Archie 3.5

System Manual

Bunyip Information Systems, Inc.

Bunyip Information Systems Inc.
310 Ste. Catherine St. W
Suite 300
Montreal, Quebec
Canada H2X 2A1

Tel: (514) 875-8611
Fax: (514) 875-8134

email: info@bunyip.com
archie-group@bunyip.com
archie-admin@bunyip.com

Contents

Chapter 1

Introduction

It was inevitable that the technology under which Archie was built changed. With the introduction of faster applications and newer media on the Internet, Archie had to reshape its structure to accomodate these new factors and stay in the race.

Speed in this day and age is a big gain and Archie is thought to have achieved it with the utilization of a string indexing tree for faster string searches.

As for the World Wide Web and its many new faces, Archie also introduces its latest web crawling and indexing engine. In addition, a new web interface has been added to accomodate the web users. And with a few more newly added features and face-lifts, Archie v3.5 is ready to undergo deployment.

This manual is an attempt to clarify the many components of Archie v3.5 and its functionality. Please keep in mind that this manual is also undergoing changes, so we appreciate any comments sent to archie-group@bunyip.com.

What follows is the list of added features in Archie v3.5. You will find in the table of contents, symbols such as the one in the margin, to help you find the relevant documentation concerning the new features. 3.5

- Support for a new database module (webindex):
  - Retrieval of HTML pages through HTTP protocol.
  - Keyword extraction.
  - Controlled crawling of the WWW.
  - Site by site basis.
  - Content extraction.
  - Configurable stoplist (keyword exclusion).
- New database structure:
  - A more reliable and independent database structure.
- New strings search engine:
  - Based on a paged, tree structured strings index.
  - Faster searches.
  - Less memory required.
  - More disk space needed for construction of the index.
- New search interface:
  - New WWW (CGI compliant) front-end to the Archie database.
- A better domain filter for any database:
  - Results can be pre-configured to return in a certain order (e.g. FTP sites closest to the server first).

  - Configurable on a server basis.
- General new features:
  - Support for ls-lR.gz files.
  - ISO-Latin-1 support for searches.

1.1 Future

We hope in the very near future to be able to produce the following list of features. If you have any suggestions to add to the list please feel free to contact us at archie-group@bunyip.com:

- Search tools:
  - Add boolean searches.
  - Allow other types of searches. (For example, return a list of hosts in any database and when they were last updated.)
  - Multi-lingual Web front-end.
- Database:
  - An automated check-up program that runs periodically on the databases. It will check the consistency of the database and fix any errors it runs into. This will reduce the amount of time an Archie administrator needs to spend maintaining an Archie server.
  - Improve on filters and parsers to handle strange and unusual site listings.
  - Parallelize the parsing and updating of different databases.
  - Build multiple databses with the same access method.
- Management tools:
  - CGI web interface to allow web and FTP administrators to register their sites automatically.
  - Replace "host_manage" with a new method for managing the host_db database.
  - Statistical tools.

Chapter 2

This Manual

2.1 Who should use this manual

Most of the information in this manual is prepared at a technical level. In order to carry out the necessary tasks for maintaining an Archie installation, you should have a reasonably strong technical background in UNIX system administration. Thus, it is assumed you feel comfortable with the level of material presented in the books listed at the end of this section.

2.2 How to use this manual

This manual has been divided into sections to deal with the major steps in getting Archie out of the (virtual) box, and into the working world of your system. The first section present an introduction to the Archie system; subsequent sections deal with Archie installation and configuration.

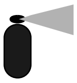
2.3 Font conventions

In this document pathnames are represented in this `font`. System manual page titles and program names are in **bold typeface**. Command line entries are prefaced with a '%' sign like this:

```
% cat /etc/passwd
```

New features in the version 3.5are indicated by the symbol presented here in the margin.

Special notes are enclosed in boxes



Sections that start with the symbol displayed here in the margin indicate troubleshooting information. These sections provide a checklist of items to be resolved if trying to locate a problem within the system. Common problems are listed, as are ways to fix them.

2.4 Suggested readings

The following books describe UNIX systems/networks at the level of expertise expected from an Archie system administrator. They provide excellent reference material if you feel there is some area of expertise in which you require more information.

- The Whole Internet User's Guide and Catalog, Ed Krol, O'Reilly and Associates, Inc., California, USA, 1992. (ISBN 1-56592-025-2)

- UNIX System Administration Handbook, Evi Nemeth, Garth Snyder, and Scott Seebass, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1989. (ISBN 0-13-933441-6)
- The UNIX Programming Environment, B. W. Kernighan and R. Pike, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1984.
- UNIX Network Programming, W. Richard Stevens, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1990. (ISBN 0-13-949876-1)
- Internetworking with TCP/IP Volume 1, Douglas E. Comer, Prentice Hall, Englewood Cliffs, New Jersey, USA,1991. (ISBN 0-13-468505-9)
- Internetworking with TCP/IP Volume 3, Douglas E. Comer, Prentice Hall, Englewood Cliffs, New Jersey, USA,1993. (ISBN 0-13-474222-2)

2.5 Where to go from here

The rest of this manual has been divided into sections to deal with the major steps in getting Archie out of the box and into the working world of your system. It is organized as follows:

- "Installing the basic Archie system" on page 15 describes how to get the new Archie software and initially installed on your system.
- To properly configure the Archie system for your site, you must have an understanding of the architecture of the Archie system; a discussion of this is presented in "System Overview" on page 5.
- "Configuring the Basic System" on page 20 deals with configuring your Archie software to carry out the tasks you require at your site.

The concluding section, contains a list of Frequently Asked Questions about the Archie system which you may like to use as your first reference when something seems to go wrong.

Chapter 3

System Overview

In order to be able to configure the Archie system properly for your site, you will need a thorough understanding of the Archie system architecture. This section introduces more Archie system concepts, to prepare you for the configuration instructions (in "Configuring the Basic System" on page 20). First, a whirlwind tour of Archie.

3.1 An Archie-eye view

On first inspection, the Archie system can appear overwhelmingly complex. In fact, Archie is a straightforward tool for resource discovery in the Internet environment. Its apparent complexity stems from the fact that it both seeks out and provides information to Internet users, and it maintains information on several levels. The following scenario is designed to help de-mystify some of the operation of the Archie system. Consider the sample Internet subset in the diagram below:

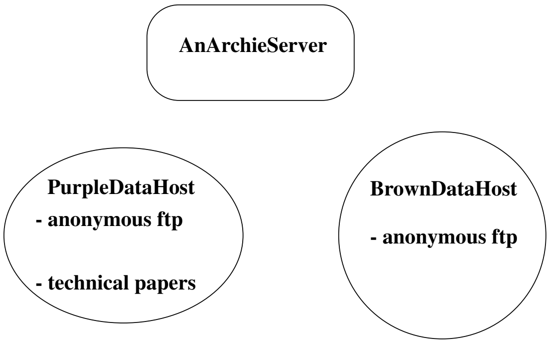


Figure 3.1: Basic Archie Relationships

3.1.1 Archie as an information accessor

Suppose AnArchieServer is configured so that it maintains information about the contents of the anonymous FTP archives at PurpleDataHost and at BrownDataHost. This means that AnArchieServer has an anonftp catalog. Furthermore, suppose AnArchieServer keeps a catalog to track the tech papers available on the Internet (only available at PurpleDataHost, in this scenario).

To support its operation, AnArchieServer maintains Data Host Information Files to store information about the particularities of PurpleDataHost and BrownDataHost. The catalog contains information about each data host's

Internet address, operating system type, available information (technical papers and/or anonymous ftp archives), etc. This is described more fully below.

Periodically, AnArchieServer will access the data hosts it knows about (i.e., PurpleDataHost and BrownDataHost, here) and determine if the available information (anonymous FTP files, tech papers) have changed. If so, the anonftp and tech papers catalogs at AnArchieServer are updated to reflect the changes.

3.1.2 Archie as an information provider

Once AnArchieServer has been set up as described above, people looking for information (either technical papers or anonymous FTP files) can query various programs running on AnArchieServer directly. For instance, an individual user can formulate a query of AnArchieServer's tech papers catalog to determine which technical papers are of interest. The AnArchieServer tech papers catalog would be able to supply the Internet address of PurpleDataHost, as well as the location of the particular technical papers in PurpleDataHost's repository. Similarly, if a user was looking for information available by Anonymous FTP, consultation of AnArchieServer's anonftp catalog would allow the user to select which files to get, from which anonymous ftp archive, before actually accessing either of PurpleDataHost or BrownDataHost. In fact, the user need not know, in advance, what Internet sites are data hosts, or which data hosts provide information: AnArchieServer provides this information as it is needed.

3.2 The two faces of the Archie system

The Archie system is an information mediator in the sense that it both seeks out information from other sites on the Internet and provides users access to the resulting catalogs.

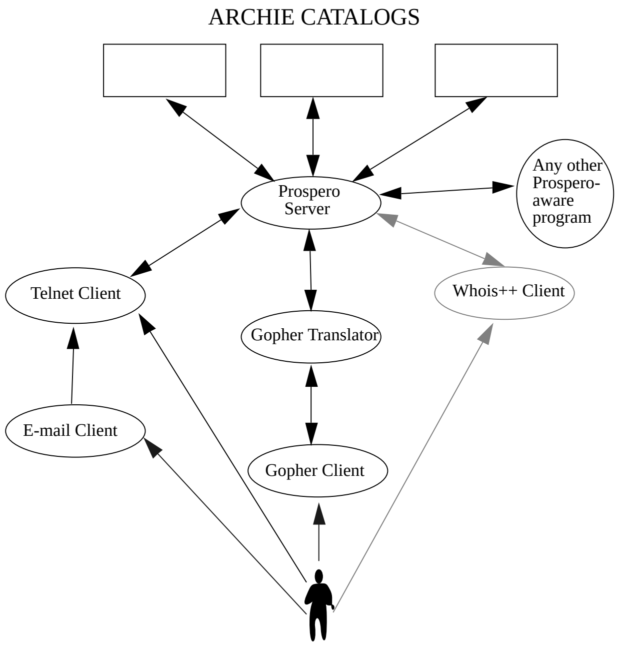


Figure 3.2: User access to the Archie catalogs

The information provider aspect of the Archie system is shown in Figure 3.2. Originally, the Archie system used a modified Prospero File System server (**dirsrv**) to provide user access to its catalogs. Another module has been added to the system to access the catalogs via WWW browsers.

The Archie system catalogs are maintained by the Update Cycle processes shown and described in the following pages. This is illustrated in Figure 3.3. 3.5

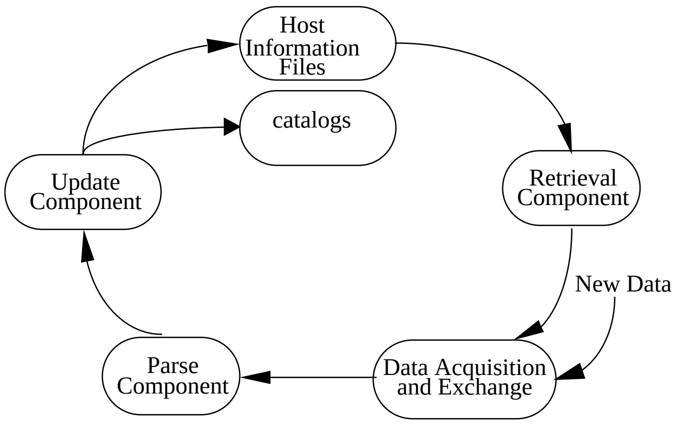


Figure 3.3: The Archie update cyle

Of more immediate importance to Archie system administrators is the information-seeker aspect of the Archie system. This is composed of several components which work together to retrieve, parse and update data in the Archie databases, and is illustrated in Figure 3.4.

3.3 Overview of the Archie Update Cycle

First, some useful definitions:

- The catalogs are the collection files which are consulted when a user carries out an Archie search. For example, the anonftp catalog contains an index of the files available from anonymous FTP archives on the Internet while the webindex catalog indexes the WWW space.
- The term Data Hosts refers to those network sites which contain the information that the Archie system will hold in its catalogs.
- The Host Information Files or Host Databases contain the latest information about each of the Data Hosts and the data stored there. The Host Information File Manager is responsible for maintaining these files.
- From the perspective of the Archie administrator, Archie servers fall into two categories. The local Archie host (site, server) is the one that the administrator is responsible for maintaining. Remote Archie hosts (sites, servers) are all those other sites on which the Archie system is being run.
- Each component is controlled by a component manager.

The major steps of the Update Cycle are:

- The Retrieval Component consults the Host Information Files to determine which Data Hosts should be scheduled for inspection. This information is used to start the Update Cycle; the rest of the Cycle is concerned with accessing these sites and retrieving any new data.
- The Data Acquisition component is concerned with obtaining any new data directly from the Data Hosts for which the Archie server is responsible.
- The Parse Component is responsible for parsing the incoming data into a form suitable for the update of the Archie server's catalogs. This often means removing extraneous information (filtering), as well as performing data conversions, collation and compression.
- The Exchange Component obtains data from remote Archie servers, if it has been configured to do so. Data so obtained is inserted into an appropriate place in the Update Cycle and is ultimately incorporated into the Archie catalogs on the local Archie host.

- Finally, the Update Component obtains the parsed data and integrates it into the appropriate Archie catalog files. This entails updating the catalogs with the information obtained from the Data Hosts (or other Archie servers), as well as making changes to the Host Information Files. Recall that the Host Information Files contain information such as the last time a Data Host was accessed for inspection; this information must be brought up to date at the end of each Update Cycle.

Before describing the Update Cycle process in more detail, some important concepts need to be described.

In order to carry out its processing, the Archie system must maintain special information about the Data Hosts from which it gets information, and the status of the information stored in the Archie catalogs that has been obtained from those sites. These particulars are stored in files collectively referred to as the Host Information Files. They are maintained by the host_manage program.

The Host Information Files fall into two categories: a primary information file and (possibly) several auxiliary information files. These reside in

    ~archie/db/host_db/host-db.dir

and

    ~archie/db/host_db/host-db.pag

The primary host information file contains particulars about the physical Data Host itself:

- the primary name of a site
- the primary IP address
- the operating system (if known)
- timezone (if known)
- access methods (catalog names)

The Archie server also maintains auxiliary Host Information Files in

    ~archie/db/host_db/hostaux-db.dir

and

    ~archie/db/host_db/hostaux-db.pag

These contain details for a particular Archie catalog. For example, if the Archie server is keeping track of the anonymous FTP listings of the site bunyip.com, there would be an entry in the auxiliary files for bunyip.com/anonftp.

These files contain, in part:

- The name of the Archie catalog (e.g., anonftp or webindex)
- The preferred name (that is Domain Name System CNAME, if set)
- Source Archie server
- Time of last data retrieval
- Time of last data parse
- Time of last update
- Number of records (catalog-specific definition)
- Number of times an update has failed
- Any comment that the Archie system has seen fit to communicate to the administrator (reasons for failure of update for example)
- The current status of the information service at the Data Host (active, inactive etc.)
- The name of the Archie host responsible for monitoring this site (for those cases in which the data is being exchanged between Archie hosts)

The Data Host Information Files are maintained by the host_manage program, which is described in "Managing the Host Information Files" on page 33.

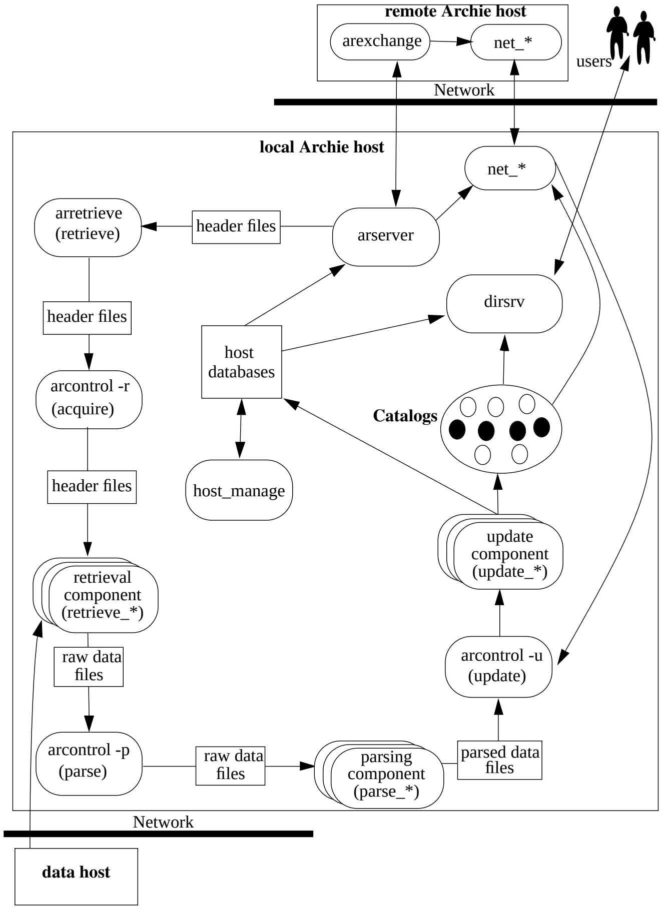


Figure 3.4: The Archie system process interactions

Every piece of New Data introduced in the Archie Update Cycle begins with an Archie header record. This contains all the information necessary for the various components to process the data obtained from the Data Host. The information in the header is modified throughout the Cycle to reflect the changing status of the data. Much of the final information in the header record is used to update the Host Information Files at the end of the Cycle. While in most cases the Archie system administrator need not be concerned with the header records of data within the Archie system, it is useful to know what the record contains, if problems should occur.

In some cases, the header itself contains all the data necessary to complete the Update Cycle. For example, for an Archie catalog describing the status (reachable or unreachable) of sites on a network, an indication of “success” or “failure” in the header would be enough to know whether to proceed with the update.

The header is in printable ASCII characters and is human-readable regardless of the format of other data which may follow it. All headers start with the string @header_begin and are terminated with @header_end. The data itself starts immediately after the final NEWLINE of the termination string.

It is not advisable to edit Archie data files, especially if the data is in binary form. Most text editors do not correctly handle binary files.

A description of the header can be found in Appendix A. Currently, all header fields are optional, with the exception of the primary hostname entry.

3.4 Update Cycle file naming conventions

All data files created and maintained during the Update Cycle have the following naming convention:

`<primary site name>-<dbname>[_<id number>].<phase suffix>[<temp>]`

where

`<primary site name>:` is the name of the Data Host.

`<dbname>:` is the name of the Archie catalog to be updated with this data.

`<id number>:` is an arbitrary number determined by components of the Update cycle to prevent name clashes (i.e. when there is more that one set of data associated with the `<primary site name>/<dbname>` pair). Note that this is OPTIONAL and may not occur in certain file names.

`<phase suffix>:` is one of retr, parse or update specifying which part of the update cycle the data is destined for, not from which the data came.

`<temp>:` is a string appended to the name if the current file has a temporary status, i.e. is undergoing a data transformation (such as being uncompressed). Currently this is "_t".

An example using this naming scheme is:

```
bunyip.com-anonftp_54.parse_t
```

All files undergoing processing are stored in the ~archie/db/tmp ("holding") directory, unless otherwise specified (using the "-t" switch).

The manual pages describe how to override the default holding directory for the Archie programs. However, the directory you choose instead of `~archie/db/tmp` must not be on a different filesystem (partition) from ~archie, as the programs make use of "hard links" which cannot cross partition boundaries. See ln(1).

3.5 Process naming conventions

By reading the header record the controlling process of Archie's Update Cycle automatically generates the name of the appropriate program to invoke for the successive phases in data processing. The syntax for the process name is

```
<component name>_<dbname>[_<os type>]
```

where

`<component name>:` is one of retrieve, parse, update or net

`<dbame>:` is the name of the Archie catalog with which this data is associated

`<os type>:` is optional and is used when different operating systems require different processing functions (e.g., when parsing a UNIX, Novell or VMS anonymous FTP listing).

For example, if the control program reads a header which specifies that this data is destined for an anonymous FTP catalog, and that it requires parsing, the control program will invoke the parse_anonftp program. This program may, in turn, look at the header and determine that it is a UNIX BSD listing and will invoke parse_anonftp_unix_bsd on the data file for processing.

There will be cases in which the operation to be performed is similar enough that the same program may be used for more than one purpose. In this case soft links, which follow the naming convention, are made to a single program.

The update process is complex and requires consistent and complete information in order to complete successfully. This means it is possible to have a failure at any step in the Update Cycle. For example, the data host may not be reachable, or the parsing of the obtained data may fail. In this case, the program will produce a file containing a header reflecting the nature of the failure. This usually contains an explanation of how or why the failure occurred (this is signified in the header record by the comment field). The "failed" data continues to be passed through the Update Cycle and at each phase the header is updated to reflect what has happened to it. Thus, even though the data acquisition has failed, the last_parsed field of the header will be updated.

3.6 The Update Cycle in detail

Now, each component of the Update Cycle will be discussed in some detail.

The Update Cycle is performed, step by step, under the guidance of the arcontrol program. It is invoked in different modes depending on the phase of the Cycle desired. It scans the holding directory for the appropriate data files and may perform some basic data transformations on those files (such as uncompressing the data) before passing them on to the appropriate programs for processing.

The steps followed by the system are as follows:

3.6.1 The Retrieval Component (arretrieve)

This component is responsible for querying the local Archie system for the list of sites scheduled for inspection. It does so by contacting the local Archie arserver process (see "The arserver program" on page 34) and downloading a set of headers corresponding to those Data Host/catalog combinations. Each header is placed in a separate file, ready for the Data Acquisition phase of the cycle.

3.6.2 The Data Acquisition Component (arcontrol -r)

arcontrol examines the header files (created by the Retrieval Component). For each Data Host/catalog pair, it determines the appropriate process for retrieving new data for the catalog from the Data Host. arcontrol invokes the selected process, passing the name of the header file as a parameter. Once the process has finished acquiring information from the Data Host, a file containing the results will be created (according to the scheme outlined in "Update Cycle file naming conventions" on page 23). The file will contain the header record and either retrieved data or an indication of failure.

3.6.3 The Parse Component (arcontrol -p)

Managed by arcontrol, the processes responsible will parse the files generated by the Data Acquisition Component into the appropriate form for the target catalog. For example, in the case of anonymous FTP listings, it will take the raw data and create an output file of the correct form for insertion into the anonftp catalog.

3.6.4 The Update Component (arcontrol -u)

Invoked by arcontrol, the Update Component process performs two functions: it will update the Host Information Files, and it will update the Archie catalog itself. For example, when a correctly parsed anonftp (anonymous FTP) listing from bunyip.com is ready, the data itself is inserted into the anonftp catalog and the record for "bunyip.com.anonftp" is modified in the auxiliary Host Information Files.

3.7 Data Exchange

This release of Archie allows the exchange of arbitrary catalog data between cooperating Archie systems. A basic flooding i-have/send-me algorithm, similar to that used in the Internet USENET news system, is used to maintain weak consistency between an arbitrary number of Archie servers. “Virtual” networks of Archie systems can be created and any individual server may belong to zero or more of these data-exchange networks.

This allows a multi-tiered system (such as the example on the map) whereby certain key sites may be responsible for inter-continental data exchanges and others would feed from them, instead of every Archie site having to contact all others worldwide.

3.8 Conclusion

This concludes the description of the Archie Update Cycle processes. With a better understanding of how Archie works, you should be now be ready to configure your Archie server to fulfill your needs. The discussion of Archie configuration begins in section 6, “Configuring the Basic System” .

Chapter 4

The database

3.5

4.1 Overview

The database has been completely revised. The goals of this reconstruction were the building of a faster system and more reliable database. The back-end of the database is composed of

- Index
- Site files
- Start database

The index is a tree structure, allowing fast searches of the strings file. Rebuilding the index is a relatively expensive process, so it is not updated after every new key is inserted into the database. Thus, in order to handle a search request, a linear search is made of the (usually) small unindexed portion of the strings file, in addition to checking the fast index. To ensure quick response times, the index should, periodically, be brought up to date with the strings file.

Within a catalog, files having the prefix ``Stridx.'', are part of the index. For example, `Stridx.Strings` is the strings file, and `Stridx.Index` is the fast index.

The site files are now independent of one another. In case of corruption, only the site that is corrupted need be considered. In certain cases, such as webindex, extra files may be present with each site file. The `.excerpt` files contain excerpts of the different URLs. The `.idx` files, present in both anonftp and webindex, hold extra information for large site files, in order to speed up searches.

The directory `start_db`, present in each catalog, holds information that ties the index with the site files.

4.2 Building the index

In order to rebuild the index, the program `db_build` must be run.

The build may take some time to run, depending on the size of `Stridx.Strings` and the amount of memory allocated to the build. One would call `db_build` as follows:

```
% db_build -d anonftp -v -k 50000 -t ~archie/db/tmp
```

The above example builds the anonftp index using 50 megabytes of memory. Temporary storage is used in the `~archie/db/tmp` directory. The man page lists all the other switches that can be used. One option of particular importance is "`% -f`", which forces the index to be rebuilt, even if the unindexed portion of the strings file is very small. By default, the program will not rebuild the index if the unindexed text is less than 1 megabyte.

You may want to rebuild the index once a week initially and then only once a month. These estimates are speculative — it depends on the rate at which new strings are added to the database.

4.3 Ordering the results

With the volume of information in the database and the amount of replication on the network, it is often desirable to return the results in some order of closeness. It is now possible to configure the order in which results are returned, according to the name of the domain to which the result site belongs. To avoid slowing down searches, this ordering is done at the time at which the data site is inserted into the database, rather than when the results are being returned.

The order is defined in the file `~archie/etc/domain.order`. The following is an example of such a file.

```
# Comment line
# Domain order for an Archie server in North America
ca:edu
com:gov
*
au:nz
```

Domains on the same line have the same precedence. Hence, `.edu` sites and `.ca` sites will be returned first, in no particular relative order. The sites in the domains `.com` and `.gov` will be listed later. To discourage users from making unnecessary, long distance file transfers, the sites from New Zealand and Australia are returned last. (We assume that users accessing a North American archie server are usually in North America.) The "*" represents all other sites.

One can also specify the domains using pseudo-domains defined in `<archie home>/etc/ardomains.cf` as well as sub-domains such as `bunyip.com`.

In order for changes, in the order of domains, to take effect, you must rebuild the precomputed orders for the existing sites. This is done with the program `% db_reorder`, listed in section 4.4.

Note that there are no leading dots "." in the listed domains

4.4 The other programs

All programs prefixed by `db_` in `~archie/bin` aid in managing the database. They are:

Program	Description
db_build	Build the database index
db_check	Verify the consistency of the database
db_dump	Dump the list of sites that are in `start_db`
db_reorder	Reorder the sites according to `domain.order`
db_siteidx	Build the `.idx` files for specific sites
db_stats	Compute statistics
fix_start_db	Fix problems in the start_db database.

Information about the command line options of these programs may be found in their respective man pages.

Chapter 5

Installing the basic Archie system

This section will guide you through the process of installing the Archie software on your system. With an understanding of the overall Archie system architecture, as presented in chapter 3, we will then proceed to the configuration of the system.

5.1 Archie system space requirements

Choosing a location for the Archie system on your file system depends on the databases that you intend to maintain. We estimate that a complete **anonftp** catalog (covering about 1200 archive sites) requires between 600 and 1000 megabytes. In the case of a **webindex** catalog, the space requirements are as much as possible!

To help decide on the location of the Archie system, let's take a look at the directory structure that will be created once the distribution is unpacked.

Directory/file pathname	Purpose
archie/	Archie system home directory
archie/cgi	CGI related directories
archie/cgi/bin	CGI binaries
archie/cgi/html	HTML files
archie/contrib/	public domain and third-party software
archie/config/	installation configuration
archie/bin/	system binaries and shell scripts
archie/db/	Archie database home directory
archie/db/anonftp_db/	anonymous FTP catalog files
archie/db/webindex_db/	WWW catalog files
archie/db/host_db/	host database files
archie/db/locks/	directory for lock files
archie/db/tmp/	temporary or "holding" directory for data files
archie/doc/	system documentation
archie/etc/	system configuration files
archie/help/	Archie client help directories and files
archie/logs/	system log files
archie/logs/archie.log	main Archie system log file
archie/logs/email.log	log file for email client (if installed)
archie/manpages/	Archie system manual pages
archie/pfs/	Archie/Prospero system home
archie/pfs/bin/	Archie/Prospero binaries
archie/pfs/pfs.log	Prospero log file

The directories `db/tmp`, `db/anonftp_db` and `db/webindex_db` will likely use the most space. Previous versions of the Archie system required that those directories be part of the same disk partition. The current version removes that restriction. The only limitation still present, in this version, is the requirement that `db/tmp` and `etc` still be in the same partition. The catalog directories may reside in separate partitions. 3.5

5.2 Archie user codes

Two user codes need to be created to enable Archie system operation and maintenance. The first, called *archie*, is used for non-administrative access to the Archie system (e.g., for accepting user queries to the Archie databases). The second, called *archuser*, is for system maintenance by the Archie system administrator. The password file entries for these two users should specify, in the home directory field, the full path to the root of the Archie system. The password file entries for these two codes also should meet the following requirements:

- both codes must have the same UID (User ID) number
- NEITHER code should have any special privileges, such as belonging to the wheel or staff groups
- the archie user code must appear *before* the archuser code in the password file
- the archuser code should have whatever shell you normally use (e.g. /bin/csh or /bin/ksh)
- the archie user's shell field should be set to /bin/false for the time being. This will be changed if a telnet client is installed (See "Setting up the anonftp catalog clients" on page 9.)
- the password field for archie should be set to asterisk ('*') for now. This will be changed if you install the Archie telnet client (See "Setting up the anonftp catalog clients" on page 9.)
- The password field for the archuser code should be set to one of your choosing
- The `/etc/group` file must be modified to create a group named "archie" for the Archie user codes. For example:
  `archie:*:1000:archie,archuser`
- both codes must have the same GID (Group ID) number in the password file, and this must be the number, set above, for the archie group.

Thus, an example of the entries in the `/etc/passwd` file is:

```
archie:*:23:1000:archie client code:/usr/local/archie:/bin/false
```

```
archuser:ufUn9Ap.:23:1000:archie admin code:/usr/local/archie:/usr/bin/csh
```

If you are using the Network Information System (NIS) you should only need to add the above entries to the /etc/passwd file for the local Archie host, not for the entire NIS system.

The naming convention of ~archie or `<archie home>` is used here to specify the home directory of the Archie system.

5.3 Unbundling the distribution

This new version of Archie contains not only a new set of binaries and a new catalog, but certain elements, internal to the system, also have been modified. If you already have an existing Archie installation, the catalogs will have to be rebuilt. Fortunately, you can convert the existing host_db database to the new format. 3.5

If you are upgrading your software from version 3.3 and want to save the host database, you will need to make a backup copy of the directory `~archie/db/host_db`. Be sure to read section 5.5 before proceeding with this. Once you have installed all the software, restore the host database then run the program `convert_hostdb`, as follows: `% convert_hostdb` The new host database will then be converted and will contain only those sites for which you are responsible (i.e. that you retrieve).

3.5

First start by FTPing the distribution from ftp.bunyip.com. In the directory archie-3.5 you will find the files that need to be retrieved.

- archie-3.5-install.tar
- archie-3.5-arch-version.tgz
- archie-3.5-arch-version-A.tgz
- archie-3.5-arch-version-B.tgz

where “arch-version” is one of: ‘sunos-4.1.4’, ‘sunos-5.4’, or ‘aix-3.2’.

Untar ‘archie-3.5-install.tar’ with the command:

```
% tar xvpf archie-3.5-install.tar
```

There you will find three scripts that will be used to install the binaries, ‘unwrap’, ‘untar’, and ‘unrotate’. The single command, ‘unwrap’ should take care of installing the binaries and other files.

To install the server software, as superuser, type:

```
% ./unwrap
```

and follow the instructions. As indicated by the script, you need to set the correct permissions. If you want to install the Archie man pages in a system-wide area, read the next section. As superuser type:

```
% cd ~archie/config
% make
```

This will change the permissions of the files in the Archie system to the appropriate values, as well as install the manual pages into the directory specified above.

In addition, a symbolic link will be created from the root directory pointing to the Prospero tree. Typing

```
% ls -al /pfs
```

will show you the link and the directory to which it points. It will look similar to:

```
lrwxrwxrwx 1 root 33 Mar 2 23:28 /pfs -> /usr/local/archie/pfs
```

It is suggested that the installation and configuration be performed as the user archuser except for those changes that must be made as the superuser.

5.4 Manual Pages

The on-line manual pages are, by default, configured to be installed in the Archie man directories `~archie/man`. However, you may opt to install the man pages in a different location, as described below.

First `cd ~archie/config` and edit the file `Makefile`. Modify the line:

```
MAN=../man
```

to be the name of the directory where you would like the Archie system manual pages placed, if it is different from the existing location.

Change the line:

```
MANEXT=n
```

to reflect the manual page name extension for that directory.

For example, if you wanted to store the Archie manual pages with the system man pages, you could change the appropriate lines, as follows:

```
MAN=/usr/man/mann
MANEXT=n
```

If the manual pages are placed anywhere other than your system's default manual directories, you may have to change your UNIX environment variable MANPATH to reflect the changes. Otherwise, the `man` command will not find the Archie manual pages.

5.5 ndbm files

A word of warning is in order about some of the files in the Archie system.

Several of the files created by the Archie system are created using the ndbm(3) library and can be identified as paired files with the .dir and .pag filename extensions. These files are considered sparse, in that the size stored in the file system inode (and reported by the ls(1) program) is not a true indication of how much physical disk space is actually occupied. The files contain "holes" that are not physically allocated. See the ndbm(3) manual for further details. There are two main consequences to this:

- The files must not be copied by standard means such as cp(1) and tar(1), although on initial inspection this may appear to work. The dump(8) program on SunOS 4.1.1 and higher allows individual files to be copied onto disk or tape. This program should be used when copying or transferring such files.
- The only true indication of the file size is obtained from the du(1) program. Sometimes ls(1) will report that the size of the file is larger than the partition, but this can be ignored. In addition, the .dir files will often be of zero length until the database has sufficient entries.

The freely available utility gtar (similar to tar) is capable of properly copying, archiving, and handling ndbm files.

5.6 Done !

You have now finished the initial installation of the Archie system. The next section will describe how to configure the system. You may want, first, to reread the overview section 3 to understand the different aspects of Archie.

Chapter 6

Configuring the Basic System

Now that you have an understanding of the basic structure of the Archie system, you are ready to continue with the configuration procedure. The sections in this chapter deal with the necessary installation fine tuning that will allow you to configure your Archie system to behave appropriately in your environment.

6.1 Configuring the pseudo-domains database

A pseudo-domain is an alias for a set of domain names. Pseudo-domains are useful for describing a collection of Domain Name System (DNS) domains which may not apparently be related (e.g., by spelling). For example, ".usa" may be defined as ".mil", ".com", ".edu" and ".us".

The Archie system makes use of pseudo-domain definitions. This is primarily for convenience in specifying domains as catalog search constraints. The files `~archie/db/host_db/domain-db.{dir,pag}` contain the information where these definitions are stored. It is the responsibility of individual Archie system administrators to configure these files as necessary. Note that this database is visible from remote Archie clients (for example in the anonftp telnet-client the domains command allows users to see the domain structure of a server). Users may also search within the catalogs using these pseudo-domain definitions as restriction criteria.

The tool for modifying this database is the ardomains program, found in ~archie/bin. The input to this program is a file following the format in the example (see below). The example and the default configuration file are provided as illustrations of the appropriate format. Some modification will be necessary to tailor the setup to your Archie server environment.

```
#This is a comment line
europe         de:ie:pt:es:uk:at:fr:.il:be:nl:ch   Europe
scandinavia    no:dk:se                            Scandinavia
northamerica   edu:com:gov:us:ca:mil               North America
asia           kr:hk:sg:jp:cn:my:tw:in             Asia
world          europe:scan:noram:asia              The world
```

In the example, the pseudo-domain europe has been defined to include the root (top-level) DNS domain names (ISO codes) of the countries of Europe. Pseudo-domain definitions can be nested. So, for example, world is composed of the pseudo-domains europe, scandinavia, northamerica and asia. Short descriptions (265 characters or less) of the pseudo-domains may optionally follow their definition.

When determining if a specific site is a member of a particular pseudo-domain, the Archie system iteratively resolves each pseudo-domain into its base constituents. Thus, world is broken down as follows:

```
europe:scan:noram:asia
```

and in turn, europe is taken as:

```
de:ie:pt:es:uk:at:fr:.il:be:nl:ch
```

The Archie system then searches to see if any of these are, in turn, pseudo-domains. This process can be nested to 20 levels.

If an entry cannot be resolved into subcomponents, it is taken "as is". Thus for example, the pseudo-domain uquebec could be arbitrarily defined as

```
uquebec mcgill.ca:uqam.ca:concordia.ca:umontreal.ca Universities in Quebec
```

This provides a list of the various domains within the pseudo-domain of universities in Quebec. The mcgill.ca string cannot be further resolved by the system, so it is considered a base component.

This technique gives both the Archie administrators and users a shorthand notation for specifying domains.

The following points should be kept in mind:

- Pseudo-domains must be defined before being used in other pseudo-domains.
- On final resolution, the names must match a real DNS domain. The Archie system makes no attempt to verify the authenticity of the base domains: they are just used for comparison with site names. A non-DNS name will never successfully match any data host names stored in an Archie catalog.
- The Archie system carries out comparisons of domain strings from right to left. Thus, m.ca matches both crim.ca and uqam.ca. This can be used to advantage in constructing domain specification strings. In the example above, the .il entry differs from the rest by having a preceding period ("."). This is so that sites in Israel (".il") do not match US military sites (".mil").

6.1.1 Loops in the domain database

The system will detect domain "loops". For example, the entries

```
canada qc.ca:on.ca Provinces in Canada
on.ca canada Ontario
```

form a loop: canada resolves into on.ca and qc.ca; on.ca refers back to canada. Archie administrators should try to avoid introducing such loops in their specifications. The ardomains program will identify any it detects.

6.1.2 Building the domain database

To create the domain database, run the program ~archie/bin/ardomains on a file containing entries as specified above. The syntax is:

```
% ardomains [<input file>]
```

The `<input file>` should reside in `~archie/etc` for future reference by Archie programs. By default, the file ~archie/etc/ardomains.cf is used if no file is explicitly given. A sample file is provided in the distribution for your convenience. Please consult the manual pages for full details on using this program.

A backup copy of the domains database is maintained in the `~archie/db/host_db` directory. Thus, the system is never without a domain database, even when ardomains is completing the update procedure on the domain database.

6.1.3 Files used by ardomains

File Name	Type	Description
~archie/bin/ardomains	Executable	ardomains program
~archie/db/host_db/domain-db.{dir,pag}	ndbm data	main domains database file
~archie/db/host_db/domain-db-new.{dir,pag}	ndbm data	active domains database file (hard link to above)
~archie/db/host_db/domain-db-old.{dir,pag}	ndbm data	previous version of domains database

6.2 The arserver program

The arserver program provides the external Internet world with a view of the local Archie Host Information Files. It is responsible for responding to queries about the Data Host information as well as performing the protocol negotiations necessary for inter-Archie data exchanges. The server is designed to be run by the UNIX inetd process and thus the file /etc/inetd.conf needs to be modified to inform inetd about the new process.

> You must be logged in as superuser to carry out these changes.

The following single line must be added to the /etc/inetd.conf file with the correct entry for ¡archie home¿ (the Archie home directory):

```
arserver stream tcp nowait archuser <archie home>/bin/arserver arserver -l
```

The following line should be added to the /etc/services file:

```
arserver 2710/tcp
```

> 2710 is not a reserved port number. Future releases will probably make use of a reserved port.

> If your system is running the Network Information Service (NIS, formerly called Yellow Pages), additional steps must be taken to incorporate the new services file in your system. Please refer to NIS documentation for details.

Once these changes have been made, inetd must be notified. The easiest way to do this is to restart the inetd process. In SunOS 4.X, this can be accomplished as follows:

```
% ps aux | grep inetd
```

This will extract a line specifying the inetd process, which will look like:

```
root 162 0.0 0.0 52 0 ?  IW Oct 23 0:30 inetd
```

The process number (<procnum>) is the second field (in this case 162).

Then issue the command:

```
% kill -HUP <procnum>
```

This tells the inetd process to re-read the (modified) configuration file.

For an RS6000 system, however, the following command will perform the necessary updates:

```
% inetimp; refresh -s inetd
```

6.3 Invoking arserver

There are three programs associated with transferring data site information between remote Archie servers, and internally within your own local Archie server. They are:

- arserver
- arexchange (inter-Archie exchange)
- arretrieve (local Archie updates)

> arexchange and arretrieve are currently soft links to the arserver program; the different names serve to distinguish between tasks to be accomplished.

The arserver program is designed to run on demand as connections are made to it. The arexchange and arretrieve programs, however, are to be run periodically. This process can be automated through the use of cron (see the UNIX manual entry cron(8) for details).

All entries for cron should be made in the archuser crontab file, since archuser is the administrative code for the Archie system. Failure to do so may cause problems with improper file permissions for the data files and executable programs.

All examples in this section will refer back to this sample file. The arcontrol program will be described in greater detail in "Controlling the Update Cycle" on page 28.

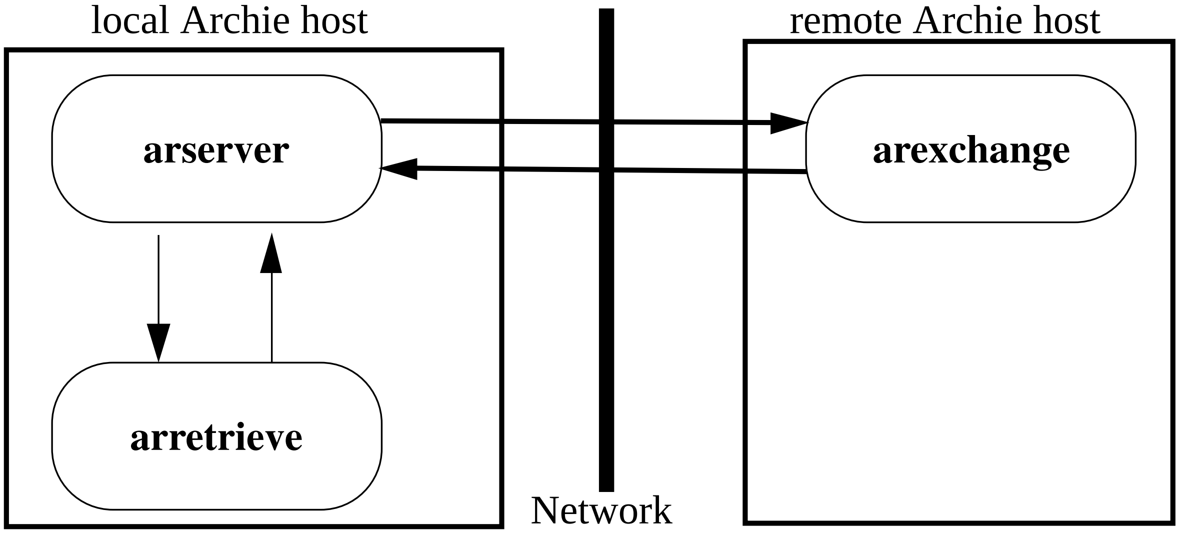


Figure 6.1: arserver, arretrieve and arexchange relationships

6.4 Configuration

The arserver, arexchange, and arretrieve commands make use of two configuration files, with which the Archie system administrator can configure such things as the data sites that are to be inspected for updating Archie catalogs (in arretrieve.cf), or the remote Archie servers that are permitted to participate in an exchange of information (in arupdate.cf).

arupdate.cf is consulted by both arserver and arexchange, while arretrieve.cf is consulted by arretrieve. Both reside in ~archie/etc. Both files have the same format, but different semantics.

Please note that both configuration files are overwritten by the process accessing them (arretrieve.cf by arretrieve and arupdate.cf by arexchange) at the end of processing. Changes written to the configuration files by the administrator while the processes are running will be lost.

The Archie site administrator is responsible for configuring these two files so that the system performs the desired local data retrievals, in addition to remote exchanges with other Archie hosts.

6.5 Configuration decisions

There are a number of factors to be considered when performing the configuration:

- What data do you want to store in your local Archie system? This means deciding which catalogs you would like to serve and from which Data Hosts you want to pick up the information. For example, you may decide that you would like to serve all anonymous FTP sites (the anonftp catalog) in Nigeria, Mongolia and Barbados or provide a webindex catalog for all of Europe.

- With whom do you want to exchange inter-Archie data? If you are in a remote location from the data that you would like to serve to your user community, you will probably find that somebody closer is already collecting this data. It is possible that they will be interested in exchanging data with you as well.
- How often does the data, that you are going to collect yourself or exchange with a remote Archie site, change? There is no need to exchange information frequently when the data itself changes infrequently. This also wastes network bandwidth that could be better utilized.
- With the current version of the Archie system, the minimum amount of time between inspections of a Data Host must be greater than the amount of time required to complete the Update Cycle from the last access. This is because the Host Information Files are updated as the last step of the Update Cycle, so the information that would be used to start the next Update Cycle is not yet available.

6.6 Choosing Data Hosts

The worldwide Archie system generally works as a set of independent, cooperating providers, serving the Internet community. While most Archie servers are primarily interested in serving their local community on the user query side of the system, the data host information exchanged between servers allows each system to provide global data, while only having to monitor a relatively small number of local data hosts. Each server is free to collect data directly from any data host desired, however cooperation in the general data exchange network benefits all concerned and is encouraged.

You may choose to gather data from Data Hosts directly, while obtaining the information about other data hosts from other Archie systems. You should decide which data hosts, if any, you are interested in contacting directly for information and which you would like to obtain from remote Archie servers. Once this decision is made, you can configure your system appropriately.

Bunyip Information Systems coordinates Archie exchanges among its customers as part of the annual support agreement.

The Archie administrator has significant control over the criteria used to determine what is updated and how often. For example, the Archie system can be configured so that nearby sites are inspected every day while more remote sites are checked every two days. Alternatively, it can be configured so that certain catalogs are updated on even days, and others on odd days. Inter-Archie data exchanges can be configured in a similar fashion.

The next sections will describe the precise format and semantics of the configuration files.

6.7 The Retrieval Phase

As mentioned earlier, the Retrieval Phase is the first step in the Update Cycle. The local Archie Host Information Files are queried through the use of the arretrieve command. The query is composed of a set of criteria (or constraints) that the specified Data Hosts need to meet in order to be inspected. (For a detailed description of the protocol used between arretrieve and arserver, see the manual page archie_protocol). arretrieve then requests that header records be obtained for any Data Hosts that meet the specified constraints. These are deposited in the holding directory, where they await the next phase of the process (refer back to "Header Records" in Appendix A for more information about Archie system headers).

The queries that arretrieve sends to arserver are based on the information that the Archie administrator has set up in the arretrieve.cf file.

```
<the_entry>:= <arserver host> <config line>

<config line>:= <config line> <config> | <config>

<config>:= <db list> <domain list> <max no> <perms> <freq> <date> <fail>
```

where

`<arserver host>`: is the Fully Qualified Domain Name (DNS) of the host running the arserver program, to which the arretrieve client will connect. arretrieve and arserver may be run on different machines. However the majority of the time this entry will be localhost.

`<db list>`: is the colon (“:”) separated list of Archie catalogs which are to be retrieved.

`<domain list>`: is the colon (“:”) separated list of domains and pseudo-domains which the Data Hosts need to match in order to be returned.

`<max no>`: is the maximum number of sites returned matching the criteria.

`<perms>`: Permissions (‘r’ead or ‘w’rite). Must be ‘w’ to enable transfer. A value of ‘r’ effectively converts this line into a comment: it will be read but no action will be taken.

`<freq>`: A number specifying the minimum number of minutes that are to elapse before contacting arserver with this query again. It may have the modifiers ‘h’ or ‘d’ immediately following the number, specifying that the value is in units of hours or days respectively.

`<date>`: Date in YYYYMMDDHHMMSS format. Typically, this specifies the last time that the program performed such a query. In fact, the date is determined by the date of the most-recently inspected site. This is explained more fully in “Dates in the configuration files” on page 43.

`<fail>`: 0. Ignored by arretrieve.

The character “ in the configuration file indicates that the entry is continued on the next line of the file.

The following example illustrates the format:

```
localhost anonftp mcgill.ca:uqam.ca 30 rw 12h 19920703162322 0\
      whois:yp bunyip.com:cndir.org 30 rw 30d 19920603162322 0
```

This can be explained as follows:

Field Content	Description
localhost	Contact the arserver program running on the same host as the arretrieve program. This is the case in almost all Archie installations.
anonftp	specifies that information for the anonftp catalog is to be inspected. Alternatively a ‘*’ may be given, signifying all catalogs.
mcgill.ca:uqam.ca	specifies that they are only to be from sites in the mcgill.ca and uqam.ca domains. Any pseudo-domain definition may also be given. Alternatively, a ‘*’ may be given signifying all domains.
30	specifies that at most 30 sites are to be returned on the retrieval.
rw	The ‘w’ specifies that this line is “enabled”. The ‘r’ is ignored.
12h	specifies that the program should contact arserver every 12 hours
19920703162322	specifies the last time arretrieve queried arserver about this `<database list>/<domain list>` combination. For the initial configuration, this should be set to your current date and time. The client queries the server for “all sites that have not been inspected since `<date>`”.
0	is ignored by the client and should be set to 0

The “ signifies that the entry is continued on the next line.

The second (continuation) line gives a configuration for catalogs called whois and yp.

In this way, the Archie system administrator can completely configure how often sites are to be inspected, with full control over what is returned. For example, if one wanted to be responsible for (and query) all Canadian and American sites every day for anonymous FTP information, but European sites only once a week, the configuration could be:

```
localhost anonftp ca:usa 40 rw 1d 19920810162322 0 \
```

```
anonftp europe 30 rw 7d 19920810162322 0
```

This example is only an illustration. In most cases, one would not be responsible for such a large region directly, but would pick up such information through data exchanges with other Archie servers.

The example illustrates a case where the Archie administrator wants to inspect data hosts less frequently than those on the same continent as the Archie server. Specifically, anonymous FTP archive sites within Canada and the United States are to be inspected every day, while those in Europe are to be inspected only every 7 days.

arretrieve is designed to be run from the cron(8) daemon with a frequency no less than that specified by the shortest interval in the configuration file frequency field.

In the above example, the shortest interval specified is 1 day. The following line would be appropriate in the crontab(5) file.

```
15 2 * * * <archie home>/bin/arretrieve -l -M <archie home>/db
```

This specifies that cron is to run the arretrieve program at 2:15 every morning. All appropriate inspections will be started up at this point. In the example given, North American sites will be inspected each day, and the European sites will be inspected only once every 7 days. ¡archie home¿, is of course, the Archie home directory. See crontab(5) for full details on the format of the file.

The arretrieve program does not retrieve any headers.

- For initial configurations ensure that the date field in the arretrieve.cf file is set to the current date.
- The first field of the arretrieve.cf file, in almost all cases, will be the special name localhost. Check to make sure that this is the case in your configuration file.
- Re-check the configuration file to be sure that the syntax is correct and that domains and catalog names can be resolved.
- Check that arretrieve is a soft link to the arserver binary and that the archuser code has read/execute permissions on the file.
- Check that the arretrieve.cf file is has read/write permission for the archuser code.

6.8 Inter Archie data exchange

The local arserver program will accept connections from remote arexchange clients. Similarly, the local arexchange program can contact those remote Archie arserver programs that give it permission. These permissions are defined in the configuration file. The default configuration file is ˜archie/etc/arupdate.cf. As noted earlier, this configuration file has the same format as arretrieve.cf, but the semantics are different.

Through the arupdate.cf configuration file, you can set up both symmetric and asymmetric exchange procedures. The symmetric relationship occurs when the connections to your Archie peers are bi-directional: you retrieve data from them and they from you. The asymmetric relationship occurs when either you only retrieve data from them (and not they from you) or vice versa. This is controlled through the use of the `<perms>` field, described below.

Arrangements for the client/server connection must be enabled by the administrators on both sides of the connection for it to be successful. Bunyip regularly posts contact information for Archie servers worldwide on its support mailing lists.

6.8.1 The arupdate.cf file as seen by arserver

For arserver, the configuration file provides a simple but effective security mechanism which allows only those sites who have previously been given permission, by the local administrator, to contact that server.

The fields in the configuration file have the following meanings to arserver:

`<client host>`: is the Fully Qualified Domain Name (FQDN) of the remote Archie host from which the arexchange client will connect. This may also be an asterisk ("*") specifying that any remote Archie client may connect.

`<db list>`: is the colon (":") separated list of Archie databases that the client is allowed to retrieve.

`<domain list>`: is the colon (":") separated list of domains and pseudo-domains that the Data Hosts need to match in order to be returned. These need to be defined in the local server domains database.

`<max no>`: is the maximum number of sites returned matching the criteria.

`<perms>`: Either 'r' or 'rw' specifying the operations the remote arexchange client may perform. If only 'w' is given, the client is not allowed to connect to the server.

`<freq>`: Ignored.

`<date>`: Ignored.

`<fail>`: Ignored.

6.8.2 The arupdate.cf file as seen by arexchange

arexchange, acting from the information in the configuration file, contacts remote Archie servers and queries them about new data which may have been received since contact was last made. If new data exists, the servers negotiate its transfer. It is then placed in the appropriate phase of the Update Cycle and processed identically to the data acquired through the local arretrieve process.

Either the local or the remote Archie system may request that the data being transferred be in a compressed format. This greatly improves throughput in the system and is strongly recommended. See the arserver and arexchange manual pages for full details.

arexchange is designed to be run from the cron(8) daemon by having an entry in the crontab file. The frequency of its invocation should be no less than the shortest interval specified in the configuration file.

The entries in arupdate.cf have the following meanings for arexchange:

`<server host>`: is the Fully Qualified Domain Name (FQDN) of the remote Archie server. CNAME records (nicknames) may be used.

`<db list>`: is the colon (":") separated list of Archie catalogs that the client is to retrieve.

`<domain list>`: is the colon (":") separated list of domains and pseudo-domains that the Data Hosts need to match in order to be returned.

`<max no>`: is the maximum number of entries returned matching the criteria. The value 0 is a special case and tells the program to accept as many incoming entries as the remote server can provide.

`<perms>`: Either 'w' or 'rw' specify the operations that may be performed. If only 'r' is given then the local arexchange client is not allowed to connect to the specified remote server (`<serverhost>`). That is, `<serverhost>` may 'r'ead local Archie information, but not 'w'rite.

`<freq>`: The frequency with which the client is to contact this server about the given domains and catalogs (given above). It may have the modifiers 'h' or 'd' immediately following the number, specifying that the value is in units of hours or days respectively.

`<date>`: Date in YYYYMMDDHHMMSS format. This is explained more fully in below.

`<fail>`: The number of failures experienced while trying to contact this server. The system generates an error condition for the Archie administrator if this number rises above a pre-determined threshold.

The following is an example of an arupdate.cf file

```
* anonftp ca 0 rw 1 19930321103513 0
archie.rutgers.edu anonftp com:mil:edu:net:us:gov 100 rw 1 19930511155204 0
archie.sogang.ac.kr anonftp asia 0 rw 1d 19930526154304 0
archie.univie.ac.at anonftp westeurope:easteurope 0 rw 1d 19930526200058 0
archie.uni-linz.ac.at anonftp westeurope:easteurope 0 rw 1d 19930330133007 0
archie.switch.ch anonftp westeurope 0 rw 1d 19930524082005 0
aries.cc.deakin.oz.au anonftp au 0 rw 1d 19930512124502 0
archie2.th-darmstadt.de anonftp de 0 rw 1d 19930524003041 0
archie.luth.se anonftp se:dk:no:fi 0 rw 1d 19930527033159 0
```

The arserver program will not run until the host databases (in ~archie/db/host_db) have been created. See the section "Managing the Host Information Files" on page 33 for information on how to do this.

6.8.3 Dates in the configuration files

The initial dates for the configuration files should be set as follows:

For the arupdate.cf file set the initial date to all zeros. That is,

```
anonftp se:dk:no:fi 0 rw 1d 19700201010001 0
```

For the arretrieve.cf file, set the initial date to the current date and time. For example,

```
anonftp se:dk:no:fi 0 rw 1d 19940422011500 0
```

The times should all be in GMT, but the system will still work correctly if you just give the time in your local timezone.

As the system is primed you may not get the maximum number of sites returned until various components in the system synchronize. Running the programs a couple of times will allow this to happen.

When arretrieve and arexchange contact the arserver program, they indicate the maximum number of sites they want returned, matching the given criteria. For example, arretrieve might ask for a maximum of 30 sites to be identified for inspection, when there are in fact 40 that match the criteria. In this case, the sites are listed in decreasing order of time since last inspection (i.e., least-recently inspected, first). The date that is stored in arretrieve.cf will be the date of the last inspection of the 31st site. This ensures that the next time arretrieve is run, the sites that were skipped will be inspected. Typically, this is only an issue when an Archie server is first brought on-line, and has much catching up to do in terms of inspecting sites. The is true for the arexchange program, too.

When the number of sites matching the criteria is less than the maximum number allowed, the date field accurately reflects the date of last inspection.

6.8.4 Updating by "proxy"

The arupdate.cf file can be configured to allow "proxy" exchanges of information. For example, not every North American Archie server need exchange information with every European server. The global Archie network can be configured so that one site in North America is responsible for picking up information from a particular European site. All other North American sites can be configured to get the information from the delegated proxy server, instead of the European server. This would reduce duplication in inter-continental net traffic. The can be done on a catalog-by-catalog basis.

You may, for example, use the following line in the arupdate.cf file

```
archie.foo.com anonftp northamerica 0 rw 1d 19930526200058 0
```

even if the server, archie.foo.com, is not responsible for the northamerica domain. You can do this if you have arranged with the administrator of this Archie site that, in addition to the data hosts for which that site is directly responsible, you would also like to obtain sites in the northamerica domain.

6.9 Controlling the Update Cycle

The arcontrol process (described in "The Update Cycle in detail" on page 11) is responsible for coordinating the entire Update Cycle. It too is invoked in different modes depending on the required functionality.

- With the -r command line switch, it runs in retrieval mode, coordinating the Data Acquisition Component of the Update Cycle.
- With the -p command line switch, it runs in parse mode, performing the necessary functions to parse the waiting data files.

- Finally, with the -u command line switch, it runs in update mode. The data is finally entered into the appropriate Archie user catalogs and the system's Host Information Files are also modified to reflect the new information.

In all cases, arcontrol's main task is to scan the holding directory for data files of appropriate type for the particular Cycle phase, perform any necessary pre-processing (uncompressing data files for example) and invoking the correct program for that phase and that catalog. arcontrol is designed to be run from the crontab file of archuser. The relevant entries from the example crontab file are:

```
0 3,15 * * * <archie home>/bin/arcontrol -r -l

30 3,15 * * * <archie home>/bin/arcontrol -p -l

45 3,15 * * * <archie home>/bin/arcontrol -u -l
```

These entries specify that, at 3:00 and 15:00, arcontrol is to run in Retrieve Mode. Thirty minutes later it runs in Parse Mode, to process some, if not all of the data files retrieved in the previous phase. Finally at 3:45 and 15:45 it is run in Update Mode, the final step of the Update Cycle.

In Retrieval mode the processing is performed asynchronously and any number of copies of the arcontrol program can be running.

6.10 Temporary and Lock files

The current version of arcontrol performs all pre-processing on data files first, before invoking the appropriate subprocesses. These files have the "temporary" suffix designator "_t". In order to prevent concurrent update of any particular catalog, a series of lock files are created. These files reside in the ~archie/db/locks directory and are called:

```
<catalog_name>_lock
```

These files remain for the duration of the update process and contain the name of the process creating them, the time created and the name of the datafile being processed. They will be removed with the successful completion of the update. The lock file will not be removed if either of the following situations occur:

- The update process which created the lock file is killed or aborts prematurely. An abort can occur if subprocesses performing the update themselves are terminated or aborted.
- The host system is rebooted or crashes, thereby prematurely terminating the update process.

6.11 Host shutdowns and crashes

Abnormal termination of the update procedure can occur if the host system crashes or is rebooted during the update. If this happens one should do the following:

- Check the integrity of the catalog in question. Depending on the type of catalog, various programs may be available to do this. See the chapters in this manual, describing the individual catalogs, to determine the exact procedure appropriate to that catalog.
- If the catalog is corrupted and can be fixed, apply the appropriate procedures for that catalog to effect the repair.
- If it is not corrupted (or has been fixed), remove the lock file.
- If you would like to reprocess data files that did not get processed because of the abnormal termination, then the update_t files should be renamed without the "_t" suffix. They will then be processed during the normal Update Cycle.

It is the responsibility of the Archie system administrator to recover from a crash.

6.12 More disk/less time

System administrators should take note of the “-U” command line option for the arcontrol program.

In order to conserve disk space, arcontrol normally compresses the data in the files processed during the Update Cycle. If you have enough disk space, you may achieve a significant speed-up by specifying this option, which disables the compression and decompression of the data during processing. See the arcontrol manual page for further details.

6.13 Monitoring the System

To assist in the process of Archie system administration, the system can be configured to mail regular reports describing its activities. This is accomplished by creating a file called mail.results in the ~archie/etc directory. This indicates to the system that it is to keep track of its activities in special files also located in the ~archie/etc directory. These are mail.fail, mail.success, mail.add, mail.retr, mail.parse and mail.delete. In order to have the results mailed back to you, edit the file ~archie/bin/mail_stats and, if necessary, change the line:

```
MAIL_PGM=/usr/ucb/mail
```

to reflect the full pathname of the mail program you would like to use.

Whichever executable is used to define `MAIL_PGM`, it must accept a “-s” command line switch with the subject of the mail as its parameter.

Also, the line:

```
ARCHIE_USER=archuser
```

should describe the valid address to which the results should be mailed. Finally, an entry in the archuser crontab file must be made, in order to run the mail_stats program (a frequency of once a day should suffice).

If you are not receiving any mail from the system.

- Check the archuser crontab file to ensure that the mail_stats program is being run.
- Check that your `MAIL_PGM` variable is correctly defined and is a fully specified pathname.
- Check that the address to which the program is sending mail is correct, by manually mailing a test message to that address and verifying tha it arrives.
- Make sure that the update cycle is in fact running, by checking the crontab entries and log entries, as well as performing manual invocations of the arcontrol program.

The tool “daily-admin” is provided for managing the various log files. The purpose of this tool is to control the amount of disk space used by the logs, as well as reporting statistics to the maintainer of the server and to Bunyip. The package “daily-admin” compresses the log file and copies it to the logs subdirectory “history”, keeping a certain number of history logs on the system.

Note that the script will now process all files that have the extension `.log` present in the directory `~archie/logs`. Also an extension having the format `YYMMDD` is used to archive the files. 3.5

The “gzip” program is used to compress the log files. If it is not found the “compress” program is used instead. It handles the logs in ~archie/logs and ~archie/pfs/pfs.log differently. In the case of archie.log and email.log, the package compresses the log file and keeps the last 32 copies on the system. In the case of pfs.log, the logs are separated by day, and each day’s information is compressed. This information is kept for 6 months. Since the programs consist of shell scripts and Perl scripts, these values may be changed.

We hope that you will use this tool to help you manage your site and at the same time help us to maintain statistics on Archie usage, as well as follow the progression of your site. You should add the following line in the archuser user’s crontab.

```
# For logs and stats
5 0 * * * <archie-home>/scripts/daily.admin
```

The script should run at around midnight local time. You may use a “-c” flag to send also a copy of the stats to an additional email address, as shown below.

```
# For logs and stats

5 0 * * * <archie-home>/scripts/daily.admin -c you@your.domain.name
```

6.14 Administrative addresses

Archie site administrators need to set up an administrative account for bug reports, comments, suggestions, additions and deletions to the catalogs, etc. Make an entry for archie-admin in the `/etc/aliases` file, pointing to the person or persons responsible for the administration of the server. For example:

```
archie-admin:  myaddress@your.domain.name
```

6.15 Moving right along

We've talked about what you'll need to do to configure the retrieval and exchange components, now we'll take a look at the setup to enable users to use the catalogs.

Chapter 7

Managing the Host Information Files

The curses(3X) based program, host_manage, allows the Archie administrator to examine and/or change the contents of the Host Information files. These activities include the addition and deletion of Data Hosts, as well as providing information about specific catalogs that are updated from the Data Host sites. Invoked as:

```
% host_manage [<site name>]
```

it will display the information about the given `<site name>`, or the first entry in the Host Information File (alphabetically sorted) if no `<site name>` is given. The screen display is illustrated below.

```
Primary hostname: blizzard.cc.mcgill.ca
Preferred hostname: www.mcgill.ca
Databases: webindex
Operating System: Unknown                Prospero Host: No
Primary IP address: 132.206.27.11        Timezone: -5:00


Database: webindex


Source: archie.bunyip.com                 Generated by: insert
Retrieve time: 19:02:34 31 Mar 1996 EST   Number of records: 136
Parse time: 17:06:00 25 Apr 1996 EST
Update time: 05:47:23 26 Apr 1996 EST     Fail count: 0


Msg:


Port: 80                                  Path:

Current status: Active
Force Update: No
Action: Update
```

The commands for editing the information are based on the standard emacs key bindings. These will be configurable in a future release of the system.

The configuration file associated with the host_manage program is ~archie/etc/hm_db.cf. It describes how the various catalog-specific information is to be displayed on the screen. Currently, the default file handles anonftp and webidnex catalog entries. This file need not change unless you add new catalogs.

Many of the fields displayed by the program are informational only and may not be modified directly by the Archie administrator. Others may be modified, but have a restricted range of values (such as the Operating System value).

In all cases, the information given by the Archie administrator is verified against the existing information file, as well as the Domain Name System. Any errors that are encountered will be indicated.

Only the major functions of the program are described in detail below. For full details on how to configure and use host_manage, please see the manual page entry.

There is a known problem with the host_manage program in its interaction with certain terminal types, particularly xterms. This can be fixed by removing the "me" attribute in the distributed /etc/termcap entry for the affected terminals. For example, in the xterm entry remove the string:

```
:me=\E[m:
```

So far, to our knowledge, this fixes the problem on all terminal types.

7.1 Adding individual Data Hosts

The easiest way to add an individual Data Host to the Host Information Files is to use the host_manage program. Adding a site to the database is a straightforward operation.

Quite often, someone will request that you add a site, and will specify the name as something like ftp.xxx.yy.zz. This name is probably not the primary hostname, but rather an alias for the primary hostname. The host_manage program requires that the host be stored under the primary hostname, with an optional preferred hostname.

Use the various DNS tools, such as host, nslookup, or nstest to determine the primary name before proceeding. In general, we recommend that you check all names before starting.

Using host_manage, do the following:

- Hit the `<TAB>` key, to bring up the Host Database Modification Menu.
- select item "Add a site to the host database".
- The message Add site will appear at the bottom of the screen and a blank template will be displayed.
- Fill in the various fields:
  - Primary hostname must contain the primary ("canonical") name of the host.
  - Preferred hostname can be the alternate name of the host, typically an alias of the primary name. Leave this blank if none is desired.
  - Operating System. This field is used by the parser. Currently only three values are possible. Hitting the spacebar cycles through these. If the site is not a VMS site, select Unix BSD.
  - Prospero Host. Leave this as "No". This is reserved for future releases.
  - Timezone. An optional field specifying the offset, from GMT, of the host being entered.
  - Database. The name of the catalog that you are entering for this host. This would normally be anonftp or webindex. Note that you can only add one database at a time.
  - Access commands. Leave blank for the moment.
  - Current Status. This field holds three values:
    - Disabled. This host is not polled.
    - Active. This host is actively being polled.
    - Inactive. The host is not in the database at the moment.
    - Not Supported. The host is currently unsupported for one reason or another (e.g. because of an incompatible operating system).
    - You will probably want to enter the host as "Active". Again just hit the space bar to cycle through the possible values.
- When you are satisfied with your entries, type `<Cntl-U>`. The system will ask you if you want to update the site, and then if you want to update the database. Answer "y" to both questions.
- The site has now been added and will be indexed during the next Update Cycle.

7.2 Deleting a Data Host

Before a site can be removed from the Host Database of the Archie system, all catalogs tracked on that host must be scheduled for deletion.

Using host_manage, do the following:

- Find the site you want to delete by filling in the Primary Database field then, typing the <return> key.
- Hit the <TAB> key to bring up the Host Database Modification Menu, then select the number corresponding to the entry "Delete an Archie database from current site".
- You will be asked if you want to delete the current database. Answer "y".
- The system should then report the message "Database <x> scheduled for deletion", where <x> is the current catalog selected and displayed on the line "Database:". The "Current status" line for that catalog will now display "Marked for deletion by Archie administrator".
- Repeat the procedure for all the catalogs for which this site is being polled. You can see this list by consulting the "Databases" field.
- Once you've marked all the catalogs for deletion, you will have to wait until the next Update Cycle runs, when all catalog entries for that site will be purged.
- As each catalog is removed from the system, the "Current status" line will display "Deleted".
- Only once this is done can you remove the site, from the Host Databases, using the "Delete a site from the host database". The site will then be physically removed from the Host Databases.

Often a site will offer multiple services, such as anonymous FTP and webindex. It is possible to add multiple catalogs for the same site with the following steps: 3.5

- Go to that site by typing in its name in the primary_name field.
- Hit <TAB> and select ''3. Add an archie catalog to current site''.
- Fill in the information and update with <Cntl-U>.

7.3 Forcing an early update

If, for whatever reason, you would like a particular site to be inspected before it normally would be scheduled, use the Force Update function of the Host Modification Menu in host_manage.

This will cause the particular Data Host/catalog pair to be retrieved and processed during the next Update Cycle, if it conforms to the domain and catalog restrictions of that Retrieval Phase. That is, this feature only overrides the time of the update, not the type.

7.4 And now the catalogs...

Now that we know how to look at the information stored in the Host Databases we can start to look at configuring individual catalogs.

Chapter 8

The anonymous FTP catalog

The Archie system is designed to maintain several different information catalogs, of various types. Nonetheless, it was originally conceived to maintain a catalog of files available by anonymous FTP, and this is still the application for which it most popular today.

Now that the arserver, arexchange, and arretrieve programs have been configured (according to the instructions in "Configuring the Basic System" on page 20), you can go about setting up the system to maintain an anonymous FTP catalog.

8.1 Overview

In general, to set up the anonymous FTP catalog, anonftp, the following steps must be followed:

- Make sure the configuration files listed in the previous sections have been modified to reflect your system and information retrieval needs. In particular, make sure that the domains database has been configured (see ardomains).
- Load the Data Host information into the Host Information Files. Use the host_manage program to add, delete and modify individual host entries. This is explained below.

8.2 Choosing Data Hosts to enter

You need only enter into the Host Databases, those Data Hosts for which you plan to be directly responsible. You need not (and should not) enter other sites. Once you start participating in the global inter-Archie data exchanges, those data hosts for which you are not responsible will be entered into your database automatically. Similarly, you need not delete those sites for which you are not responsible. The exchange subsystem will propagate this information from the "master" Archie system responsible for that site.

In general Bunyip maintains a policy to only index Data Hosts which are aware, and approve of the practice.

8.3 Support for ls-lR.gz

3.5

Archie can now retrieve from anonymous ftp sites pre-generated ls-lR.gz files In order to activate this you will need to setup the file `~archie/etc/options` in the following way.

```
#
#TYPE NAME PATH
#
#
COMPRESS GZIP /usr/local/bin/gzip
UNCOMPRESS GZIP /usr/local/bin/gunzip
```

Where `/usr/local/bin/gzip` is where the program `gzip` is located on your system.

You also need to fix the file `~archie/etc/arretdefs.cf` by replacing the line

```
anonftp:unix_bsd:image:.Z:anonymous:::-R:*?:ls-lR
```

by

```
anonftp:unix_bsd:image:.gz,.Z:anonymous:::-R:*?:ls-lR
```

Hence when using the `-Z` option in retrieve mode

```
% arcontrol -r -Z
```

Archie will try to first locate the ls-lR.gz file. If it can't it will look for ls-lR.Z, ls-lR in that order and as a last resort dynamically create the new listing.

8.4 Adding sites

You may want to start by entering one site into the Host Database (using host_manage) then testing the system before entering more sites. Once entered, you can manually run arretrieve, and arcontrol in its various modes (retrieve, parse and update) to get a feel for how the system operates.

8.5 Parser failures

When the parsing phase of the anonftp catalog fails on a particular data host the temporary parse file (with the parse_t suffix) is not removed from the holding directory (`~archie/db/tmp`). In addition, the filtered file is renamed with the suffix .filtered to allow the system administrator to see both the unfiltered and filtered versions. The system administrator may, if desired, manually fix the input data if desired.

It is the responsibility of the administrator to remove failed parsing files when finished.

The system provides the administrator with the approximate location of the parsing error and displays the line that caused the problem. This error can be viewed through the use of the host_manage program after the update phase of the cycle has been completed. Alternatively, the Archie log file contains a more detailed explanation of the error. However, as illustrated in the example Figure 5, the parse_t file is not the one actually parsed since the filter program first runs on the input. As a result, the error line generated is that from the output of the filter.

8.6 The parsing filters

By default, the distribution is configured to use perl language scripts for the filter_anonftp_unix_bsd filter (which is a soft link to the file `~archie/bin/filter_anonftp_unix_bsd.perl`). The perl interpreter is available on many anonymous FTP archive sites. If you do not have perl installed at your site, you can change this soft link to point instead to the file `~archie/bin/filter_anonftp_unix_bsd.sed`, which is an alternative filter based on the standard UNIX sed(1) program. This second filter is less efficient than the perl filter so we recommend that you install perl and use that in preference.

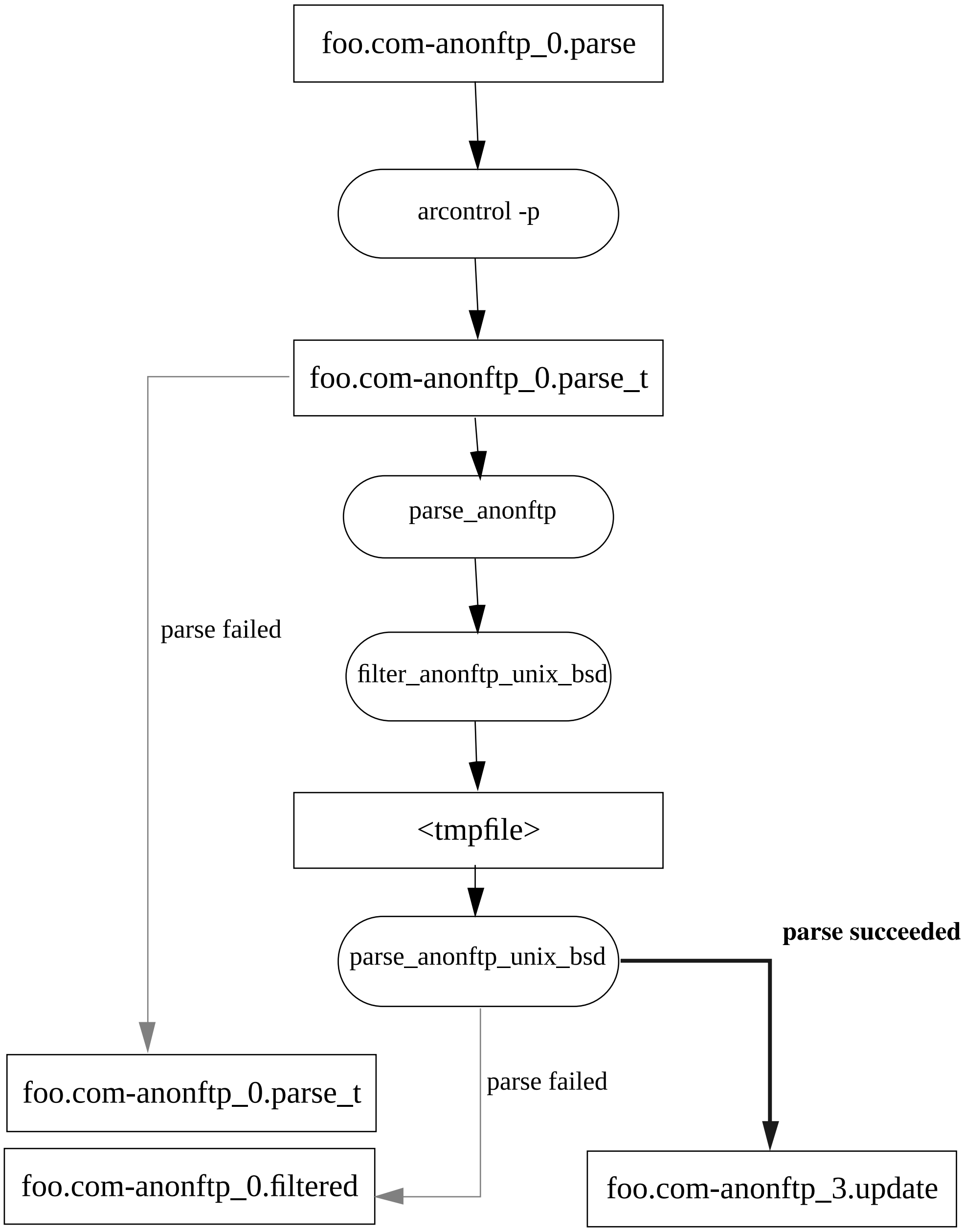


Figure 8.1: The parsing process

8.7 Testing things out

To ensure the system is properly set up, the following programs can be tested by running them from the command line. The results will be written to stdout. Recall that, in normal operation, each of these steps would be run from the cron(8) daemon at predetermined times (see “Configuration” on page 23). Almost all programs in the Archie system will accept a -v (verbose) command line option and you may want to invoke the programs with this flag when testing out the system.

- Load some information into the Host Databases, either through the host_manage.
- Run arretrieve. This will contact the local arserver and request a set of header files, forming the initial data for the Update Cycle.
- Run arcontrol in Data Acquisition mode (the -r command line switch). This will read the header files and connect to the Data Hosts listed in them. It will then perform the required action to obtain the recursive listing from each site.
- Run arcontrol in Parse mode (with the -p command line switch). This will clean up and parse the data into the form required for insertion into the catalog.
- Finally, run arcontrol in Update mode (the -u switch). This inserts the new data into the anonymous FTP (anonftp) catalog and modifies the Host Information Files.
- If you have started the Archie/Prospero server (dirsrv) you can then use any standard Archie client to query the database.
- If you have a WWW server, you can use the cgi-client program to query the database

Chapter 9

Setting up the clients

Before the advent of the World Wide Web, the Archie searches were performed through ascii interfaces. Now we supply a client that can be incorporated to your Web server. This client allows a user to search through the anonftp and webindex databases. The other clients only perform searches in the anonftp database. We will examine the different clients in this chapter.

9.1 Anonftp catalog clients

Currently, the Archie distribution from Bunyip includes two client programs (other than those freely available on the Internet) that enable users to search through Archie anonftp catalog. These are the telnet and email clients. Installation and configuration of these programs are discussed in this section.

In Figure 9.1 below, we show the relationships between the various Archie and standard system programs.

9.1.1 Client choices

The telnet-client enables your user community to connect to an Archie server and query the anonftp catalog through a standard telnet (remote login) session. A user shell is provided and supports a query language to enable the user to query the server. The email-client program enables users to send electronic mail containing a series of commands to the Archie server which will then be processed and returned in email to the user. You will have to decide, as system administrator which of the supplied clients, if any you plan to install. You should factor such things as

- Possible load on the system
- Usefulness of the clients to your community
- Administrative overhead

into your decision. The email and telnet clients can be run "independently" of one another in that you do not have to enable either one for the other one to be made accessible.

9.1.2 The user system programs

The following table describes the programs in the Archie user query system. These programs can be found in the `~archie/bin` directory.

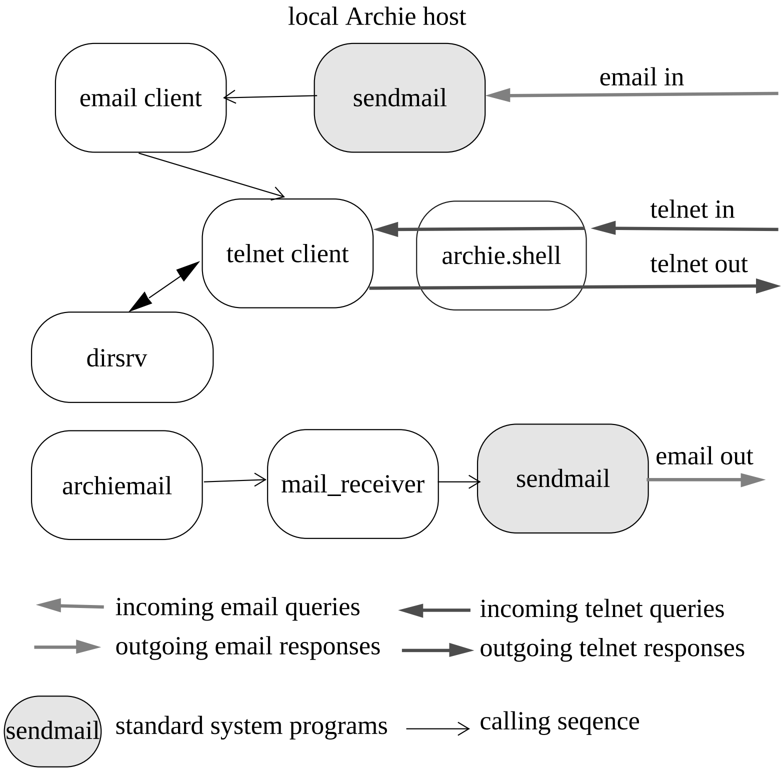


Figure 9.1: Archie system clients

Program Name	Description
telnet-client	User interface via the telnet or rlogin protocols. The only program that performs queries. Called by email-client to process queries.
email-client	The user interface which processes incoming network email queries.
cgi-client	The WWW interface that can be attached to your Web server to perform queries. This will described in more details in section 9.2.1
archie.shell	Bourne shell script wrapper around the telnet-client which restricts the number of simultaneous user telnet Archie sessions. (Optional).
archiemail	"Front end" to the Archie outgoing mail system. Used by the telnet-client "mail" command. Only used if the "mail" command or email-client are going to be enabled.
mail_receiver	Does most of the work in sending mail to the user (generated by the "mail" command or from the email-client): parsing headers, compressing, encoding, splitting and calling sendmail to mail the resulting data
split_file	Used to break large amounts of output into smaller pieces (if necessary), since some mailers are only capable of dealing with messages of limited size.

9.1.3 The telnet client

If you would like your user community to have access to the Archie system via the standard telnet or rlogin programs, you should install the telnet client.

Client privileges and security

The installation phase (as described in "Installing the basic Archie system" on page 15) will, in most cases, install the telnet client to operate as a setuid (root) program. With these privileges, the client can change the current directory to the Archie home directory and call the chroot(2) system call to make this the root of its own directory tree. This has the advantage of reducing any unforeseen security problems when users use rlogin or telnet to connect to the Archie server.

Whether the chroot call is made or not in the telnet client, the GNU less program shipped with the Archie system has been modified to prevent the examination of external files or execution of external processes. The modified version of the sources for this program can be found in `~archie/contrib`.

> The telnet client will relinquish its root status after it has performed this chroot(2) system call, or exit if anything unusual occurs during the steps leading up to (and including) the return from root status.
> The telnet client will function even if it is not given permission to run with root privileges.
> For security reasons, on those platforms in which it is supported Bunyip strongly recommends you install the telnet client to operate with root privileges.

If the telnet-client program is installed suid root and is running on a platform which supports the chroot(2) security feature, copies of the system files `/etc/resolv.conf`, `/etc/services` (unless you are using NIS) and `/etc/termcap` must be placed in the `~archie/etc` directory. This is because the use of the chroot(2) system call will leave the original files inaccessible to the Archie clients.

Enabling the telnet mail feature

Most of the work of setting up the Archie client programs is done by the installation script. However, you must first set up your system to accommodate the new programs. This can be done as follows:

- Create an archiemail entry in the /etc/services file (or the services database, if you are running NIS) and in the `/etc/services` file. For example:

  ```
  archiemail 2709/tcp #The Archie mail server
  ```

- Create a corresponding entry in the /etc/inetd.conf file. For example:

  ```
  archiemail stream tcp nowait archie <archie home>/bin/archiemail
  ```

  Don't forget to restart the inetd program (through the use of the UNIX kill command, as described in "The arserver program"on page 22).

- Create an alias, archie-errors, in the /etc/aliases file. This is where errors generated by incorrect user mail addresses are sent, and is also the Reply-To address in the mail responses sent to the user. To avoid dealing with spurious email, you can set the alias to throw away incoming mail, by setting the /etc/aliases entry to:

  ```
  archie-errors:/dev/null
  ```

  Remember to run, after these changes, the newaliases program or its NIS equivalent (if you are using that system).

- Modify your sendmail configuration file. This is usually called sendmail.cf and resides either in the `/etc` or `/usr/lib` directories. The change that has to be made is to the T configuration lines. The default file will look something like:

  ```
  # Trusted users
  Troot
  Tdaemon
  Tuucp
  ```

  Now add the line:

  ```
  Tarchie
  ```

  This allows the Archie system to send mail with a "Sender:" of archie-errors address. If this is not done, incorrectly addressed mail will bounce from the remote mail receiver back to the Archie user, creating a mail loop which can quickly fill your postmaster's mailbox.

  After making this change you may need to "compile" your sendmail.cf file if that is how your system is configured. Check to see if there is a file called sendmail.fc in the same directory as the sendmail.cf file. If there is, the run the following command:

  ```
  % sendmail -bz
  ```

  This will cause the sendmail configuration to be "compiled" (or "frozen").

- Modify the `/etc/passwd` file for the archie user, with the full path for the telnet-client program as the shell. For example:

```
archie::23:1000:Archie User:<archie home>:<archie home>/bin/telnet-client
```

Note that, in this example, there is no password for the user archie. This is normal for the operation of the most Archie installations. However, while you are testing the system, you may wish to have a temporary password for this account.

Finally, check over the "configure" sections in the archiemail and mail_receiver scripts which reside in `~archie/bin`. You should ensure that the `PATH` variable is correctly set to find the programs used by these scripts.

Limiting the number of telnet sessions

archie.shell is a shell script to limit the number of simultaneous connections to the telnet-client.

- Edit the archie.shell file to reflect the maximum number of simultaneous sessions you want to allow, and the niceness level (see the UNIX manual entry for nice(1)). The lines to be modified are clearly marked in the archie.shell shell script.
- Edit the /etc/passwd file to change the archie user's shell from telnet-client to archie.shell.
- A soft link between the telnet-client in `~archie/bin` and -telnet-client should exist from the default distribution. If it does not, run the following command in that directory.

```
% ln -s telnet-client -telnet-client
```

The .archierc file

On start-up the telnet-client program reads a file called `~archie/.archierc` if it exists. The purpose of this file is to configure the program's behavior to your particular needs. A number of variables can be set and commands executed before the user prompt is generated. See the manual page archie_clients(n) for a complete explanation.

Testing the telnet client

Once the telnet-client program has been installed and configured, test it out by performing a telnet session to the Archie host, and log in as archie. Perform a "find" command to determine if everything is working correctly.

- The telnet client will not start on login:
  - Make sure the permissions on the file `~archie/bin/telnet-client` are correct. It should be executable by all, owned by root and run suid. That is the ls(1) listing should look something like:

    ```
    ---s--x--x 2 root 516096 Jul 16 02:22 telnet-client
    ```

- The error message "Can't resolve hostname (ardp)" is displayed:
  - Check the telnet-client "server" variable (issue the command "show server"). It should be the valid name or IP address of an Archie host or in many cases the name localhost if the telnet client and the dirsrv are running on the same host.
- On start-up or after a command is issued an error is generated which says "Timed out (ardp)":
  - Check the telnet-client "server" variable (issue the command "show server"). It should be the valid name or IP address of an Archie host or in many cases the name localhost if the telnet client and the dirsrv are running on the same host.
  - Make sure that the dirsrv process is running
  - Check that the anonftp database has been initialized
  - If using a remote (not local) Archie server host, it is possible that network delays can cause this problem.

9.1.4 The email client

The email-client program allows users to access the Archie system via standard Internet mail. It is in fact, just a specialized wrapper around the telnet-client program which does the work of processing the incoming queries (see Figure 9.1). The requests from the email client are given the lowest priority when queued on the Prospero server so as not to adversely affect the operation of interactive clients.

The email-client can be used in one to two main ways, either "interactively" in which incoming email is processed as it is received, or "batched" in which the incoming email is stored and processed later, usually at off-peak times. In either case, to enable the email-client, first you need to enable the telnet-client mail feature as described in section "Enabling the telnet mail feature" on page 42.

"Interactive" email requests

For interactive email service add the following line to the /etc/aliases file.

```
archie: ``| <archie home>/bin/email-client''
```

As usual, the newaliases program (or the NIS equivalent) will have to be run in order for this change to take effect.

There are a number of options which can be provided to the email-client binary including some to log incoming messages. See the archie_clients manual page for further details.

If your telnet-client binary is not installed as ~archie/bin/telnet-client then you should use the "-T" option to tell the email client where the telnet-client can be located.

Batching email requests

Email requests need not be processed directly as they are received on your system. Instead, they can be batched and processed at a later (off-peak) time. To do this change the standard entry in the /etc/aliases file to the following:

```
archie ``| <archie home>/bin/batch-email''
```

By default, this will put each piece of mail in a separate file in ~archie/db/tmp. You can override the temporary directory by using the -t option to batch-email which will cause the email messages to be stored in an alternate directory. You should make sure that this is a "safe" directory whose contents will not be lost for example, on a system reboot.

You can then set your system to process the mail at a later time with the process-email program.

For example, if you enter the line:

```
0 1 * * * <archie home>/process-email
```

into the crontab for archuser then each day at 01:00, the mail will be processed. This also takes a -t option like batch-email to provide a different temporary directory. process-email will process each piece of mail in sequence.

Testing the email client

Test the email client by sending mail to archie@<your host name>. If running as "interactive" email, a response should be generated in a number of minutes. Note that this must be done from a machine other than the Archie host due to the fact that on most systems, the sendmail program runs the user daemon when mail is sent from off the local host, but as the user sending the mail when run on the local host.

Email to the Archie server will not work from the Archie server host.

9.1.5 Auxiliary files

The following files in ~archie/etc are used by the telnet-client and email-client programs. They may be modified to reflect the local configuration.

Filename	Used By		Description
	telnet client	email client	
email.help		x	Response to a “help” command or an empty message. For any other form of more specific help (e.g., “help set”), the email-client returns the same information provided by the telnet client.
manpage.ascii	x	x	The pre-formatted version of the Archie manual page.
manpage.roff	x	x	The troff(1) (or nroff(1)) version of the Archie manual page.
motd	x	x	The “message of the day” that the email and telnet clients display to users at the top of email messages or when they log in if set in dirsrv (See “The Message of the Day” on page 52).
serverlist	x	x	The list of servers given to the user in the “servers” command.
whatis	x	x	The Archie whatis database. Entries (one per line) of short descriptions of resources available on the network.

9.1.6 The Help system

The Archie telnet and email client help system is rooted in the directory `~archie/help`. The system currently has full multi-lingual support. In order to support languages other than the default supplied with the distribution, one only need create a directory under `~archie/help` with the name of that language. For example a directory called francais would enable French language support by allowing the users to type:

```
archie> set help francais
```

at the prompt in the telnet client or in the message body to the email client. The files in the `~archie/help/english` need then to be translated into the new language. The files containing the actual data are called ‘=’ and the subdirectory names themselves provide the menu for that part of the help system. See the manual page archie_clients(n) for further details.

9.1.7 Quoting in the clients

For the email and telnet Archie clients, quoting in the command-line interpreter is handled in much the same way as for the standard csh (see the man page csh(1)). So, for example, the ‘.’ can be treated as follows:

```
LIST .il
```

The ‘.’ is treated as a regular expression

```
LIST
.il
```

The ‘.’ is quoted, and passed along (untouched) to the Archie search also if you specify

```
LIST il$
```

you will make sure that the host names returned end with the string “il”. For further details on regular expression consult the regexp(3) manual.

9.2 Webindex catalog clients

This section takes a brief look at the WWW client. It provides access to both webindex and anonftp catalogs. Note that this section is still under construction and any comment is very much welcomed. 3.5

9.2.1 The cgi-client

The files are:

cgi/html/archie.html	Simple search form
cgi/html/archie-adv.html	Advanced search form
cgi/html/archie-help.html	Help page
cgi/bin/archie.cgi	perl script called by the HTML forms
cgi/bin/cgi-client	binary called by the perl script

Where to put them ?

In order to provide a consistent way for accessing the Archie database, we think that the urls should always be of the same format for all public Archie server.

This format being

`http://archie.foo.bar/archie.html`

`http://archie.foo.bar/archie-adv.html`

`http://archie.foo.bar/archie-help.html`

We need to specify the `.html` extension to ensure that all WWW server will actually return the documents as HTML documents. Otherwise, some will presume that they are simply ascii.

Configuration

The configuration procedure is as follows:

1. Move the .html and .gif files to where your web pages are accessed.
2. Move (or link) "archie.cgi" to the cgi-bin directory of your web server.
3. Edit the html files "archie.html" and "archie-adv.html" to reflect the appropriate path of the cgi script "archie.cgi" after you have placed it in the right path.
4. Edit the file "archie.cgi" and configure the first few lines as indicated in the file.
5. Make sure that the UID for the "cgi-client" binary is set.
6. Test the forms and if you don't like the format of the output, take a look at the "archie.cgi" perl script and edit the output variables to your convenience.

How does it work

The "archie.cgi" perl script receives a query (from the archie*.html forms) and calls up "cgi-client". The cgi-client binary processes the request and sends the results to the perl script in plain record-template format. "archie.cgi" reads the results, reformats the output into HTML and then sends the new output to the web server.

How to reformat the output

"archie.cgi" is the perl script you need to modify. The second segment after the configuration part holds the variables that are sent back to the Web server. Configure any variable to your needs.

Where is the information logged

"cgi-client" logs queries, the number of hits and any error messages to a file in the log file `~archie/logs/query.log`

Chapter 10

The Archie/Prospero server (dirsrv)

The basic Archie system is primarily concerned with the location, retrieval and collation of data on the network and does not explicitly define a method to serve this information to the end-user. However, a close relationship between Archie and the Prospero Virtual System exists: all of the most popular freely-available Archie clients as well as those provided by Bunyip use the Prospero protocol to communicate with a modified Prospero server. This server allows users to search and access the various Archie catalogs.

The Archie system includes a Prospero server, modified from the freely-available version, to accommodate many of the new features that Archie offers. The directory `~archie/pfs/bin` contains the complete range of pre-compiled Prospero binaries that are supplied for your convenience.

The Prospero binaries that are supplied have been modified based on the directory structure in the Archie distribution and are rooted in the `~archie/pfs` directory.

If this structure is changed, the system will need to be recompiled. Please check the Prospero configuration files to make sure that they are correct for your installation.

The modified Prospero server retains the full functionality of the basic Prospero system and can be used as such. Please see the Prospero documentation or send mail to info-prospero@prospero.isi.edu for more information.

10.1 Configuring the server

The Archie/Prospero server needs to be configured before it is started.

10.1.1 Configuring dirsrv for the catalogs

Before you start the server, you may need to modify the dirsrv configuration file, ~archie/etc/catalogs.cf. This file tells dirsrv the catalogs to serve to the outside world and where they reside. It has the following format:

`<catalog name> <type> <access method> <location>`

`<catalog name>` The name of the catalog. Now anonftp !

`<type>` The type of catalog, For the moment this is always archie.

`<access method>` The way this catalog is accessed. See text below.

`<location>` Pathname, relative to ~archie/db of where this catalog resides.

For example, here is the default catalogs.cf file that comes with the Archie system distribution.

```
#
#This is a comment
```

```
#database type access method location
# name
anonftp archie anonftp anonftp_db
```

Note that comments can be placed in the file with the use of a “#” character. The comment continues from that character to the end of the current line. Any configuration line may be continued on to the next line with the use of a “
” character at the end of the line to be continued.

10.1.2 Starting the server

You will require root access to perform the following operations.

The following line may be added to the /etc/services file (or to the appropriate NIS file, if it is running on your host). However, the server will operate without this line being present.

```
dirsrv 1525/udp
```

To allow the Prospero server to start whenever the Archie server machine is rebooted, the following command should be added to the end of the /etc/rc.local file on the Archie server host.

```
% <archie home>/pfs/bin/pstart
```

where `<archie home>` is the full path of Archie home directory.

You may also start the Prospero server manually by issuing the pstart command from the command line. If a version of the program is already running, an error message will be displayed and the new version will not start.

The pstart program should be set to have the Archie group as its group. This can be achieved as follows:

```
% chmod g+s pstart
```

This will normally be done by make when initially installing the system.

The pstart process is responsible for starting the program, dirsrv which actually performs the queries. This program must not be renamed, or else it will not be able restart automatically in case of failure.

dirsrv makes extensive use of the files in `~archie/db`. If you plan to do extensive work on these files (such as rebuilding the databases there), make sure that dirsrv is terminated and restarted after the work is complete. Failure to do so may cause the program to fail or operate unpredictably.

The Archie/Prospero server dirsrv in normal operation writes all log messages to the file `/pfs/pfs.log`

It is possible in theory simultaneous heavy update and query loads will cause the dirsrv program to fail. In general, it's best to plan catalog updates for off-peak query hours.

10.2 The dirsrv log files

The dirsrv program writes its logging information to the file `/pfs/pfs.log`. This file contains information about the general operation of the server (startup times, errors encountered etc.) and the user queries processed. Lines containing the string ARCHIE/MATCH signify searches in the anonftp catalog. See the manual sections in the catalog for a detailed description of the syntax of these log entries.

IMPORTANT
Information contained in the dirsrv log file must be considered confidential and should not, under any circumstances, be released without first removing any information which may identify individual users or hosts from which the queries originated.

10.3 The Message of the day

The administrator can set a "Message Of The Day" on the Prospero server via the use of the padmin command found in the `~archie/pfs/bin` directory. This message will be available to all remote Archie (Prospero) clients and is normally displayed by the telnet and email clients for the anonftp catalog supplied with the Archie distribution. You can either prepare a file beforehand containing the message that you want installed or you can type it on the command line. You then issue the command as the superuser (root):

```
% padmin -set MOTD < messagefile
```

If you do not redirect the input the program will prompt for the message on the command line and read it in from stdin. You may wish to put this line in your `/etc/rc.local` file as the MOTD is only set through the padmin command will not be “”remembered” between host reboots or crashes.

- The Archie/Prospero server will not start:
  - Make sure the pstart program has setgid permissions and that it is owned by the archie group.
- dirsrv complains "cannot bind to socket":
  - There is another dirsrv process already running.
- the server will not respond to queries from Archie clients
  - Check the log file for possible error messages explaining the problem.
  - Check to make sure that the `~archie/etc/catalogs.cf` file is correctly configured.
  - Make sure that the permissions on all files in the `~archie/db` directories are correct.
  - Check to see if the server is running under heavy server load. If it is, the query may just be taking a long time to execute. This can be done by monitoring the log file and checking the ongoing use of the system.
  - Make sure that the catalogs that you wish to access have been initialized. The Archie/Prospero server requires that the catalog desired has been initialized before any queries can be processed. See the section on the particular catalog for more details.

10.4 The Log file

The dirsrv process logs client requests to the file `/pfs/pfs.log`. Lines containing the string ARCHIE/MATCH signify searches of the anonftp catalog, while entries with ARCHIE/HOST contain queries related to the telnet-client list command. The syntax of the two entries are described below. Note that the '/' character acts as a separator in the Prospero system.

10.4.1 The MATCH request

A typical anonftp catalog search query is (this is split here across two lines for readability).

```
4-May-94 19:08:03 127.0.0.1(root)[ggB(P53.3Mar94)] - L ARCHIE/MATCH(15,100,100,0,=)/public_access *
```

Field Contents	Description
4-May-94	Date of query
19:08:03	Time of query (local)
127.0.0.1	IP address of host where query originated. Note that for the telnet and email anonftp catalog clients the will ususally be 127.0.0.1 (localhost).
(root)	Name of user sending the query. For the telnet and email clients this will normally be archie. If the email is being run in "interactive" (not batched) mode, this user will be daemon.
[ggB(P53.3Mar94)]	Client ID. All Prospero-based clients are allocated by the Prospero group (info-prospero@prospero.isi.edu) a unqiue identifier. This identifier is changed with each new version of the client.
ARCHIE	Specifies to Prospero that this query is intended for the Archie subsystem.
MATCH	Search of the anonftp catalog
(15,100,100,0,=)	Paramters to the search. The first 3 values set the maxhits, maxmatch, and max_hitspm variables respectively. The 4th value ("0") is the "offset", currently unused. The final charater specifies the type of search to be performed. Possible values and meanings are: = exact R regex r exact_regex X string-only regex x string-only exact_regex C subcase c exact_subcase K string-only subcase k string-only exact_subcase S sub s exact_sub Z string-only sub z string-only exact_sub n string-only exact
"public_access"	String to be searched for

10.4.2 The HOST request

The HOST Prospero request is used to perform both the telnet-client list as well as the "directory expansion" feature found on some Archie clients (notably xarchie). The function performed is determined by the form of the final argument: if it is a regular expression the list function is carried out, otherwise the argument is taken as the host and pathname of a directory to be expanded. HOST commands have the following syntax:

```
2-May-94 01:32:23 132.206.35.4(suzie)[B42E] - L ARCHIE/HOST/sequoia.ccsd.uts.edu.au/pub
```

Field Contents	Description
2-May-94	Date of query
01:32:23	Time of query (local)
132.206.35.4	IP address of host where query originated. Note that for the telnet and email anonftp catalog clients the will ususally be 127.0.0.1 (localhost).
(suzie)	Name of user sending the query.
[B42E]	Client ID.
ARCHIE	Specifies to Prospero that this query is intended for the Archie subsystem.
HOST	list or "directory expansion".
sequoia.ccsd.uts.edu.au	Hostname to be expanded. If this value contains regular expression characters (such as "*" or ']') then the names of all sites matching that expression are returned (the list command).
pub/xntp	Directory pathname to be expanded. That is all files and subdirectories contained within the given directory (if it exsits) will be returned.

Chapter 11

Frequently Asked Questions

This chapter contains a list of Frequently Asked Questions which have been compiled during the operation of Archie over the past several years. It is hoped that it can serve as a quick reference to some of the more common problems associated with running an Archie server.

We hope to regularly add new questions and answers to this chapter and make it available as a stand-alone postscript document in your account on ftp.bunyip.com.

11.1 General

We have DNS. Why does Archie need to store the IP address of a particular host?

> Archie uses IP addresses internally to provide a form of "unique identifier" within the host databases. The Archie system tries to avoid having the same physical host listed in the database under different names (which hosts can have through CNAME records). One can see why when one realizes that if Archie didn't try to do this, then the same host could potentially be indexed several times for each one of its several names. As a result, Archie tries to use the IP address as a kind of "unique identifier" for the host. The system stores the primary host name (the DNS A record) as well as the primary IP address for that host. In most cases, this IP address changes infrequently and serves the purpose. In the case that the host has multiple IP addresses (interfaces) DNS will normally return a "primary" address: this tends to be the first PTR record under DNS. You may notice that in the site files for the anonftp and webindex databases, the IP address is used as the name for the site file. The system is designed to detect when a primary IP address has changed and modify its internal information.
>
> In addition to the internal uses of the IP address, some users, particularly those on the MILNET don't have access to the DNS. As a result, Fully Qualified Domain Names can't be used and the IP addresses are their only route to the archives of the world.

Can I mount the Archie temporary directory (~archie/db/tmp) on another partition?

Can I mount the Archie catalogs (~archie/db/*) on another partition?

> Yes and no... The only restriction at this point is to have `db/tmp` and `etc` in the same partition. If a database, for example `db/anonftp_db`, resides in a different partition, the files will be copied instead of being moved, resulting in slower execution.

The hostname on my Archie server doesn't contain the fully qualified domain name. When I try to collect data from anonymous FTP sites that require a password with a FQDN (like foo@bar.com instead of foo@bar) the process fails. How do I get the system to send a fully qualified domain name?

The primary name of my Archie server host is "foo.bar.com", but I want the Archie system to use the alias "archie.bar.com" (CNAME record) when it contacts remote sites and for the "Source:" line in the host_manage program display. Can this be done?

Yes, if you create a file called ~archie/etc/archie.hostname which just contains the hostname that you would like Archie to call itself then everything will work the way you would like. Note that the name of that file is exactly that "archie.hostname". "hostname" is not a variable to be replaced by some other name.

11.2 Clients

Even with the autologout feature, some sessions on the telnet-client still seem to hang around even though they have been idle for a long time. Why?

The problem usually occurs when users suspend the telnet-client in the process of sending data to the tty (remote connection). The client will try to perform the autologout, but can get suspended trying to close the connection at the other end. We are looking into possible solutions.

How do I tell how many people are using the system?

The question here usually is, "how many queries am I getting?". The easiest way is to look in the file ~archie/pfs/pfs.log and count the lines that have ARCHIE in them (grep -c will do this for you). If you want to get fancy you can separate the lines by date (each line is date and timestamped) to get a count on a per-day basis. You can count the number of people using the telnet client by using the last(1) command ("last Archie" should do it) and you can count the number of queries and users on the email interface by enabling the logging (-u and -l or -L) and counting the lines in there.

How do I restrict the number of concurrent telnet sessions?

Look in the chapter "Setting up the anonftp catalog clients" in the section "Limiting the number of telnet sessions" on page 43. This will explain what you need to know.

Why does mail returned from the Archie email interface say that it comes from "archie-errors"?

This is a historical "accident" with no real meaning. The problem addressed is that if mail came back "From: archie" and it bounced (for whatever reason) then the bounced mail would go right back to the Archie server which would then try to interpret it. In most cases the email interface would generate a help message which would then probably bounce again etc., etc. As a result, mail headers get rewritten so that they bounce to a "bit bucket" and get lost forever. We do not suggest that you point the alias of "archie-errors" at a file or a person.... it or they will probably get a lot of bounced mail that was generated because some user made a typo. The name "archie-errors" can be changed to anything you like however but it will require a slight modification to the system. If you are interested send mail to the Archie contact address here at Bunyip and we'll tell you how to do it.

One of our users is connected to us on a slow (9600 baud) SLIP link and when using an Archie client the whole link becomes unusable. Is there a problem with Archie on slow links?

¡archie home¿ Yes. Archie uses the Prospero system as its transport mechanism. Prospero is a UDP based protocol so when errors occur and packets go missing it is the responsibility of the dirsrv process to perform the retransmissions. On slow links the time allocated for the client to acknowledge receipt of the server packets may expire even though the client has received the packet. As a result the server and client go into a competition to transmit each other packets and this can bring the link to a halt. We know about this and is working on a solution to the problem.

Instead of having the email-client run every time a piece of mail to the Archie address arrives, can I batch the mail and have it processed at off-peak times?

Yes, you can. The program ~archie/bin/batch-email will allow you to do this. See the section "Batching email requests" on page 44 for an explanation.

11.3 The Host Database

How do I change the primary name of a host in the Host Database?

Because the Archie system "keys" on the primary hostname of a host, changing its value requires a deletion of the host from the database followed by a re-insertion with the new name. See "Deleting a Data Host" on page 35.

How do I add a site to the Host Databases so that I can index that site?

Details for this procedure can be found in the section "Adding individual Data Hosts" on page 34 of this manual.

How do I completely delete a site from my Host Database?

You can either use the host_manage program. See the manual pages for further details.

11.4 Parsing

Why does the error line number reported by the parser not correspond to the actual line number that caused the error?

This is a bug in the system. Basically, the number of the line reported is counted from the start of the listing. However, the Archie header adds 8-10 additional lines at the top of the file, causing the mis-count. The actual line causing the error is in fact the number of lines in the header plus the line number reported by the parser. This will be fixed in a patch, soon to be released.

11.5 Data Exchange

I am exchanging data with another Archie server and even though things seem to be working OK (I run arexchange with the -v option and data is being transferred), I don't get any data files in the ~archie/db/tmp directory. What's happening?

The problem results from a combination of things. First, the site with which you are exchanging data has a set of catalog entries which have been deleted. In order to allow this information to be propagated to sites like yourself, the system holds on to these records for a while, until the Archie administrator at the remote site physically deletes the information from his or her Host Database (with the purge_hostdb command). When this has not been done in a long time (or a lot of catalog entries have been deleted recently), these records will be transmitted to you so that your Archie system can delete those entries itself. However, your system is smart enough to recognize that if you didn't have those entries in the first place there is no need to download a "deletion" record for them. Therefore the data that you see being exchanged is really a set of deletion records but since you don't have those entries anyway, your system is ignoring them. The problem comes when you are asking for 50 entries at a time from the remote system and they have more than 50 deleted entries.... you end up with nothing. The solution to the problem is to contact the remote Archie administrator and ask them to run the purge_hostdb command and do some garbage collection.

When I try to do an exchange with a remote Archie server I get a message saying "Network command not terminated with a CR". What is the problem?

When trying to connect to my local arserver program with arretrieve, I get the message, "Read EOF from network connection". What's up?

The arserver process to which you are talking not starting correctly. For example, if arserver sees an option in its command line in `/etc/inetd.conf` which it does not understand, it writes the error message to stdout. Now inetd connects stdout of the processes it runs to the network connection and so you are seeing whatever errors arserver is printing. Note that this problem almost only occurs on start-up of the remote server before it has time to read its options and open its log file, otherwise it would write the errors to its own log file. The solution is to contact the Archie admin of the remote site and ask them to look into the problem. If it is happening to you with the arretrieve program then the same thing applies. Check the information on the command line to arserver in `/etc/inetd.conf` and make sure it is OK. You can also check the Archie log and search for "arserver" and see if wrote any errors there.

11.6 Prospero and dirsrv

If I have made changes to the databases, do I have to restart the dirsrv process?

If you have just made normal, everyday changes to the database then you don't have to restart the dirsrv process. However, if you have added an entire catalog (for example, the anonftp catalog) then you will have to kill and restart dirsrv. For efficiency, dirsrv keeps certain files open all the time (rather than opening and closing them on every query). If you delete the catalog, dirsrv won't know about the new entries and will try to reconcile the files it has open with the new ones, usually killing the process. You can prevent this senseless death by killing and restart the process yourself.

I have noticed the dirsrv process crashing, what's wrong?

The dirsrv process can crash for one of two reasons: (a) the database is corrupt or (b) the database is being updated at the time of the crash. Both are due to "incorrect" pointers in the database the first possibly being permanent the second being transitory. Consistency checking is kept to a minimum in order to speed up the searching process.

db_check attempt to determine if the various elements in the catalog are consistent. Unfortunately, due to the complexity of the catalog structure it is not always obvious how the errors reported by dirsrv are connected to those listed by db_check. Minor inconsistencies given by WARNING messages from db_check are informational in nature and don't affect operations. ERROR messages are more serious. db_check should be run while the catalog is quiescent (not being updated) for a real reading of the current status. If it reports errors, I would advise another update cycle be run and the catalog checked again before worrying about rebuilding it. However, as a rule if pages of errors are reported it is likely that the catalog has been corrupted.

11.7 Catalogs

How do I tell if the one of the catalogs is corrupted?

Well, there are a number of ways. First, check the log file for the word "corrupt". The dirsrv process does some minor checks to make sure everything is okay during its normal operation. (Note that these checks are not very complete as they would slow down the performance of the server and we'd like it as fast as possible). Secondly, if the update cycle fails with programs terminating abnormally, then you could have a problem. For the anonftp catalog (and some others) there is a program called "check_¡catname¿" where ¡catname¿ is the name of a catalog. So, for example, there is a program called "check_anonftp". This program will try to do some comprehensive consistency checking and report what it has found. It may report some messages as "WARNING" which usually means that something is a bit out of order, but nothing serious. If you get "ERROR" messages however, you probably have a corrupt catalog (if the program crashes you definitely have a corrupt catalog).

How do I fix a corrupt anonftp or webindex catalog?

Try running the program fix_start_db if the program db_check says so. If a single site is corrupt, by re-inserting it, the problem should go away.

What causes the anonftp or webindex catalogs to become to become corrupt?

The catalogs can become corrupted in two main ways:

- The host crashes or is rebooted during the update phase of the cycle when the catalogs are being modified. Therefore it is important therefore that the system not be intentionally rebooted when an update is occurring. Of course, host crashes are usually unintentional and can't be prevented.
- The insert_anonftp or insert_webindex program terminates abnormally during an update. This is rare, but can happen as a result of the system core memory being exhausted. The insertion routines are written in such a way that once modifications to the catalog files begins, very little memory is required from the system. However, if the host has no more available memory, even a little will be refused.

In either case, you should perform the routine appropriate for the particular catalog to see if it has been corrupted. If it has the restoration procedure will have to be invoked. In either case the anonftp or (webindex) file will be left around in the ˜archie/db/locks directory to mark this fact. (This will be reported if you have the system set up to mail you status reports).

Why the database get larger and larger over time?

Well, for one thing, there are more and more machines on the network. However, this is not the only reason. The other major one is that due to the way that Archie keeps the strings in the database, "dead" information can collect over time. For example, one anonymous site contains a filename like "furglesnort", it is indexed by the anonftp catalog and then the site removes the file. However, the anonftp catalog will keep the "furglesnort" string in the database (marked inactive) for the rest of time. As a result, such strings build up in the catalog never to be used. While it would be possible to "garbage collect" in the catalog, the structure would require that enough space exist to build a copy of the catalog during the process. The catalog for anonftp is on the order of 500 Mb now so this is not really a viable option. We recommend that the database be rebuilt ever six months or so to remove the dead records.

11.8 Database

Why does the file Stridx.Lock stay around?

The file is used to perform locking through the lock(2) and flock(2) mechanisms for building the index file. The presence of the file does not indicate that the index is locked.

Appendix A

Header record

The following header record fields are currently used by the Archie system:

Field Name	Function
primary_hostname	The primary hostname of the site to which the data belongs. These names are used internally by the Archie system
preferred_hostname	The name under which users see this site listed. It will be a valid Domain Name System canonical name (CNAME) for that site if one has been set
generated_by	The component of the Archie system which has generated this header. Valid values are: `parser:` Output from the parse phase `retrieve:` Output from the data acquisition phase `server:` Generated by the data retrieval phase `admin:` Generated by an administrative procedure `control:` Generated by the controlling routines (usually after an error) `insert:` Generated by normal update routines (seldom seen)
source_archie_hostname	The name of the Archie server responsible for monitoring information at this Data Host
primary_ipaddr	The primary IP address of the Data Host used internally by the Archie system
access_methods	The name of the Archie catalog to which this data belongs. E.g., anonftp (for anonymous ftp listings), webindex (for WWW pages) etc ..
access_command	The catalog-specific sequence of parameters used during the Data Acquisition phase to perform the acquisition of the raw data from the Data Host.
os_type	The operating system type of the Data Host.
timezone	The timezone of the Data Host.
retrieve_time	The time of data acquisition from the data host. This is written as YYYYMMDDHHMMSS (year, month, day, hour, minute, second) and is always in UTC (GMT).
parse_time	The time the data was parsed. Written in the same format as the retrieve_time field.
update_time	The time the data was updated. Written in the same format as the retrieve_time field.
no_recs	The number of "records" in this data. For example, the value for a file listing would be the number of files in the listing. This field is not used by all catalogs.

Field Name	Function
current_status	Lists the current status of the data host. This can be: `active:` available to be queried and inspected `inactive:` temporarily disabled by the system `del_by_archie:` scheduled to be deleted. Usually means that the data in the system is out of date `del_by_admin:` scheduled to be deleted by the local administrator `disabled:` deactivated by the local Archie administrator `not_supported:` catalog type is not supported at this data host
update_status	One of fail or succeed. Used internally by the system to determine result of the previous phase of the update.
prospero_host	One of yes or no describing if the Prospero system is in operation at that site.
data_name	Name of individual data in current file.