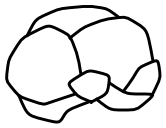


Contact

Research Computing
The Ohio State University
1224 Kinnear Road
Columbus, OH 43212

Consulting

Phone: (614) 292 8492
Internet: trollius@tbag.osc.edu

Copyright

This document is protected by copyright.
Author: GDB

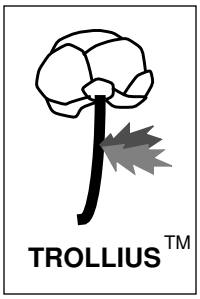
© Copyright 1992 The Ohio State University

Trademarks

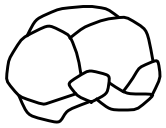
Trollius is a trademark of The Ohio State University and the Cornell Research Foundation.

UNIX is a trademark of AT&T.

X Window System is a trademark of the Massachusetts Institute of Technology.



Trollius Installation Guide



file.) Users choose among different installations by setting their shell's search path accordingly.

New Releases

Assuming you have ported a release of Trollius to a custom hardware system, how do you handle a new release of the software? Carefully read the *Ohio Trollius Release Notes* and consult the **misc/Fixes** file for information on changes from the previous release. When Trollius changes have been made in a particular file, you will either have to reintegrate your local changes or the Trollius changes, whichever is less work. If a file has only experienced local changes, you can substitute it wholly into the new release. Local changes pertaining to a custom hardware system should be isolated in the machine dependent subdirectory discussed earlier. At all times, the UNIX **diff** tool is your best friend.

Before installing a new release it is a good idea to save the old installation directory as a backup.

Saving Space

After Trollius has been built, all of the objects can be removed with the following command, invoked at the top of the source directory:

```
% make clean
```

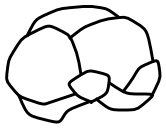
If further space is required, the source directories can be taken off-line. Only the installation directory need be maintained on-line. You will want to save the **man** subdirectory, perhaps by copying it into the installation directory and setting the **MANPATH** environment variable accordingly. To save space during the installation procedure, load, install, clean and remove one source directory at a time. This will greatly decrease the maximum space requirement.

Optimization

Trollius is not optimized for any particular version of UNIX. There is, however, one custom module designed for UNIX System V that speeds up host inter-process communication (and hence, file system access) by a factor of four to six. To install this optimization, change the **OKIO** variable in the Trollius **Config/config** file to **msg** before installation (or repeat the two installation steps).

Binary Release

Installation of a binary release is simple. Place the supplied installation directory at any location. If a LSTT binary release is used, both the **LSTTHOME** and **TROLLIUSHOME** environment variables must be set to the LSTT and Trollius installation directories, respectively. If a Trollius binary release is used, the **TROLLIUSHOME** variable must be set.



Advice and Experience

Due to the non-standard nature of UNIX, vendor and local deviations can cause unforeseen problems for the makefiles and sources. We almost certainly did not cover all of the possibilities. If you've never installed software on a UNIX machine, you might want to find someone who has.

When Things Go Wrong

It is not beyond the realm of possibility that the depend or build (install) step will fail. The reason can usually be tracked down to permissions or a unplanned incompatibility in a UNIX header file. Some versions of make will not stop if a step fails. This is why we recommend keeping the output of make in a log file.

A compilation can fail when a UNIX defined structure differs between the Trollius source and the local flavour of UNIX. A second problem is differing macros in UNIX header files. The Trollius source has been successfully compiled on SunOS 3.2 thru 4.1.1 and many other UNIX products. The only way around these problems is to take a good look around, roll up your sleeves and eliminate the problem by either dodging it, or making liberal use of #ifdef.

Partial Makes

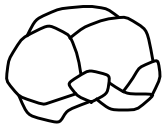
Whenever a bug is repaired or an enhancement is added, a complete rebuild, as described in the previous section, is sure to recompile and reinstall the proper modules. The depend step is necessary only if new files have been added, old ones have been removed (in both cases the **`SRC`** and **`OBJ`** variables in the appropriate Makefile must be altered) or if a change has occurred in the inclusion of header files. A complete rebuild can soon become tedious and a partial rebuild can be done with care.

The first rule of partial makes is **never make in a source subdirectory**.

Rebuilding a program is a simple matter of finding the right object subdirectory and invoking **`make`** one time. Rebuilding a library is more difficult. If a module affecting a library has been changed, the appropriate library and all programs which link the library must be rebuilt. Also, many library modules are ITB/OTB portable and both versions of the libraries must be rebuilt. We recommend a complete top level make after any library or header file (**`share/lib/h`**) change.

Multiple Installations

A convenient feature is the ability to support multiple installations for experimentation. To build another installation directory, simply repeat the building steps with a different value of the **`HOME`** variable. (A command line setting of this variable is not sufficient - you must edit the **`config`**



Building Installation Directories

Locate to the top of a source directory. The build procedure is a two step process. First the Makefiles establish dependencies on header and source files.

```
% make depend >& DEPLOG &
```

You can monitor the progress with the tail utility:

```
% tail -f DEPLOG
```

Analyse the log file (**DEPLOG**) to ensure that the Makefiles could find all the required source and header files. In the second step, the executables and libraries are compiled, linked and archived as necessary:

```
% make install >& MKLOG &
```

Again check the log file for any compilation errors. The installation directory is now built. These steps are repeated for each source directory.

Tkill Anarchy

The Trollius tkill tool can be given setuid root capability so that one user may terminate another user's Trollius session due to extraordinary circumstances. This is a powerful and not fully secure capability. If desired, perform the following steps in the Trollius source directory:

```
% su
% make kill
% exit
```

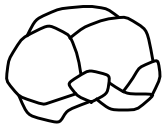
Shell and Environment Variables

Trollius users will have to add the host executable directories to their shell's search path. These directories are defined by **OBDIR** in the **config** file of each source directory. As previously noted, all source directories may use the same installation directory and hence, have the same value for **OBDIR**. The installation directories can be moved after they are built and a shell's search paths will have to change accordingly. If the Trollius installation directory is moved, users must define the variable **TROLLIUSHOME** in their environment to be the new directory pathname. If the LSTT installation directory is moved, the **LSTTHOME** variable must be set.

Trollius users should add the man subdirectory of each source directory to their **MANPATH** environment variable.

Documentation

User documents are found in the **doc** subdirectory in PostScript format. They contain the following fonts: Times 12, Helvetica 12, Helvetica 24, Courier 12, Palatino 24 and Palatino 10



Local Variations

Each source directory has a file, **`Config/config`**, that contains configuration information for building and installing the objects and executables. The file contains many variables that are rarely changed or not intended to be changed by the installer. A few variables must be set by the installer.

`HOME =` <installation directory>

Objects and executables are built in the source directory (actually in object subdirectories of the source directory) and installed in the installation directory. The installation directory, which can be located anywhere, is configured by the **`HOME`** variable. Separate source directories may choose separate installation directories or they may choose the same directory.

A bonus feature of LSTT is that it will search the Trollius installation directory for libraries and header files. The appropriate Trollius installation subdirectories must be configured into the LSTT source directory.

`ITDIR =` <Trollius **`IDIR`**>
`ILTDIR =` <Trollius **`ILDIR`**>

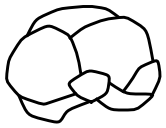
Machine Dependent Subdirectory

Software that is specific to a particular hardware system is located under the **`machine`** subdirectory. Trollius supports different hardware systems and they each have a subdirectory under **`machine`**. If new support for custom hardware has been implemented, it will appear in a new machine dependent subdirectory. The installer must create, in the source directory, a symbolic link called **`port`** to the appropriate machine dependent subdirectory.

`% ln -s machine/`<hardware system> **`port`**

Boot Schema

The installer should create boot schemata for typical configurations of the local multicomputer(s). This relieves most users of the tedious task of preparing a boot schema. Create one boot schema for each common configuration and place them in the **`schema`** subdirectory of the Trollius source directory. Remove any existing schemata that do not apply at the local site. Consult the *Trollius User's Guide* for detailed information on how to write a boot schema.



General Notes

Source Directories

The Trollius Project develops, maintains and delivers software in three separate source directories:

LSTT	the Logical Systems Transputer Toolset
Trollius	the essential operating system
Buckeye	the supplementary operating system

The source directories must be installed in the order listed above. Installation of the Buckeye source directory is optional. LSTT installation is required if Trollius is to be installed for transputer based multicomputers.

Space Requirements

Once all source directories are loaded on disk and built into objects and executables, the total space requirement will approach 30 megabytes. Strategies to use less space are described in *Saving Space*.

Release Format

The Trollius Project delivers software via ftp and occasionally via cartridge tape or floppy diskette. Regardless of the medium, the installer will always receive one or more files of the form:

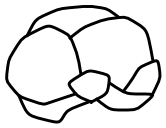
```
x.tar.Z
```

The x is replaced by the name of a particular source directory. This is a compressed tape archive and it can be located anywhere. The UNIX procedure for unwrapping the source directory is:

```
% uncompress x.tar.Z
% tar xvf x.tar
```

A patch file may also be supplied that contains a list of serious bug repairs since the last general Trollius release. After unwrapping the file as described above, the installer must read each patch carefully, decide if the patch is applicable to the local site and apply the patch accordingly, where directed.

```
patch.tar.Z
```



How to Use This Document

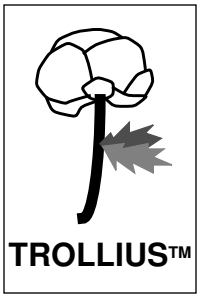
This document covers the installation of Ohio Trollius and related software products. It is intended for system administrators and assumes an understanding of the UNIX system as well as the **`make`** tool. A complete read of this document is recommended prior to undertaking an installation.

Two important abbreviations are used through the Trollius literature:

ITB	"In The Box" refers to software which runs on dedicated computing nodes where Trollius is the native operating system.
OTB	"Out of The Box" refers to software which runs on UNIX computers where Trollius is a UNIX application.

The font and syntax conventions have been kept to a minimum.

`code`	This font is used for things you type on the keyboard or see printed on the screen.
<symbol>	This is a symbol used to abstract something you would type.
Section	Italics are used to cross reference another section in the document or the title of another document.



Trollius Installation Guide

Trollius is a programming environment for distributed memory MIMD parallel computers, aka multicomputers. This document describes how to install and build Trollius from a source level release. As with all operating systems, the hardware interface is a crucial issue in installation. For information on how to develop a custom hardware interface for Trollius, consult the *Trollius Porting Guide*. For information on the exact contents of a given release, consult the *Ohio Trollius Release Notes*. Other documents are available that describe how to use and program Trollius.

Ohio Supercomputer Center

Research Computing
The Ohio State University

Document Series 2/6 **March 3, 1992**