

The Netlib Mathematical Software Repository

[Shirley Browne](#)

University of Tennessee
browne@cs.utk.edu

[Jack Dongarra](#)

University of Tennessee and Oak Ridge National Laboratory
dongarra@cs.utk.edu

[Eric Grosse](#)

AT&T Bell Laboratories
ehg@research.att.com

[Tom Rowan](#)

Oak Ridge National Laboratory and University of Tennessee
rowan@msr.epm.ornl.gov

D-Lib Magazine, September 1995

The Netlib repository contains freely available software, documents, and databases of interest to the numerical, scientific computing, and other research communities. The repository is maintained by AT&T Bell Laboratories, by the University of Tennessee and Oak Ridge National Laboratory, and by colleagues world-wide. Many sites around the world mirror the collection and are automatically synchronized to provide reliable and efficient service to the global community through a variety of access mechanisms.

- [Background and History](#)
- [Mirror Sites and Access Methods](#)
- [Contents](#)
- [Future Plans](#)

d-lib forum

d-lib magazine

d-lib next

Background and History

[Netlib](#) began services in 1985 to fill a need for cost-effective, timely distribution of freely available, high-quality mathematical software to the research community [1]. At that time mathematical software was normally distributed as large packages, often on magnetic tape. A researcher may have needed only a single routine to solve a particular problem, but there was no convenient mechanism for distributing small pieces of software or individual routines. In addition, there was no central repository for research software. As a result, valuable software produced from research in numerical analysis was often unavailable to others who might benefit from the work.

In response to these needs, [Jack Dongarra](#) and [Eric Grosse](#) created the Netlib mathematical software repository. They designed and implemented an e-mail service that automatically answered requests for software and information. Users could access the system at any time and receive a prompt reply, usually within a few minutes. The first e-mail servers ran at Argonne National Laboratory near Chicago and at AT&T Bell Laboratories in Murray Hill, New Jersey. The Argonne server was later moved to Oak Ridge National Laboratory in Tennessee.

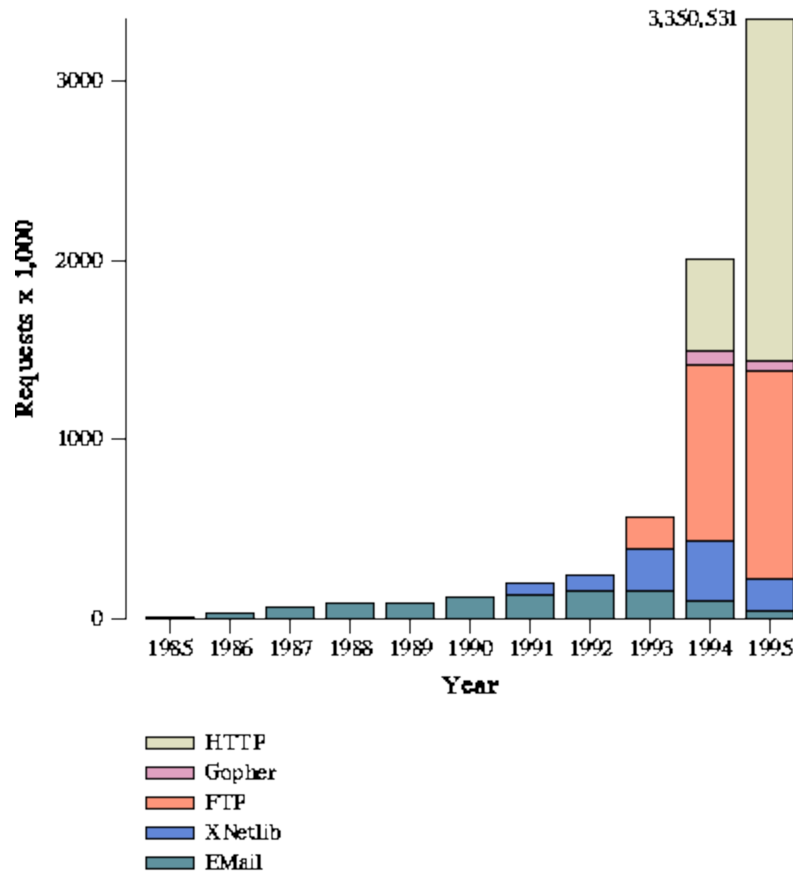
Although the original focus of the Netlib repository was on mathematical software, the collection has grown to include other software (such as networking tools and tools for visualization of multiprocessor performance data), technical reports and papers, a Whitepages Database, benchmark performance data, and information about conferences and meetings. Most of the material in Netlib is freely available to all, but a few packages have restrictions on or require licensing for commercial use.

The number of Netlib servers has also grown from the original two, to servers in Norway, the United Kingdom, Germany, Australia, Japan, and Taiwan. A mirroring mechanism keeps the repository contents at the different sites consistent on a daily basis and automatically picks up new material from distributed editorial sites [3]. Each server is the master for a particular subset of the collection, which is then mirrored by all the other servers.

The original e-mail interface to Netlib is still available, although its use has decreased. In 1991, the Xnetlib [2] X Window interface was made available. Now largely supplanted by repository access via Web browsers, Xnetlib used TCP/IP connections for rapid response and introduced several new mechanisms to facilitate searching through and downloading from the growing collection of software and documents. Anonymous FTP access to Netlib was added in 1991 at Bell Labs and in 1993 at the University of Tennessee. Gopher access was added at the University of Tennessee in 1994. Web access was started at Bell Labs in 1993 and at the University of Tennessee in 1994. The following graph shows the growth in the number of requests using the various access methods at the Tennessee site.

Requests Made to the Netlib Repositories at the Univ. of Tennessee & ORNL

6,736,667 total requests to these repositories as of Sep 15, 1995



Data as of - Sep 15, 1995 at 02:20:12

Netlib differs from many other publicly available software repositories in that the collection is moderated by an editorial board and is widely recognized to be of high quality. However, the Netlib repository is not intended to replace commercial software. Commercial software companies provide value-added services in the form of support. Although the Netlib collection is moderated, its software comes with no support and no guarantees. Netlib's lack of bureaucratic, legal, and financial impediments encourages researchers to submit their codes by ensuring that their work will be made available quickly to a wide audience.

Mirror Sites and Access Methods

The official Netlib mirror sites are the following:

New Jersey [e-mail](#) [WWW](#)

Tennessee	e-mail	WWW
Norway	e-mail	WWW
England	<"mailto:netlib@ukc.ac.uk">e-mail	WWW
Germany	e-mail	WWW
Australia	e-mail	WWW
Taiwan	e-mail	

Although the Web interface to Netlib is more convenient and more popular, the e-mail interface can better illustrate the underlying mechanisms used by Netlib. A Netlib e-mail request consists of lines of one of the following forms:

send index
send index from

library
send routines from *library*
find *keywords*

Some typical examples of requests are the following:

send index

Retrieves summary and description of Netlib repository.

send index from eispack

Retrieves description of the EISPACK library.

send dgeco from linpack

Retrieves routine DGECO and all routines it calls from the LINPACK library.

send only dgeco from linpack

Retrieves just DGECO and not subsidiary routines.

send dgeco but not dgefa from linpack

Retrieves DGECO and subsidiaries, but excludes DGEFA and subsidiaries.

send list of dgeco from linpack

Retrieves just the file names for DGECO and subsidiaries, but not the contents.

find eigenvalue

Retrieves names of all routines and libraries that have the keyword *eigenvalue* in their index file description

The files in Netlib are arranged into topical directories called *libraries*. A library may contain both files and subdirectories. Each library includes a special file named **index**, which describes all the files and subdirectories contained in the library. The index file entries follow a prescribed attribute/value format that is described in the [Netlib Index Format Guidelines](#). As an example, see the [index file for the LAPACK directory](#).

Netlib's default behavior is to send not only the routines explicitly requested, but to send in addition all routines that the requested routines call. By automatically sending subsidiary

routines, Netlib saves the user from making multiple requests and from scanning index files for the names of every routine that might be needed. A user need only request the highest level routine to get everything needed to solve the problem.

The mechanism that enables a routine plus its subsidiary routines to be returned is called *dependency checking*. In general, a routine plus all the routines it calls directly or indirectly form a *dependency tree*. Each library for which dependency checking is turned on contains a configuration file that lists the dependencies for each file in that library. Subsidiary routines may come from the same or from different libraries. Another top-level configuration file specifies for each directory which directories will be searched to resolve dependencies for that directory. Depending on how this top-level file is set up, subsidiary files from a different directory may or may not be retrieved. The top-level configuration file also provides for common misspellings of library names.

The Web interfaces at the different Netlib mirror sites are less uniform than the e-mail interfaces because each site develops its own Web pages and decides independently what services it will offer from its Web server. We will discuss the Tennessee and New Jersey Web interfaces here and encourage the reader to explore the Web pages at the other sites.

[Netlib at the University of Tennessee/Oak Ridge National Lab](#) displays a menu that includes pointers to the Netlib directory tree and a searchable attribute/value index, as well as to the machine performance database, NA-Net numerical analysis on-line community, conferences database, and Netlib FAQ. In addition, a user may access a graphical display of usage statistics that is updated nightly. A user may browse the directory tree to find relevant libraries and individual files or may query the attribute/value database by entering keywords. A list of files and libraries, either a directory listing or the result list from a search, is presented in hypertext format with individual files and libraries highlighted as anchors. Files that have dependencies have two anchors -- one for downloading just the single file and another for downloading the file together with its dependency tree. The dependency checking provided by the Web interface is not as flexible as the e-mail dependency checking mechanism, since it does not include the e-mail options of excluding a subtree or of retrieving dependencies from a user-specified list of directories. We are working on adding these additional capabilities in a way that will still be intuitive and easy to use. Although the Netlib dependency checking mechanism was designed specifically for software, it could probably be generalized to work for documents in HTML comprising a set of hierarchically-organized files.

[Netlib at Bell Labs](#) allows browsing of the Netlib directory tree and searching by keywords. Dependency checking is available to a Web browser from a specially configured [Bell Labs FTP server](#) and will soon be incorporated into the HTML index files on the HTTP server as well. The Bell Labs site also points to specialized catalogs for the approximation and optimization mathematical software areas.

For information about additional access methods, including anonymous FTP, gopher, and CD-ROM, see the [Netlib FAQ](#).

Contents

The Netlib collection currently contains over 30,000 files of software and documents organized into around 200 top-level directories, taking up a total of approximately a gigabyte of disk space. To build the Netlib collection, Dongarra and Grosse collected a variety of freely available high-quality packages and also contacted colleagues who had authored high-quality software. Software packages were included in the main numerical analysis areas of linear systems, eigenvalue problems, quadrature, nonlinear equations, differential equations, and optimization. In addition to the traditional numerical analysis areas, Netlib also contains graphics and computational geometry software and parallel processing tools. The collection changes gradually as new software packages and routines are added and as existing packages are updated. Software contributions go through an editor who determines whether a proposed contribution is within the scope of Netlib and is of sufficient novelty and quality to merit inclusion.

The mathematical software portion of Netlib is classified using the [Guide to Mathematical Software \(GAMS\)](#) system, a tree-structured taxonomy of mathematical and statistical problems. A user may download a copy of the entire GAMS hierarchy or may use a Web browser to traverse it as a decision tree. Netlib is participating in a project to redesign GAMS to use a more flexible faceted classification scheme.

Some of the libraries Netlib distributes -- such as EISPACK, LINPACK, FFTPACK, and LAPACK -- have long been used as important tools in scientific computation and are widely recognized to be of high quality. The Netlib collection also includes a large number of newer, less well-established codes. Most of the software is written in Fortran but programs in other languages, such as C and C++, are also available.

Netlib houses documents as well as software, including the working documents and standards for the High Performance Fortran Forum and Message Passing Interface Forum. Also included in Netlib is the [BibNet bibliographic database](#) maintained by the University of Utah and mirrored by the Netlib sites. In addition to software and documents, Netlib includes a [machine performance database](#) and a [conferences database](#).

Some digital document libraries have a notification service that informs subscribers of newly available documents. The notification service for a software repository is somewhat different, because it informs subscribers of changes and bug fixes to the software as well as additions of new software. In the early days of the Netlib repository, when all access was by e-mail and the traffic was mostly from professional numerical analysts, we relied on log files to send out notification of important bug fixes to everyone who had retrieved affected files. Now, because access is more anonymous and a wider spectrum of users are involved, the old scheme has been replaced by explicit subscription. People may indicate interest in specific Netlib directories using the e-mail *subscribe* and *unsubscribe* commands. For example,

```
subscribe eispack
```

Automatic notification is sent, on a daily basis, when files in the directory are changed. The subscriber lists also give the authors and editors a way to judge what community is particularly interested in a given Netlib collection.

Future Plans

Netlib is moving in several directions that will enhance and facilitate use of the collection and that will transfer lessons learned from its development to other repository development efforts. These directions include location independence for named files, expert system capability for assisting naive users, flexible distributed processing for information-intensive or composite applications, and packaging of repository technology.

Currently a user must know the URL of a Netlib mirror site to access it. Ideally, a user should be able to take a location-independent name for a Netlib file and resolve it to a copy at the nearest mirror site. If this site were inaccessible, alternative copies could be tried. Such a capability would allow a single name to be returned by a search service or to be cited as a reference, while giving users the reliability and performance benefits of replication. We are working on replication and name resolution scheme for network resources, and we hope to incorporate this scheme into Netlib in the future.

The current organization and classification schemes used in Netlib may still leave some users with too wide a choice of routines for solving particular problem. In addition, inexperienced users may be unsure what classes or libraries are most relevant to search. For these reasons, we are developing expert help systems for particular sub-domains that will assist users in mapping problem features to appropriate problem-solving modules.

Netlib is typical of most repositories in that software and documents are located by browsing and searching and are then transferred over the network to the user's machine where they are read or used. The usual method of using software located in a repository is to download the software source or binary files to one's local machine, and then to compile and execute the software using locally stored input data. This method does not work well for information processing applications for which large amounts of data distributed at different locations need to be processed, summarized, and assimilated. For some information processing applications, the data may reside in remote repositories, and the user may wish to write customized programs for processing the data. It will often be more efficient to transfer the program to the data and execute it remotely. Furthermore, for composite problems, the information processing and computational activities may need to be carried out at widely distributed sites by cooperating agent programs, or via downloadable client interfaces that allow users to interact with remote computational servers. To provide such capabilities, we are exploring the use of safe execution environments for agent programs and browser helper modules written in interpreted languages such as Java, Tcl/Tk, and Python.

A separate but related effort to Netlib is the [National HPCC Software Exchange \(NHSE\)](#). The purpose of the NHSE is to capture, preserve, and make available all software and information produced by the federal [High Performance Computing and Communications \(HPCC\) program](#) and related efforts. The NHSE promotes the development of discipline-oriented software and document repositories, and of contributions to and use of such repositories by members of the high performance computing community. The NHSE will provide a uniform interface to a virtual HPCC software repository that will be built on top of the distributed set of discipline-oriented repositories. Lessons learned from years of maintaining Netlib are helping NHSE developers

design and package appropriate tools for building and maintaining domain-specific HPCC repositories.

References

- 1 Jack Dongarra and Eric Grosse. [Distribution of Mathematical Software via Electronic Mail](#). *Communications of the ACM* 30(5):403-407, 1987.
- 2 Jack Dongarra and Thomas Rowan and Reed Wade. [Software Distribution Using XNETLIB](#). *Transactions on Mathematical Software* 21(1), March 1995.
- 3 Eric Grosse. [Repository Mirroring](#). *Transactions on Mathematical Software* 21(1), March 1995.

Copyright © 1995 Shirley Browne, Jack Dongarra, Eric Grosse, and Tom Rowan



hdl://cnri.dlib/september95-browne