDC".
```

```
Modem model is "3710-A1-201".
DTE-DCE flow control scheme [xonxoff]?
```

The flow control scheme requested is either `xonxoff` for software flow control or `rtscts` for hardware flow control. Beware of using an improper flow control scheme for the selected tty device. On systems where faxaddmodem understands how tty device names reflect flow control characteristics, selecting a flow control scheme not supported by the device will cause the script to prompt for confirmation and/or to change the device name or the flow control scheme.

```
Using prototype configuration file att-dataport-2...

The modem configuration parameters are:

ModemFlowControl:        xonxoff
ModemHardFlowCmd:        AT&R0\D1\Q3
ModemSoftFlowCmd:        AT&R1\D0\Q1
ModemRate:               38400
ModemRecvFillOrder:      MSB2LSB
ModemSendFillOrder:      LSB2MSB
ModemSetupAACmd:         AT+FAA=1
ModemSetupDCDCmd:        AT&C1
ModemSetupDTRCmd:        AT&D2
Class2CQCmd:             AT+FCQ=1;+FBADMUL=20;+FBADLIN=10
Class2DCCQueryCmd:       "!(0,1),(0-5),(0-4),(0-2),(0),(0),(0),(0-7)"

Are these ok [yes]?
```

The modem-specific configuration parameters are obtained from prototype configuration files that reside in the **config** subdirectory of the HylaFAX spooling area. These parameters have been taken from working systems and should provide a functioning configuration based on the modem type and the selected flow control scheme. If no prototype configuration file exists for a modem then faxaddmodem will prompt for parameters. There are generic prototype configuration files for Class 1, Class 2, and Class 2.0 modems. Because there are many configuration parameters for modems it may be preferrable to use a normal text editor instead of faxaddmodem when constructing a configuration file for an unsupported modem, To do this, simply accept the default parameters and then edit the generated configuration file before starting up a server for the modem. Once a working configuration file is created it is simple to create a prototype file from it; consult the faxaddmodem(1M) or config(4F) manual pages for information on doing this.

```
Creating new configuration file "/var/spool/fax/etc/config.ttym2".
Creating "/var/spool/fax/FIFO.ttym2" in the spooling directory.
Done setting up the modem configuration.
```

At this point faxaddmodem compares the parameters setup for the modem against the parameters setup for the HylaFAX scheduler process. If any parameters have changed it prompts to see if they should be used to create a new configuration file for the scheduler. If no parameters have changed this procedure will not take place.

```
No existing scheduler configuration, let’s do this from scratch.

The non-default scheduler parameters are:
```

```
CountryCode:            1
AreaCode:               510
LongDistancePrefix:     1
InternationalPrefix:    011
DialStringRules:        etc/dialrules.sf-ba
SessionTracing:         11
UUCPLockTimeout:        30

Are these ok [yes]?
Creating new configuration file "/var/spool/fax/etc/config".

Don't forget to restart faxq with "-m ttym2" or configure init
to run faxgetty on ttym2 (if faxq is already running).
```

That's all there is to setting up a basic configuration; or at least *all there should be to it!* Once a configuration file is setup HylaFAX must be notified that a new modem is to be used. Otherwise, if the modem was previously in use nothing needs to be done; the HylaFAX server processes will notice the new configuration files and automatically use their contents.

Starting Outbound Service

Outbound service is carried out by the HylaFAX scheduler process, the faxq(1M) program. There is one faxq process for all modems on a system. The faxq program learns about modems that can be used for outbound jobs in two ways: from arguments supplied on the command line, or from messages received from HylaFAX faxgetty processes that are setup to run on each tty device where a fax modem resides.

Specifying modems on the command line is useful when HylaFAX is to be used in a *send-only configuration*. Doing this however limits the functionality of the scheduler because it will not know the capabilities of the modems it is scheduling. Instead faxq will assume that each modem is capable of anything it is requested to do (e.g. send 2D-encoded facsimile). When the faxq program is able to identify modem capabilities it can do a more effective job of assigning modems to jobs based on the capabilities required by the job.

Modems are specified on the faxq command line using the `-m` option. A modem is identified by the tty device it is attached to. Thus, to start the scheduler up to use two modems, the following might be used:

```
# /usr/etc/faxq -m tty01 -m /dev/tty02
```

Note that devices may be specified with or without a leading **/dev/** prefix. Note that faxq must be started by the super-user and that it will automatically detach itself from the controlling tty. The latter can be defeated for debugging purposes with the `-D` option.

When faxq is used in conjunction with faxgetty no modems need to be specified on the command line. If modems are specified however it does not matter; faxq will just treat the modems as capable of all tasks until it receives information from the faxgetty processes identifying the true capabilities of the modems.

If faxq is being used in a send-only mode and a new modem is added to the system, then faxq must be restarted with new parameters. The faxquit program is best used to stop a running faxq process; it works by sending faxq a ''quit'' message. Note that if faxq is busy processing a job that it will not process the quit request until it is no longer busy with its current tasks. Using faxquit is preferrable to killing faxq with a signal because it insures that faxq will terminate in an orderly fashion. Consult the faxq(1M) manual page for more details on its operation.

Setting up Inbound Service

To setup HylaFAX to provide inbound facsimile or data service a modem configuration file must be setup and a faxgetty program must be started to listen for input on the tty device. The configuration file setup is usually done at the same time that the outbound server is configured; i.e. when faxaddmodem is run. The faxgetty server for the modem should be setup to be run by the init(1M) process according to local system conventions. For System V-based systems this is done by editing the **/etc/inittab** file to spawn faxgetty on the appropriate port. For example, if a modem is to be started on **/dev/ttyf2** the following line might be appropriate:

```
t2:23:respawn:/usr/etc/faxgetty ttyf2    # port 2
```

For systems that use a BSD-style setup the following line might be appropriate for the **/etc/ttyab** file.

```
cua0    "/usr/etc/faxgetty /dev/cua0"   dialup          on
```

Note that faxgetty may be run on a modem port whether or not it is to provide inbound service. By setting the ‘‘rings before answer’’ parameter to zero, a faxgetty will not answer an incoming phone call unless it is explicitly commanded to by the faxanswer(1M) program. This may be desirable if a phone line is used for other than fax and data.

In general it is desirable to run faxgetty whenever possible because faxgetty informs the central HylaFAX scheduler process whenever modems are in use and also identifies the modems’ capabilities. Since faxgetty also does a reliable job of reseting and configuring recalcitrant modems it may even be desirable to run faxgetty on non-fax modems.

The following sections discuss HylaFAX features associated with servicing inbound calls.

Facsimile Service

In normal operation HylaFAX will automatically answer inbound phone calls and receive facsimile. Facsimile reception is done by the faxgetty server process. Inbound facsimile are written to the **recvq** directory in the spooling area on the server machine. Facsimile data is stored as a TIFF Class F (TIFF/F) file and protected according to the **RecvFileMode** configuration parameter specified in the modem configuration file. The maximum number of pages that will be received in a single call can also be controlled with the **MaxRecvPages** configuration parameter. Finally, HylaFAX provides an access control list mechanism for restricting recived facsimile according to the TSI string passed as part of the facsimile protocol; consult the **QualityTSI** configuration parameter in the config(4F) manual page for details.

Data Service

By default HylaFAX does not enable support for inbound data calls. Data service is not enabled so that naive users do not accidentally setup inbound access to their system before proper password controls are in place. To enable inbound data service the modem configuration file must be setup to accept data calls and to invoke the normal system getty program to process the incoming call. Normally this involves enabling the use of *adaptive-answer* functionality and the setup of the **GettyArgs** parameter in the configuration file.

Adaptive answer is the ability for a modem to determine whether an incoming phone call is for data, fax, or voice use. If a modem supports a good adaptive-answering facility then it should be enabled with the **ModemSetupAACmd** and the faxgetty process will automatically service fax, data, or voice calls *as identified by the modem*. Most Class 2 and Class 2.0 modems provide adaptive-answer support and the prototype configuration files that come with HylaFAX automatically enable it if it is provided by the modem. Most Class 1 modems do not provide adaptive-answer support, but HylaFAX provides adaptive-answer support in the server (this is discussed separately).

Setting up the **GettyArgs** parameter requires an understanding of how to invoke the system getty program. HylaFAX will invoke the getty program when a data call is recognized and set up the standard input, output, and error descriptors to point to the appropriate tty device. The getty program should not reopen or reinitialize the modem before doing its work. Some getty programs are incapable of this and are unsuitable for use with HylaFAX. The parameters passed to the getty program must also identify the speed to use to communicate with the local modem. Some getty programs want to automatically detect this rate based on the `CONNECT` message that a modem sends to the host when a connection is established; these programs are unsuitable for use with HylaFAX. A getty program used with HylaFAX must be able to handle a fixed speed for host-modem communication (i.e. the port rate is locked); a speed that is specified on the command line.

For System V-style getty programs the appropriate parameters are typically of the form:

```
GettyArgs:      "-h %l dx_%s"
```

where the `-h` parameter instructs getty to not hangup the device first, the `%l` parameter is translated to the device name (‘‘the tty line’’), and the `dx_%s` parameter identifies the **/etc/gettydefs** entry to use (`%s` is translated by HylaFAX to the speed used to communicate with the modem). Note that the exact parameters to supply depend on the getty program used; consult local documentation to understand what options should be used.

For BSD-style systems, **GettyArgs** is usually of the form:

```
GettyArgs:      "std.%s -"
```

where `std.%s` refers to an entry in the **/etc/gettytab** file for a fixed speed port; e.g.

```
g|std.19200|19200-baud:sp#19200:
h|std.38400|38400-baud:sp#38400:
```

(Note that as before, the ``%s'' is replaced by the speed for host-modem communication.)

Voice Service

[*To be added.*]

Modem Configuration Issues

Beware that when faxgetty processes control a modem they may leave the modem in a state suitable for sending and receiving facsimile. That is, they may leave the modem running in Class 1, Class 2, or Class 2.0. This may have implications for data communication programs such as *tip*, *cu*, and *uucp*. For example, it may be necessary to force the modem into Class 0 (for data communication) when placing a call:

```
AT+FCLASS=0DT<phone number>
```

How things work depends entirely on the contents of the modem configuration file. It is usually a good idea to setup the configuration parameters so that the modem is left idling in Class 0 (for data use) when HylaFAX is not actively using a modem.

Note that when HylaFAX places an outband facsimile call it automatically forces the modem into Class 1, 2, or 2.0 ***before*** *issuing* `ModemDialCmd`*. Thus the old FlexFAX trick of changing the class in the* ***ModemDialCmd*** *parameter should not be used.*

BSDI Guidance

Beware that the standard BSDI FlexFAX distribution places certain server applications such as faxaddmodem in non-standard locations. This can cause confusion when HylaFAX is installed because the old faxaddmodem may be accidentally run and generate incorrect modem configuration files.

IRIX Guidance

On Silicon Graphics Indigo and Indy machines you can not use a Macintosh modem cable to connect your modem to the DIN-8 connector on the back of your host. A Macintosh cable uses a special wiring pattern to pass the RTS and CTS signals between the host and modem. This wiring is not compatible with the wiring used on SGI machines. While it may appear that the the modem and cable work, hardware flow control will not function properly and data will eventually be lost. Consult the serial(7) manual page for an explanation of how to wire up modem cables.

The tty device that is used must reflect whether hardware or software flow control is to be used. Under IRIX, modem devices (i.e. those that monitor DCD) come in two flavors: `ttyf*` devices support RTS/CTS flow control while `ttym*` devices support XON/XOFF flow control. If you want to use hardware flow control to communicate with your modem you should use a `ttyf*` device, otherwise use a `ttym*` device. If you fail to use the correct device you may still get the correct flow control (because later versions of IRIX actually permit flow control to be switched irrespective of the device used), but you are likely to collide with other modem users such as cu, uucp, ppp, and slip that still use the old-style device names (so UUCP lock files may be created for a different name than the one HylaFAX is using).

IRIX 5.2 and 5.3 have a bug in the device driver for the on-board serial ports on IP20 and IP22 systems that causes RTS/CTS flow control to be turned off as a side effect of setting `CLOCAL`. If hardware flow control is to be used, set the **ClocalAsRoot** modem configuration parameter to `No`; see config(4F). Note however that this may cause problems with modems that are quick to drop DCD when carrier is lost because the final status message to the host may not be received before IRIX closes the file descriptor on the server process.

SCO Guidance

The standard SCO serial I/O driver (SIO) does nothing with modem control lines if `CLOCAL` is set on the tty device. The usual workaround is to use the FAS driver instead.

More to be added.

Solaris Guidance

Do not use RTS/CTS flow control with serial ports built around the Zilog ZS8530 chip unless you have patched your system to support hardware flow control when DCD is not asserted to the host (consult the FAQ for an explanation).

Solaris 2.3, and some other SVR4-derived systems, silently truncate or discard syslog messages longer than about 120 characters.

SunOS Guidance

Do not use RTS/CTS flow control with serial ports built around the Zilog ZS8530 chip unless you have patched your system to support hardware flow control when DCD is not asserted to the host (consult the FAQ for an explanation).

Ultrix Guidance

[*Ed: this information comes from Tom Lislegaard* .] Hardware flow control does not work in this release, modems must be configured to use software flow control.

Serial ports are conventionally called **tty00**, **tty01**, etc; see MAKEDEV(8) if you need to create devices.

Advanced server configuration.
HylaFAX table of contents.

ADVANCED SERVER CONFIGURATION

This chapter describes how to configure some of the advanced features of HylaFAX. In addition it discusses how many of the modem-specific configuration parameters work together when sending and receiving facsimile. The set of per-modem configuration parameters provides a flexible mechanism for working around many modem and system limitations. This chapter includes the following sections:

- Local Identifier Support
- Dial String Rules
- Tagline Support
- Adaptive Answer Support
- Caller-ID Support
- Distinctive Ring Support
- Copy Quality Checking
- Continuation Cover Pages
- Time-of-Day Usage Scheduling
- Per-destination Controls
- Automatic Delivery of Received Facsimile
- Rejecting Junk Facsimile
- UUCP Lock Files
- IXO/TAP Support
- Modems that Lie about their Capabilities
- Configuration Parameter Usage During Modem Setup
- Modem Parameter Usage for Outbound Calls
- Modem Parameter Usage for Inbound Calls

Local Identifier Support

The facsimile communication protocols include messages that identify the sending and receiving devices using a 20-character string. The T.30 specification restricts these strings to use only numbers, the ‘‘+’’ symbol, and space. Many fax machines and modem however are capable of accepting a wide range of

ASCII characters. By default HylaFAX will use the canonical form of the local phone number for the local identity when sending or receiving facsimile. This phone number is specified with the per-modem **FAXNumber** configuration parameter. If the **LocalIdentifier** parameter is specified, however, HylaFAX will use it instead. Unlike **FAXNumber** which is converted to a canonical form that conforms to the T.30 specification, the value of the **LocalIdentifier** can be any ASCII string; for example,

```
LocalIdentifier: "Errno Consulting"
```

Beware that not all Class 2 and Class 2.0 fax modems support arbitrary ASCII local identifier strings. If you encounter problems verify your modem supports this functionality by sending it (if it's a Class 2 modem):

```
AT+FCLASS=2
OK
AT+FLID=?
(20)(32-127)
OK
```

The response to the `AT+FLID=?` command should indicate the range of ASCII characters the modem will accept in a local identifier string.

Dial String Rules

A dial string specifies how to dial the telephone in order to reach a destination facsimile machine, or similar device. This string is supplied by a user with each outbound job. User-supplied dial strings need to be processed in two ways by the HylaFAX server processes: to craft a canonical phone number for use in locating the receiver's capabilities, and to process into a form suitable for sending to a modem. In addition client applications may need to process a dial string to formulate an external form that does not include private information such as a credit card access code. Phone number canonicalization and dial string preparation are done according to *dial string processing rules* that are located in a file specified in the server configuration file;

```
DialStringRules: etc/dialrules
```

The generation of an externalized form for a dial string is done by rules that *optionally* appear in `DIR_LIBDATA`**/dialrules** on client machines.

Dial string rules are an important part of HylaFAX as they permit local PTT conventions to be handled entirely on the HylaFAX server machine (e.g. the need to dial ‘‘1’’ before certain numbers but not others). Client applications are then free to use a canonical format for phone numbers that is location independent. Dial string rules combined with the **ModemDialCmd** also permit modem-specific idiosyncrasies such as the syntax for doing a ‘‘flash hook’’ to be handled transparently. Finally, dial string rules can be used to provide dialing *aliases* such as mapping ‘‘a-zA-Z’’ to the corresponding numeric codes.

Consult dialrules(4F) for detailed information.

Tagline Support

HylaFAX includes support for ‘‘*tag lines*’’; a line of text imaged at the top of each outgoing facsimile page.

Tag lines imaging requires the specification of:

- a format string,
- the name of a file containing an uncompressed Portable Compiled Font (PCF) from the X11 Window System, and
- the host bit order.

When tag lines are enabled the fax server will image a line of text across the top of each outgoing facsimile page using the specified tag line format string and font. This procedure is done *on-the-fly*, just like a fax machine does it. However, because the fax server has lots of information about the outbound facsimile job, the imaged tag line may be setup to display a large variety of information.

A 100 dpi 18point Lucida Typewriter font from the X Window System is included in the HylaFAX distribution for use in imaging tag lines. Experiments indicate this is a reasonable font to use. You can however use an alternate font of your liking.

On Silicon Graphics systems a tag line font can be installed by converting one of the compressed font files provided with the standard X server; e.g.

```
% zcat /usr/lib/X11/fonts/lutRS18.pcf.Z > /var/spool/fax/etc/lutRS18.pcf
```

To enable use of the font and imaging of the tag lines setup the **TagLineFont** configuration parameter:

```
TagLineFont: etc/lutRS18.pcf
```

If font files are stored on your system in an uncompressed format then you can reference the file directly using an absolute pathname. Relative pathnames must be specified with respect to the top of the spooling area. If the **TagLineFont** parameter references a non-existent file or a file that does not contain a valid PCF font then tag line support will be disabled and a message will be logged through syslog.

The default tag line format string is ‘‘`From %%n|%c|Page %%p of %%t`’’ which prints three fields containing the phone number of the server, the date and time of the outgoing job, and the page number and total pages for the job. Consult the **TagLineFormat** parameter description in config(4F) for information on configuring a different format string.

The tagtest(1M) program may be used to try out different tag line formats and fonts before they are configured for use by the server.

Finally, be certain the host bit order is configured to reflect the order of bits on your machine. The host bit order should be properly setup at the time the configure script is run. If the server has the wrong host bit order then tag lines will be imaged incorrectly.

Adaptive Answer Support

Adaptive answer is the ability for a modem to determine whether an incoming phone call is for data, fax, or voice use. If a modem supports a good adaptive-answering facility then it should be enabled with the **ModemSetupAACmd** and HylaFAX will service fax, data, or voice calls *as identified by the modem* and specified in the modem configuration files. For example,

```
ModemSetupAACmd: AT+FAA=1
```

Beware however that for some modems adaptive-answer support works properly only when the modem is left in Class 2 or Class 2.0; this may be a problem if you want to configure the modem to ‘‘idle in Class 0’’ to avoid confusing naive programs that make use of the modem for outbound calls.

The adaptive-answer algorithm used by some modems can confuse some fax machines and/or data modems. If you do not intend to service both data and fax calls you may not want to configure adpative-answer for inbound calls.

If a modem does not support adaptive-answering, then there are several options. If the modem is a Class 1 modem, then HylaFAX provides a simple adaptive-answering strategy whereby incoming calls are first answered as if they are for a fax machine and, if that fails, then answered as if they are for a data modem. This facility is enabled by specifying something like:

```
AdaptiveAnswer:          yes              # enable adaptive answer
AnswerRotary:            "fax data"       # answer for fax, then data
ModemAnswerCmd:          AT+FCLASS=1;A    # default is to answer as fax
ModemAnswerDataCmd:      ATH+FCLASS=0;A   # hangup and answer as data
Class1RecvIdentTimer:    10000            # timeout fax answer in 10 secs
```

in the configuration file. The above lines cause the fax server to do the following in response to an incoming phone call:

1. Issue "`AT+FCLASS=1;A`" to answer the phone call in Class 1; i.e. as a fax machine (issuing CNG tones).
2. Send TSI and DIS frames as required by the fax protocol.
3. Wait for DCS from the caller (if it is a fax machine).
4. Timeout waiting for DCS in 10 seconds (or whatever is specified for `Class1RecvIdentTimer`).
5. Issue "`ATH+FCLASS=0;A`" to hangup and then re-answer the phone in Class 0; i.e. as a data modem.

This technique assumes many things about the capabilities of the modem and the local telephony service and may not work for all Class 1 modems or for all locales.

Note also that by reversing the order of the items specified in the `AnswerRotary` parameter you can get HylaFAX to answer first for data and then for fax. If calls are answered first as data, then you may need to constrain the timeout used to recognize a data call so that time still remains to setup a fax connection. Consult your modem manual and the `ModemAnswerResponseTimeout` configuration parameter.

A second facility supported by the fax server in lieu of adaptive answering is a *"rotary of answering techniques"*. The general idea is that a list of alternative ways to answer the phone is supplied and the server will rotate through the list on *consecutive inbound calls* until it finds one that works. For example, one might specify something like:

```
AnswerRotary: "fax data"
```

which would instruct the server to answer incoming calls as if they were from a fax machine until a call was received from a data modem, in which case it would then answer subsequent calls as a data modem until a non-data call was received (in which case it would go back to fax). The rotary list can have up to three items, with items being selected from one of: *fax*, *data*, *voice*, and *any* (answer a call of an unknown type). The *voice* answering request is reserved for future development. Finally, in conjunction with the rotary answer facility there is an `AnswerBias` parameter that can be used to specify an index into the rotary list to use after *successfull* calls. In the above example, this parameter can be used, to force calls to always be answered first as data by specifying:

```
AnswerRotary: "fax data"
AnswerBias:       1
```

Note that the adaptive-answer and rotary answer facilities differ mainly in whether the rotary of answering techniques is applied to a single call or to consecutive calls.

Caller-ID Support

Caller-ID is a feature provided by phone companies whereby information about a calling party is passed to the receiver *prior* to answering a call. This facility is not universally available and not all modems are capable of processing the provided signals. For modems that are capable of processing caller-ID information HylaFAX provides the following support.

- Any caller-ID information received by a faxgetty process is recorded in trace logs provided the **ServerTracing** configuration parameter is set to enable the collection of server messages (i.e. bit 0).
- Inbound calls may be accepted or rejected according to an access control list specified by the **QualifyCID** configuration parameter.

Note that HylaFAX parses caller-ID information according to two configuration parameters, **CIDName** and **CIDNumber**, that specify how to identify a caller’s name and phone number (if provided by the modem). For example, for a ZyXEL U1496 modem, the appropriate configuration parameters to use for caller-ID are:

```
CIDNumber:      "CALLER NUMBER: "       # pattern string for phone number info
CIDName:        "CALLER NAME: "         # pattern string for identity info
```

Consult config(4F) for a full description of the caller-ID-related configuration parameters.

Distinctive Ring Support

Distinctive ring is a feature provided by phone companies whereby multiple phone numbers can be directed to a single phone line with different ring patterns for each phone number. This is a simple mechanism that can be used to distinguish between incoming fax, voice, and data calls (i.e. all fax calls are directed to one number, voice to a different number, and data to a different number).

HylaFAX supports distinctive ring capabilities through the **RingFax**, **RingData**, and **RingVoice** modem configuration parameters. Modems that support distinctive ring send a different `RING` status message to the host depending on the ring pattern presented by the phone company. By setting the configuration parameters faxgetty will match the appropriate `RING` status message and use the information to treat the inbound call as fax, data, or voice before answering the call. For example,

```
RingFax:        RING1          # treat ring pattern 1 as fax
RingData:       RING2          # treat ring pattern 2 as data
```

Copy Quality Checking

HylaFAX will automatically check the quality of received facsimile and regenerate rows of data that have an error in them. This facility is termed ‘‘*copy quality checking*’’ and is only done if the modem is incapable of doing this task itself, or if the software is configured so that modem copy quality checking is disabled.

To configure copy quality checking in a Class 2 or Class 2.0 modem, when supported, setup the **Class2CQCmd** parameter for the modem. For example,

```
Class2CQCmd: AT+FCQ=1;+FBADMUL=5;+FBADLIN=10
```

This enables copy quality checking for receiving and sets the copy quality parameters so that acceptable page quality requires that no more than 10 consecutive bad rows of data may be present and less than 5% of the rows in a page have errors in them. To disable copy quality checking in the modem use

```
Class2CQCmd: AT+FCQ=0
```

If a modem indicates that it supports copy quality checking but **Class2CQCmd** has not been specified then HylaFAX will note this when the modem is prepared for use.

Copy quality checking in the server is controlled by two configuration parameters: **PercentGoodLines** and **MaxConsecutiveBadLines**; see config(4F) for more information. These parameters define the criteria by which the server decides if a received page has acceptable copy quality. Pages that are deemed unacceptable are rejected and the sender is told to resend the page.

Beware that many fax machines are incapable of resending pages that have unacceptable copy quality because they do not buffer a full page image.

If you encounter problems with the copy quality checking support you can disable it by setting either the **PercentGoodLines** or **MaxConsecutiveBadLines** configuration parameters to 0; e.g.

```
PercentGoodLines:        0        # disable copy quality checking
```

Note however that doing this means you may receive facsimile that are unreadable.

Continuation Cover Pages

A *continuation cover page* is a cover page automatically generated by HylaFAX when retransmitting an outbound job after a protocol error abnormally terminated a previous transmission. This cover page is only generated when a user-submitted cover page has already been transmitted; it is intended to identify the sending system and provide information about why the retransmit is being done. HylaFAX generates the cover page using the parameters supplied when the job was submitted. In addition it provides a *comments* section that identifies the transmission as a continuation of a previous job and describes why the previous attempt to transmit failed (this may be useful to the receiver).

By default HylaFAX does not enable continuation cover pages. To enable use of this feature the **ContCoverPage** configuration parameter must be set to the pathname of a cover sheet template file to use in generating continuation cover pages. This cover page template file is passed to the mkcover(1M) script, or to the program specified with the **ContCoverCmd** configuration parameter; consult the mkcover(1M) documentation for a description of this file.

Time-of-Day Usage Scheduling

HylaFAX can be configured so that outbound calls are placed only during certain hours of the day. This can optionally be configured on a per-destination basis, making it easy to reduce phone costs by processing toll calls only during the time when off-peak rates are available.

By default outbound calls are permitted at any time. To constrain calls to a particular time or range of times the **TimeOfDay** parameter can be setup according to the syntax specified in config(4F). This syntax is consistent with the syntax used by the UUCP software. For example, to permit outbound calls only between 9AM and 5PM local time the following might be used:

```
TimeOfDay:      "0900-1700"
```

Per-destination Controls

HylaFAX supports many controls on the operation of the scheduler and related software. For example, outbound jobs may be restricted so that processing is only done at certain times of the day. All of these parameters can be specified in the configuration file used by the central HylaFAX scheduler process. Parameters specified in this way apply to *all outbound jobs*. It is also possible to specify these parameters on a *per-destination basis* using the **DestControls** configuration parameter. This parameter specifies the pathname of a file that holds per-destination control parameters; destctrls(4F).

Per-destination controls are especially useful for controlling which phone numbers can be called. For example, by specifying an entry of the form:

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

any outbound job that would cause HylaFAX to phone 911 would automatically be rejected with the submitter notified by electronic mail using the specified message.

Another useful feature of this facility is the ability to delay toll calls to off-hours so that off-peak phone rates may be used. Entries of the form:

```
^[+]1415.*      TimeOfDay = "Any"
^[+]1510.*      TimeOfDay = "Any"
.*              TimeOfDay = "Wk1700-0830,Sat,Sun"
```

might be used to permit calls to US area codes 415 and 510 at any time of day, but force all other calls to be done only during off hours. Note that the parameters specified in the configuration file serve as the *default values* to use when scheduling an outbound job. That is, if a parameter value is not obtained from the destination controls file then it is taken from any parameters specified in the configuration file. For example, if the above rules had not included the last line that matched ‘‘`.*`’’, then the time-of-day to use for long-distance calls would have been that specified in the configuration file (by default `Any`).

Automatic Delivery of Received Facsimile

It is desirable to deliver inbound facsimile directly to receipients. HylaFAX routes inbound facsimile through the faxrcvd(1M) shell script. This script may be tailored to use local knowledge and/or tools to route and deliver inbound facsimile. In particular, if facsimile are received from well-known senders and always directed to a specific person faxrcvd can be tailored to automatically dispatch these facsimile by electronic mail; consult the manual page for details. Services such as Direct Inward Dial (DID) are not supported because the required fax modems use non-standard programming interfaces. In the future, when updated versions of the ITU facsimile protocols are supported in commercial fax modems HylaFAX will be able to use routing information provided in the protocol to do automatic delivery of inbound facsimile.

Rejecting Junk Facsimile

By default HylaFAX will accept inbound facsimile from any caller. If the **QualifyTSI** configuration parameter is setup then HylaFAX will receive inbound facsimile only if the caller sends an acceptable Transmitting Subscriber Identification (TSI) string; a string that uniquely identifies who the sender is. HylaFAX determines which TSI are acceptable by using the rules given in the file specified with **QualifyTSI**. This mechanism provides a way to automatically reject *junk facsimile*. Note however that because a TSI is not authenticated in any way it is a simple matter for a caller to masquerade as an acceptable sender and bypass the access control mechanism. Consult tsi(4F) for complete details of this facility.

UUCP Lock Files

HylaFAX implements shared inbound/outbound usage using UUCP lock files. In normal operation each modem has a faxgetty process listening for data from the modem. When an inbound call is received the modem emits a `RING` status message that wakes up the appropriate faxgetty process. faxgetty then checks the UUCP lock file for the port to make sure the port is not busy for outbound use. If the port is busy (i.e. a UUCP lock file is present), then faxgetty will stop listening on the port and wait for the outbound use to terminate. If the port is not busy, then faxgetty will answer the call according to the parameters specified in the configuration file.

For this scheme to work outbound callers must install a UUCP lock file before using a modem. This lock file must be created using the same conventions understood by HylaFAX. In particular the lock file name must be consistent and the file contents must include the process ID of the lock file owner written either as a binary or ascii value according to the system-specific conventions.

HylaFAX supports several schemes found on different systems. The appropriate scheme is normally setup according to the target system at the time that HylaFAX is configured for building. It is possible, however, to override these parameters through the runtime configuration files; consult the config(4F) manual page for information about the **UUCPLockType** configuration parameter.

In addition to the lock type there are configuration parameters that control the protection mode of the lock file (**UUCPLockMode**), the directory in which lock files are created (**UUCPLockDir**), and a timeout parameter used to purge *stale lock files*. Consult config(4F) for a full description of these parameters.

IXO/TAP Support

Experimental support is included for sending messages to alpha-numeric pager devices using the IXO/TAP protocol. This protocol is implemented with the sendpage(1) client program and the pagesend(1M) server program that does the actual delivery.

There are several configuration parameters associated with the IXO/TAP support. The most important parameter is **PagerSetupCmds** which is a set of commands to send to the modem to configure it for a call to the pager service provider. This command should usually be defined to constrain the modem to connect at 1200 baud using V.22 (and no error correction protocol). Other parameters are related to the IXO protocol implementation and should not need to be changed; consult config(4F) for full details.

Other than the per-modem configuration parameters described above, the only other pager-related information that may need to be setup is information in the info database to constrain the maximum length of an alpha-numeric pager message and any password string that must be sent to the service provider during the initial login sequence. Consult info(4F) for information on **pagerMaxMsgLength** and **pagerPassword**.

Modems that Lie about their Capabilities

Class 2 and 2.0 fax modems all too frequently have bugs in their firmware. Frequently it is possible to workaround firmware problems by restricting a modems' capabilities. To do this the **Class2DCCQueryCmd** configuration parameter may be set in the per-modem configuration file. By default this parameter holds the command to send the modem to query its capabilities. However if the query string has a leading ''!'' then no command is sent to the modem and instead the remainder of the string is treated as the response returned by the modem. Thus to avoid using a particular modem capability all you need to do is setup a capabilities string that has a restricted set of modem capabilities.

Capabilities are returned using a syntax specified in the Class 2/2.0 specification:

```
(vr),(br),(wd),(ln),(df),(ec),(bf),(st)
```

where,

`vr` indicates which vertical resolutions are supported,
`br` indicates which signaling rates are supported,
`wd` indicates which page widths are supported,
`ln` indicates which page lengths are supported,
`df` indicates which data formats are supported,
`ec` indicates whether or not the optional Error Correction Mode (ECM) is supported,
`bf` indicates whether or not the optional Binary File Transfer (BFT) protocol is supported, and
`st` indicates which minimum scan times are supported.

Each parameter is specified as single value, range of values (e.g. `0-3`), or list of values (e.g. `0,1,2`). The values are given in the Class 2/2.0 specification and can also be found in the source distribution in the file **faxd/class2.h**.

An example use of this mechanism is found in the prototype configuration file for the AT&T DataPort modem. The rev C01.66.10 firmware for the DataPort returns an invalid string for the `AT+FDCC=?` query command so the prototype configuration file supplies a valid one instead:

```
Class2DCCQueryCmd:      "!(0,1),(0-5),(0-4),(0-2),(0),(0),(0),(0-7)"
```

Beware of specifying that a modem has capabilities that it does not; HylaFAX is likely to try and make use of them!

Configuration Parameter Usage During Modem Setup

There are many configuration parameters that control the operation of HylaFAX in resetting and initializing a modem. This section presents these parameters and explains how they fit together in normal operation. By understanding how these parameters are used it is possible to control many aspects of the operation of HylaFAX.

In normal operation HylaFAX will first *reset* a modem and then prepare it for use with a *setup procedure*. Modem reset involves the following steps:

1. Lower the Data Terminal Ready (DTR) signal, pause **ModemResetDelay** milliseconds, and then raise DTR. DTR is used to force a modem reset rather than sending a command such as `ATZ` to the modem because many modems can get confused and enter a state where commands sent from the host are ignored. Instead HylaFAX uses a hardware control signal to force the modem to do a reset operation. This usage requires that a modem respond to a DTR transition by resetting itself. Since modems all take different amounts of time to do a reset operation the time that HylaFAX waits for the modem to complete the reset operation is programmable. A certain Hayes Technical bulletin claims one should wait 2.6 seconds for a modem to complete a reset operation. In practice this is much longer than most modems need and this parameter can be reduced with a significant speedup in the overall setup process.
2. Once the modem has been reset (assuming the modem monitors DTR and does the ‘‘right thing’’ in response to DTR being lowered), HylaFAX sets up the baud rate and flow control parameters for the tty device using the **ModemRate** and **ModemFlowControl** parameters.
3. HylaFAX then flushes any data received from the modem since the reset and sends the following sequence of commands:

```
ModemResetCmds          any additional reset commands
ModemEchoOffCmd         disable modem echo of commands
ModemVerboseResultsCmd  enable verbose result codes
ModemResultCodesCmd     enable transmission of result codes
ModemNoAutoAnswerCmd    disable modem from auto-answering calls
ModemOnHookCmd          place modem ‘‘on hook’’
ModemPauseTimeCmd       configure delay for "," in dial string
ModemWaitTimeCmd        configure time to wait for carrier
modemflowcontrolcmd     establish DTE-DCE flow control scheme
ModemSetupDTRCmd        force desired modem response to DTR transition
ModemSetupDCDCmd        force desired modem response to DCD transition
```

   Note that these commands are sent as two separate `AT` commands, broken between **ModemOnHookCmd** and **ModemPauseTimeCmd**; this is done to avoid problems with certain modems.

The reset sequence must be successful for HylaFAX to consider the modem in an acceptable state and to proceed with the subsequent steps:

1. Query the modem using **ModemClassQueryCmd** to verify it supports the functionality specified by the **ModemType** (this parameter can be used to control which class to use when a modem supports multiple classes).
2. Set the modem into the appropriate class using **Class*X*Cmd**, where ***X*** is one of 0, 1, or 2, depending on the application.
3. Identify the manufacturer, model, and firmware revision information using **ModemMfrQueryCmd**, **ModemModelQueryCmd**, and **ModemRevQueryCmd** parameters. This information is obtained early in the setup procedure in case special-case handling is required for the modem. Note also that for Class 1 modems it is typically not possible to query the modem for some of this information and it must instead be ‘‘hard-coded’’ in the prototype modem configuration file that is selected according to the product ID code returned by the `ATI0` command.
4. Identify the modem capabilities using the **ModemDCCQueryCmd** for Class 2/2.0 modems, or according to the capabilities of the Class 1 driver and the result of `AT+FTM=?` and `AT+FRM=?` commands for Class 1 modems.
5. For Class 2/2.0 modems: establish whether the modem supports copy-quality checking and polled retrieval of documents; both these capabilities are automatically provided by the driver for Class 1 modems. If the modem supports copy quality checking, according to the result of the **Class2CQQueryCmd** and copy-quality setup commands are provided with the **Class2CQCmd** parameter, then the modem is setup accordingly.

Modem initialization is then carried out by sending the following commands to the modem:

- *Class 2/2.0 modems.*

```
Class2Cmd              force the modem into Class 2/2.0
class2FlowControlCmd   establish DTE-DCE flow control scheme
Class2TBCCmd           select stream modem host-modem communication
Class2BORCmd           set the bit order for HDLC and page data
Class2PHCTOCmd         set the timeout during page transfer (Phase C)
Class2CQCmd            enable copy quality checking
Class2NRCmd            enable negotiation reporting (Class 2.0 only)
Class2PIECmd           disable program interrupt handling (Class 2.0 only)
Class2BUGCmd           enable HDLC frame reporting if requested
Class2DCCCmd           initialize modem capabilities
```

  optionally followed by the following additional commands when the modem is to be configured to (possibly) receive data:

```
Class2RELCmd           enable reception of byte-aligned EOL codes
Class2CRCmd            enable facsimile reception
Class2LIDCmd           set local station identifier
ModemSetupAACmd        enable adaptive-answer support
```

- *Class 1 modems.*

```
Class1Cmd              force the modem into Class 1
class1FlowControlCmd   establish DTE-DCE flow control scheme
ModemSetupAACmd        enable adaptive-answer support
```

Flow control is explicitly set separately for Class 0 (data) operation and Class 1, 2, 2.0 (fax) operation.

To configure the use of a different flow control scheme in each class simply override the default definition of the **Class*X*Cmd** parameter; e.g.

```
Class2Cmd: AT+FCLASS=2<xon>
```

Note however that this will only work for outbound usage; doing something similar for inbound processing is more complicated because the modem, rather than the host, decides when to switch between Class 0 and Class 1, 2, 2.0. Consequently HylaFAX must *react to events rather than initiate them*. This work is also complicated by the fact that many modems do not accept commands (or are confused by commands) from the host during certain critical sections of inbound call processing.

Modem Parameter Usage for Outbound Calls

An outbound facsimile job is broken up into *phases* that correspond roughly to the phases defined in the CCITT T.30 facsimile protocol:

- call placement
- selection of station session capabilities
- per-page capabilities negotiation
- page transfer
- page acknowledgement

The last three phases may be repeated multiple times depending on the number and characteristics of the facsimile being transmitted.

To send a facsimile a phone call must be placed. HylaFAX takes the following actions:

1. Lock the modem.
2. Initialize the modem for outbound facsimile use.
3. Instruct the modem to place the phone call.

The corresponding set of commands sent to the modem are:

- *Class 2/2.0 modems.*

```
Class2Cmd              force the modem into Class 2/2.0
class2FlowControlCmd   establish DTE-DCE flow control scheme
Class2TBCCmd           select stream modem host-modem communication
Class2BORCmd           set the bit order for HDLC and page data
Class2PHCTOCmd         set the timeout during page transfer (Phase C)
Class2CQCmd            enable copy quality checking
Class2NRCmd            enable negotiation reporting (Class 2.0 only)
Class2PIECmd           disable program interrupt handling (Class 2.0 only)
Class2BUGCmd           enable HDLC frame reporting if requested
Class2DCCCmd           initialize modem capabilities
Class2LIDCmd           set local station identifier
Class2SPLCmd           request polling status (if a polled receive is to be do
Class2DDISCmd          setup initial session parameters (optional)
```

```
ModemDialCmd           place the call
```

- *Class 1 modems.*

```
Class1Cmd              force the modem into Class 1
class1FlowControlCmd   establish DTE-DCE flow control scheme
ModemDialCmd           place the call
```

Note that the modem is always switched into the appropriate class and flow control scheme before placing the call. For Class 2 modems that do not properly implement the `AT+FDIS` command the **Class2DDISCmd** parameter must be configured so that HylaFAX will setup initial session parameters *before* placing the phone call.

If the call completes and a facsimile transmit session is started then HylaFAX will send **ModemSendBeginCmd** to the modem. This parameter can be used to enable servicing that must be done only while Carrier Detect (CD) is asserted to the host. For example, on Sun systems it is not possible to enable RTS/CTS flow control on serial ports implemented with the ZS8530 unless CD is asserted; to workaround this problem for outbound calls one can use:

```
ModemSendBeginCmd:      "<rts>"
```

to force HylaFAX to enable RTS/CTS flow control on the host tty device after carrier is established. Beware however that some modems raise and lower CD during Class 1 operation and this can cause problems for systems with restrictions of this sort.

The next phase in the processing of an outbound job is the selection of station capabilities. The receiver is required to transmit a series of messages that provide its identity and capabilities (e.g. whether it can accept 2D-encoded facsimile data). HylaFAX examines the received capabilities and compares them to the capabilities required for the job. If the receiver is capable of accepting the documents to be sent then HylaFAX proceeds with the transmission. Otherwise HylaFAX will disconnect and, either re-prepare the outbound documents according to the receiver's capabilities or reject the job because it cannot be sent.

Page transfer requires the selection of session capabilities that correspond to the page to be transferred and the actual transfer of the page data to the modem. HylaFAX handles documents with varying page characteristics. In particular this means that documents with a mixture of high res and low res data (e.g. a low res cover page followed by high res image or text) are handled transparently.

For Class 2 modems session parameters are set using the **Class2DISCmd** (note that this is not the same as **Class2DDISCmd**). If the negotiated session parameter needs to be changed then this command will be sent to the modem. The Class 2 specification requires that a modem accept this command at any time prior to an `AT+FDT` command. However because some modems do not properly implement this part of the specification the **Class2DDISCmd** is provided to workaround modem limitations. Beware however that modems that require **Class2DDISCmd** may not be capable of renegotiating session parameters at all; this can be an annoying problem.

For Class 2/2.0 modems, page data is transferred using the `AT+FDT` command. This command instructs the modem that a page of facsimile data will follow. The modem is released to negotiate session parameters (as needed) and HylaFAX then waits for a response that indicates page data may be transferred to the modem. Class 2 modems indicate they are ready for page data by sending `CONNECT` followed by XON (DC1) to the host. (The **ModemWaitForXON** configuration parameter controls whether or not HylaFAX should wait for XON because some ‘‘Class 2 modems’’ do not follow this aspect of the spec.) Class 2.0 modems indicate they are ready for page data by sending a `CONNECT` response to the host (but no XON). Class 2 modems return `OK` to the host when they have accepted all the page data. Class 2.0 modems do not return `OK`.

After a page of data has been transferred to the modem HylaFAX indicates whether this is the last page to be sent and/or whether more documents are to come. A *post-page message* is then sent and a post-page response is awaited. Class 2/2.0 modems handle this exchange entirely within the modem. For Class 1 modems HylaFAX implements this part of the protocol on the host: the post-page message may be retransmitted as many as three times. A post-page response indicates whether a page was successfully received and/or whether or not the receiver wants to resynchronize the line due to a perceived loss in signal quality. HylaFAX is capable of retransmitting pages that a receiver deems unacceptable. HylaFAX will automatically retransmit a page up to three times if it does not receive a proper acknowledgement from the receiver.

Modem Parameter Usage for Inbound Calls

Inbound call handling is the most complicated part of the HylaFAX software. This is because modems vary so widely in the way they process inbound calls for fax, data, and voice use. Also, inbound calls typically require that a modem be configured *before* a call is received and many modems do not accept commands from the host until after a call has been recognized, accepted, and processing initiated.

HylaFAX handles inbound calls in the **faxgetty** program. The modem is initialized as described above and faxgetty then waits for a ring status message from the modem. On receiving input from the modem the following processing takes place:

1. The modem is locked. If the modem cannot be locked because there is an outbound user in control of the modem then it is released and **faxgetty** will wait for the lock file to be removed.
2. Wait for **RingsBeforeAnswer** rings, waiting at most 5 seconds for each ring status message. If distinctive ring support is configured (**RingFax**, **RingData**, and **RingVoice**) then the ring status messages are checked and the type of call is potentially deduced. If Caller-ID support is configured (**CIDName** and **CIDNumber**), then any Caller-ID information returned by the modem is parsed.
3. If the appropriate number of rings was not received, then the line is hung up and the modem is reset and initialized.
4. Check any Caller-ID information. If **QualifyCID** is configured and the caller is not acceptable, then reset and initialize the modem.
5. If the call type is already known from distinctive ring information, then the call is processed immediately. If faxgetty was instructed to answer any type of call, then the call is answered as appropriate. However, if a particular type of call was to be processed (e.g. data) and the deduced type of call does not match, then the call is rejected and the modem is reset and initialized.
6. If the call type is unknown then the call is answered according to the the current type specified by **AnswerRotary** and **AnswerRotor**.
7. If the call is not successfully answered and if **AdaptiveAnswer** is enabled, the next item in the

AnswerRotary list is used to immediately re-answer the call. This is repeated until a call is successfully answered or the list of choices in **AnswerRotary** is exhausted.

Call answering is done as follows:

1. Send the modem one of **ModemAnswerDataCmd**, **ModemAnswerFaxCmd**, and **ModemAnswerVoiceCmd** according to whether the call is to be answered as data, fax, or voice, respectively. If any of these are not set then **ModemAnswerCmd** is used instead.
2. Wait for a response from the modem that signifies carrier has been established and which identifies the type of call. The exact set of responses is given in config(4F). If **ModemWaitForConnect** is enabled, then HylaFAX will read responses from the modem until a `CONNECT` response is sent to the host or an error is detected.
3. If a call is successfully answered, then one of **ModemAnswerDataBeginCmd**, **ModemAnswerFaxBeginCmd**, and **ModemAnswerVoiceBeginCmd**, is sent to the modem according to the call type deduced above. This mechanism is useful for modems that automatically change DTE-DCE communication to a fixed line speed or flow control setting for inbound facsimile calls.

At this point the call has been answered, carrier established, and the call type is known.

- Data calls are handled by invoking a system getty program; but only if the **GettyArgs** parameter is set.
- Voice calls are handled by invoking a system vgetty program; but only if the **VGettyArgs** parameter is set.
- Facsimile calls are handled directly in faxgetty.

For a data or voice call faxgetty forks and execs the appropriate program. The UUCP lock file is setup so that it is owned by the child (important for SLIP and PPP) and the parent handles cleanup of various resources when the child process terminates.

For fax operation faxgetty drops into the facsimile receive protocol.

To be added.

How to operate a HylaFAX server.
HylaFAX table of contents.

CLIENT SETUP AND OPERATION

HylaFAX is based on a client-server architecture. A single fax server machine with one or more fax modems may service a network of client machines that do not have fax modems. Client and server machines typically communicate using a special HylaFAX client-server protocol that is transported on top of the TCP/IP protocols. Alternatively, clients can be setup to submit facsimile using other transport mechanisms such as electronic mail or a network printing service. This chapter discusses only the setup of a UNIX system as a client machine that uses the standard HylaFAX client applications to submit and manage outbound facsimile jobs. Building a mail-to-fax gateway is discussed in a separate chapter. You are on your own if you want to use a printer protocol to hook clients up; though it is not much more complicated than the mail route (you should look at how the mail support is done).

From the IRIX Binary Distribution

When installing HylaFAX from the IRIX binary distribution format, client machines can be setup simply by installing the `hylafax.*.client` images:

```
% inst -f dist/hylafax
...
inst> install hylafax.*.client
inst> go
```

(actually, the default packages to install should be the ones needed to setup a client machine). If you do not have the font metric files installed on the client (they are part of the **dps_fonts** image) you will also want to install `hylafax.sw.afm`.

From the Source Distribution

There is no direct support for building client-only systems or distributions from the source distribution. The two simplest ways to setup a client-only distribution are by packaging the appropriate client pieces together with something like tar(1) and then ‘‘dropping the bits’’ onto the client machine, or by mounting the `DIR_BIN`, `DIR_LIBDATA`, and `DIR_LIBEXEC` directories from a system where the software has already been installed. (`DIR_BIN`, `DIR_LIBDATA`, and `DIR_LIBEXEC` refer to target-specific pathnames to the directories where binary applications, library data, and library executable programs are placed, respectively). The exact set of files provided for a HylaFAX client is specified in the hylafax(1) manual page (see the FILES section).

Setting up the ‘‘fax’’ Service

With the applications and data files available on a client machine, the only other task to perform on the client machine is to setup an entry in the **/etc/services** file or equivalent YP/NIS database for the ‘‘fax’’ service. This work is automatically done on the server machine by the faxaddmodem script using a line of the form:

```
fax     4557/tcp                # FAX transmission service
```

On the client machine however you will want to set this up by hand unless the the system is configured to use YP/NIS and the fax service is registered globally through YP/NIS. Note that if no service entry exists for the ‘‘fax’’ service then a default builtin port number is used. If the HylaFAX client-server software is configured to communicate on a port other than the default port, then the server must be registered.

Setting up Server Access

Once a client machine is setup for use the server machine may need to be configured to permit jobs to be submitted. Specifically, the file **etc/hosts** in the spooling area on the server machine must be setup to permit each client machine access. Consult the hosts(4F) manual page for information on how this is done.

HylaFAX table of contents.

HylaFAX Server Operation

In normal operation HylaFAX server machines should start the scheduler process when the system is booted. If faxgetty processes are to be used, they should be setup so that the init(1M). process will start them on each port where a fax modem resides. Under IRIX the normal installation process sets up the appropriate configuration information for the faxq process to be started up by init(1M). On other systems you will need to arrange for this yourself (typically by editing **/etc/rc.local** or similar). faxgetty processes must be manually setup by editing the appropriate control file for **init**; usually either **/etc/inittab** or **/etc/ttytab**.

If you need to start the faxq server by hand, consult the faxq(1M) manual page and the shell script **etc/hylafax** that is used under IRIX.

Incoming facsimile are placed in the **recvq** subdirectory of the spooling area and probably will need to be cleaned up periodically. Likewise there is logging information in the **log** subdirectory and accounting information in the **etc** subdirectory of the spooling area that may need some attention.

The faxcron script is designed to handle the periodic maintenance of the spooling area. This script is designed to be run from cron(1) every day.

If you want to do accounting check out the xferstats and recvstats scripts for a basic attack on how to process the etc/xferlog accounting file that has records of all facsimile transmissions and receptions. Note that faxcron uses these scripts to deliver statistics about facsimile there were recently transmitted and received.

Otherwise the only matter to be concerned with is the support for data connections. If your modems are capable of differentiating data connections from facsimile connections the fax server can be configured to startup a getty program whenever a call from a data modem is received. Alternatively, if your modem does not support an adaptive-answer facility, but it is a Class 1 modem, the server may be able to do adaptive answering in software. In any event, beware that if you enable data connections you should take the normal precautions you would take when there are dialup ports on your machine. Specifically, make sure that you have passwords, appropriate file protections, and proper configuration of *uucp* or similar.

If you encounter problems sending or receiving facsimile you can enable copious tracing information by editing the HylaFAX configuration files. Consult the Troubleshooting section and the config(4F) manual page, in particular, for help. There is also information in the HylaFAQ on problems one might encounter during server operation.

HylaFAX documentation.
HylaFAX table of contents.

TROUBLESHOOTING

This section contains tips for dealing with the most common problems encountered when setting up and running the software. The following subsections cover specific areas:

- Client Basics
- Client Access Control Problems
- Server Basics
- Queue Management
- PostScript Document Preparation
- Communication Problems
- Getty Problems
- Problems With UUCP, cu, tip, etc.

You can also consult the HylaFAQ for answers to common questions.

TROUBLESHOOTING: CLIENT BASICS

There are several components to the complete HylaFAX software package:

- client applications (sendfax, faxstat, faxrm, etc.),
- the job submission and server status process that runs on the server machine and communicates with client applictions (faxd.recv),
- the facsimile server processes that service each modem (faxgetty), and
- the central scheduler for outbound jobs (faxq).

If you are having trouble first try to identify which component is failing. All client applications support a `-v` option to enable various levels of debugging. It is possible with one or more `-v` options to trace the protocol between the application and the *faxd.recv* process on the server machine. *faxd.recv* has a `-d` option that enables tracing of the protocol it receives and its general operation. If you are in doubt about where a problem lies, try the following:

Run *faxstat* to request server status. You should see something like:

```
Server on localhost:FIFO for all modems: Running.
Server on localhost:ttym2 for "+1 510 526-8781": Initializing server.
```

or possibly,

```
Server on localhost:FIFO for all modems: Not running.
```

If you do not see something like this, then you are having problems communicating with the *faxd.recv* program on the server machine or there is a configuration problem on the server machine. If you cannot establish a connection to the *faxd.recv* process on the server machine, then verify that you have your `FAXSERVER` environment variable setup correctly (if the server is not on the machine where the *faxstat* program is run) and/or that both client and server programs are communicating on the same TCP port. On the server machine make sure that the *faxd.recv* program is properly configured to be invoked by the inetd program; check the contents of /etc/inetd.conf, or similar, for a line of the form:

```
fax     stream  tcp     nowait  fax     /usr/etc/faxd.recv      faxd.recv
```

Be certain that inetd has ‘‘reread’’ its configuration file; either send it a SIGHUP or restart it. (This work should automatically be done when the faxaddmodem program is run on the server machine.)

Note also that the ***fax*** *service must be defined on the server machine in order for inetd to startup the faxd.recv program--check for this entry in the /etc/services file and/or the YP/NIS database.*

As a last resort, you can always use an existing network service such as telnet to communicate with the *faxd.recv* process;

```
flake% telnet localhost fax
Trying 127.0.0.1...
Connected to localhost.asd.sgi.com.
Escape character is '^]'.
help
error:Protocol botch, malformed line "help".
Connection closed by foreign host.
```

[*Ignore the protocol botch, this is just to show you that you can reach faxd.recv using telnet.*]

If the network-related configuration is setup properly but *faxstat* still does not return one of the above lines; use the `-v` option to trace the client-server protocol:

```
hyla% faxstat -v
connect to oxford.chez (155.11.194.2) at port 4557
-> version:2
-> userID:sam
-> serverStatus:
```

```
-> userStatus:Sam Leffler
-> .
<- server:all modems:FIFO:Running
<- server:"+1 510 526-8781":ttym2:Running and idle
<- EOF
Server on oxford.chez:FIFO for all modems: Running.
Server on oxford.chez:ttym2 for "+1 510 526-8781": Running and idle.
```

A protocol and/or configuration problem should be evident from the trace information. You can also configure *faxd.recv* to log its operation through syslog by supply a `-d` option in the **inetd.conf** file:

```
fax     stream  tcp     nowait  fax     /usr/etc/faxd.recv      faxd.recv -d
```

Once again, remember that *inetd* will not see a change to the **inetd.conf** file until it is restarted or sent a SIGHUP; and that *faxd.recv* logs its debugging information through syslog (using the *syslog* facility setup when the *configure* script was run to configure the HylaFAX software).

TROUBLESHOOTING: CLIENT ACCESS CONTROL PROBLEMS

Given a working client-server setup, problems that you encounter are either server-related or related to misconfigured permissions on the server machine. The *faxd.recv* program will process any client requests that do not create a new outbound job, alter an existing job's parameters, or remove a job from the queue. To do one of these *privileged* operations, a client must have access according to the **etc/hosts** file located in the spooling are on the server machine. Consult the manual page hosts(4F) for details on this file.

TROUBLESHOOTING: SERVER BASICS

One faxq process manages the scheduling and initiation of all outbound traffic; it carries out many tasks:

- manage the queue of outbound jobs
- prepare PostScript documents for transmission
- start the processes that do the low-level work to transmit facsimile

One faxgetty server process is usually run for each fax modem on a machine. These processes are responsible for handling inbound traffic; they also carry out many tasks:

- answer incoming calls and decide whether the call is fax or data
- do the low-level work to receive facsimile
- invoke the system *getty* program to handle data calls

Proper setup of HylaFAX involves setting up the modem configuration files and the ancillary programs and command scripts that are invoked by faxq and by faxgetty. The setup of the ancillary programs should be done when the software is configured and installed. The modem-related setup and much of the system-related configuration work should be done when the *faxaddmodem* command is used to create a modem configuration file. Problems that arise in the normal operation of a server typically fall into two categories:

- incorrect configuration and/or setup of the server software,
- incorrect configuration of the modem.

In either case HylaFAX provides extensive tracing facilities that should supply the information needed to locate and correct a problem.

The tracing controls are completely described in the config(4F) manual page. faxq tracing information is controlled by two configuration parameters: **ServerTracing** and **LogFacility**. The **ServerTracing** parameter controls which parts of the work done by faxq that it should generates messages for. Server tracing information is captured via syslog(3) using the syslog facility specified by the **LogFacility** configuration parameter (by default ‘‘`daemon`’’). To capture server tracing you must enable the appropriate bits in the **ServerTracing** parameter and setup the syslogd process to capture `daemon.info`, `daemon.debug`, and `daemon.err`; or substitute for `daemon` to reflect the value of the **LogFacility** parameter. Note that setting **LogFacility** to something like `local5` it is easy to capture HylaFAX syslog messages in a separate file; just setup the **syslog.conf** file appropriately, e.g.

```
local5.*            /var/spool/fax/etc/syslog
```

The `SessionTracing` parameter controls tracing information *during the time HylaFAX is engaged in conversation with another fax machine*. Protocol-related problems will be found in the information logged under this control. Session tracing information is stored in files in the **log** subdirectory in the spooling tree and is returned to users via electronic mail when notification is requested, or when an unrecoverable error is encountered. Consult log(4F) for more information.

TROUBLESHOOTING: QUEUE MANAGEMENT

There is very little that can go wrong with the server with respect to managing the queue of outbound jobs. Setting the system time backwards on the server machine can cause problems as time-to-send stamps for jobs are recorded as absolute time values. Jobs may be rejected without a phone call if a **rejectNotice** entry is present for a destination phone number in the destination controls file destctrls(4F). Beware that multiple jobs to the same destination are usually serialized to reduce phone calls. Jobs that are blocked in this way have a "*Blocked by concurrent...*" status. You can change the maximum number of concurrent jobs that will be scheduled to a destination with the **MaxConcurrentJobs** configuration parameter.

Otherwise if a problem exists enable the queue management bit in the **ServerTracing** parameter. There is also a separate bit for tracing low-level job queue operations; this should not be needed.

TROUBLESHOOTING: POSTSCRIPT DOCUMENT PREPARATION

When debugging problems related to the preparation of PostScript documents, collect the appropriate command line arguments to the ps2fax(1M) program and invoke it directly on an offending document. Beware, however, that if the fax server is started up by the init(1M) program that it may inherit a different shell environment. In particular, beware of problems with search paths when the PostScript interpreter program is linked with Dynamic Shared Objects (DSO's); e.g. when Ghostscript is linked with the the X11 driver and a DSO version of the X library.

*On machines with dynamic shared libraries (e.g. SunOS), if you link Ghostscript with the X11 device driver and use shared X11 libraries that are not in a standard location, then you may need to augment the HylaFAX **util/ps2fax.gs.sh** script with something of the form:*

```
LD_LIBRARY_PATH=/usr/local/R5/lib:/usr/openwin/lib
export LD_LIBRARY_PATH
```

Troubleshooting: Communication problems

If a problem occurs during server initialization, set **ServerTracing** to 11, or similar, in the modem configuration file and check the session trace logs. Problems at this point are typically caused by:

- misconfiguration of the flow control scheme,
- not disabling the *getty* program on the modem port (so that multiple processes are simultaneously trying to read/write to the modem), and
- invalid initialization sequences.

Normal installation of HylaFAX will enable sufficient session tracing to debug session communication problems. Unexpected problems may require you to alter the value of the **SessionTracing** parameter.

When debugging modem-related problems, only enable the tracing that you really need. Enabling all tracing can affect the operation of the server by altering the timing of operations.

The default configuration files come with **SessionTracing** set to 11 which is a good setting for Class 2 modems (i.e. a lot of information is provided, but the load on the server should not keep it from operating properly). For Class 1 modems a setting of 0x4f will also cause HDLC frames to be collected. Beware of tracing timer operations and modem I/O; these trace flags are only useful if you are trying to debug a problem specifically related to a timer not going off, or a problem where data appears to be corrupted.

Note that when capturing a trace for the purpose of submitting a bug report, the less extraneous information that you include, the easier it is for people to help understand the problem. A good way to capture a transcript of a problem is to use the transcript shell script in the bin directory of the spooling area. For example,

```
% cd /var/spool/fax
% bin/transcript 2241 14159657824
```

where "2241" here is the process id of the *faxd* process and "14159657824" is the canonical phone number associated with the problem (i.e. the destination if an outgoing call or the local number if an incoming call). Note that transcript will only present the trace associated with the last session, so you need to run it immediately after the problem occurs (or else extract the session by hand).

Most of the time HylaFAX will return the relevant session log for a communication failure in the notification message sent to a user when an outbound job fails. Note however that the contents of this log is defined by the value of the **SessionTracing** parameter specified in the per-modem configuration files. If this parameter is set too low then session logs may be returned that do not show sufficient information to diagnose a problem.

TROUBLESHOOTING: GETTY PROBLEMS

HylaFAX will invoke the *getty* program when a data connection is established and the `GettyArgs` parameter is set to a non-null string. When HylaFAX starts up a *getty* program it sets the `stdin`, `stdout`, and `stderr` descriptors to the modem device (closing all other descriptors), creates a new process group, and turns off the `CLOCAL` bit on the device file so that if carrier is dropped the process group should receive a SIGHUP.

- Be sure that the system is not also configured to start *getty* on the modem's tty port.
- Beware of misconfigured *getty* arguments that cause *getty* to reopen and/or initialize the modem device.
- Also be careful that on some systems the **/etc/ttytab** entry for the modem port must be setup as a *dialup* line as opposed to a *local* line.

The HylaFAQ has information about other system-specific problems that you may run encounter.

TROUBLESHOOTING: PROBLEMS WITH UUCP, CU, TIP, ETC.

If you have a problem running the fax software together with other communication programs such as *uucp*, *cu*, *tipslip*, *ppp*, etc. first verify that your data communication software is configured to use the correct modem initialization strings. In particular, beware that most of the prototype modem configuration files for the fax server will leave the modem ‘‘*idling*’’ in Class 1, Class 2, or Class 2.0. This means that in order to place a data call the modem must first be reset to Class 0. Other common problems involve the ownership and protection of the modem devices. When the fax server is running it forces the tty device to be owned by the ‘‘**fax**’’ user (typically the same UID as the ‘‘**uucp**’’ user) and to have the mode specified by the **DeviceMode** configuration parameter. Finally, beware that there are several different styles of UUCP lock files; verify that your UUCP and related programs use the same style that HylaFAX is configured to use. The UUCP lock file scheme used by HylaFAX may be specified with the **UUCPLockType** configuration parameter. If you specify this parameter be certain to put it in both the faxq configuration file and each modem configuration file; otherwise one HylaFAX server process may do the right thing while another may not.

HylaFAX table of contents.

Mail to Fax Gateway

It is easy to setup a simple mail to fax gateway facility with the tools included in this distribution and some simple additions to your mail delivery agent configuration.

1. Setup the HylaFAX software as usual.
2. If your system uses sendmail to deliver mail, then follow the instructions in **faxmail/mailfax.sh-sendmail**. (Thanks to Eric Allman for the sendmail configuration hack.)
3. If your system uses smail (e.g. Linux users), then follow the instructions in **faxmail/mailfax.sh-smail**.
4. Restart your mail software, refreeze your configuration or whatever is necessary to cause the configuration changes to be seen by the system.

Voila! Now mail to *user@dest.fax* will get formatted and submitted as a facsimile job to *user* at the specified *dest*. By writing a more involved mailfax script you can add things like different resolutions by parsing the *user* string. For example, an X.400-style scheme might use:

```
user/R=low/N=done@host.fax
```

where ‘‘R=low’’ specifies a low res facsimile and ‘‘N=done’’ requests notification when the job is done. If you want symbolic destinations, then you can provide a host to phone number mapping facility.

The normal access control mechansims on submitting a facsimile for transmission are enforced; you may need to use them if you setup a fax gateway on your system!

Dirk Husemann has contributed a more elaborate scheme that supports the transmission of MIME-encoded documents; check the FAQ.

There is also the "Remote Printing Experiment" being conducted by folks on the Internet. This provides a distributed fax delivery service that uses email to relay facsimile to a local site for delivery. For info about that stuff send mail to tpc-faq@town.hall.org or visit their WWW site.

HylaFAX table of contents.

Contributed Software

You should be able to find contributions from HylaFAX users in the contrib subdirectory of the public ftp area where you found this software (or look in sgi.com:~ftp/sgi/fax/contribNew contributions are welcome, so long as the author’s identity is clearly marked so that questions go directly to the author and not to me. The HylaFAQ also has information about specific contributed software programs.

HylaFAX table of contents.

ACKNOWLEDGEMENTS

This software is more than 5 years old and is the product of many folks' work. Robin Schaufler did the original scheme for delayed submission and worked on the client-server protocol. More recently, the following people have helped either by testing or by contributing fixes and/or improvements:

Matthias Apitz	Chris Beekhuis	Peter Bentley
Marc Boucher	Bill Campbell	Brent Chapman
Tom Corson	Alan Crosswell	Mark Diekhans
Greg Ferguson	Steve Fine	Andrew Ford
Nico Garcia	Wolfgang Henke	Viet Hoang
Bert Hooyman	Ken Hornstein	Dirk Husemann
Brian Katzung	Masao Kitano	Carsten Koch
John T Kohl	Rickard Linck	Rick Lyons
Tom Lislegaard	Rob MacKinnon	Kevin McManamon
Les Mikesell	Bill Morrow	Andy Moskoff
Chris Munonye	Rob Newberry	Jonas Olsson
Dave Packer	Damon Permezel	David Pike
Amir Plivatsky	Andy Rabagliati	Eric Rescorla
Marshall Rose	Daniel Rosenblatt	Joel Rosi-Schwartz
Tim Rylance	Joseph E. Sacco	Brent Townshend
Peter White	David Vrona	Paul Vixie
Christian Zahl		

(and surely others). Also, a special thanks to Ed McCreight for helping me understand the stuff *between the lines* that's necessary to make a working Class 1 driver.

The regular expression support is based on Henry Spencer's POSIX 1003.2 compliant regex package that has

```
Copyright 1992, 1993, 1994 Henry Spencer.
All rights reserved.
```

Consult **util/regex/COPYRIGHT** for the full copyright notice associated with this software. The code to read PCF fonts is distantly related to the X11R5 code that is

```
Copyright 1990 Massachusetts Institute of Technology
```

Consult **faxd/PCFFont.c++** for the full copyright notice. The `textfmt` program is distantly related to the `lptops` program written by Nelson Beebe; there was no copyright notice on the version of the code that textfmt grew out of.

The **config.guess** and **config.sub** scripts are part of the GNU autoconf package and covered by the GNU Public License (GPL). Several ideas in the configure script are directly "borrowed" from autoconf (and I have tried to maintain as much compatibility as possible).

HylaFAX table of contents.

HylaFAX Documentation

There is documentation! There is **gobs** of documentation. When in doubt **read the manual pages**. There are manual pages for all the programs and manual pages for all the files and directories that you may be curious about. Of course there is also source code for everything, but this should (hopefully) not be needed. A useful introduction to the client applications is given in hylafax(1). If you want to learn how the server and spooling system work, look first at hylafax(4F).

There are several mailing lists for users of the HylaFAX software, consult the next section for an easy way to join the appropriate one(s).

Finally, the following ancillary materials are part of this site:

- FlexFAX Modem Bakeoff report (old)
- FlexFAX/HylaFAX modem information
- ZyXEL modem voice support information
- Supra/Rockwell Class 1 documentation
- Supra/Rockwell Class 2 documentation
- Telebit T3000/WorldBlazer FAX commands
- Hayes technical reference for modem users
- Peter Chen's "What you need to know about modems"
- FAX modem command scorecard (*old*)

The following materials are available elsewhere:

- comp.dcom.fax FAQ
- CCITT Standard Fax Test Images - for testing fax quality
- CCITT Group 3 Fax Standard Online - fax standards documentation
- Facsimile Standards Publications Ordering - Info on ordering fax standards publications
- ITU Facsimile Ref Index - Auto-search of ITU for their fax related publications
- ITU Documents - General ITU dociment index, with search
- Rockwell Fax info - Rockwell 144ACx AT Command Set (883R2) & other fax related materials
- Fax Protocol & Standards Reference - by Gray Associates

HylaFAX mailing lists and how to submit bug reports.

HylaFAX table of contents.

HylaFAX Mailing List and Other Final Words

There are two mailing lists for users of the HylaFAX software:

- flexfax@sgi.com, a public list for all users of the HylaFAX software;
- flexfax-announce@sgi.com, a low volume list where announcements of new versions of HylaFAX

are posted.

Note that `flexfax@sgi.com` tends to have a fair amount of traffic about the server-side software; if you are not running a fax server it may not be worth joining. Beware also that this mailing list has many people on it. Please take this into consideration when posting notes to it; i.e. *avoid posting large trace logs and the such.* Also, when corresponding about this software please always specify:

- what version you have (check here if you are unsure of your software version),
- what system you're running on, and,
- if the problem is modem-related, identify the modem and the firmware revision.

For example: "HylaFAX v2.3.0alpha99 under Solaris 2.3 with gcc 2.5.7; ZyXEL 1496E with 6.11a firmware."

Note that there is no need to subscribe to both the `flexfax` and `flexfax-announce` mailing lists; postings to the announcement list are automatically fed to the normal `flexfax` list. All lists are maintained with the Majordomo mailing list software. If you want to subscribe to any of these lists, you can use the form below, or send electronic mail to Majordomo@engr.sgi.com.

Subscribe `flexfax@sgi.com`
Subscribe `flexfax-announce@sgi.com`
EMail Address:

If you have not previously done so, please fill out and submit the survey form. The information provided in this form is purely for use in assisting users of HylaFAX.

HylaFAX table of contents.

HylaFAX Use and Copyright

Silicon Graphics has seen fit to allow me to give this work away. It is free. There is no support or guarantee of any sort as to its operations, correctness, or whatever. If you do anything useful with all or parts of it you need to honor the copyright notice enclosed below which basically says ''*you can't use my name or Silicon Graphics' name without our written permission*''. I would however be interested in knowing about any interesting work with HylaFAX and, hopefully, be acknowledged. Finally, note that HylaFAX is a trademark of Silicon Graphics, Inc.

HylaFAX table of contents.