|  |  |
| --- | --- |
| **1.** | **Installing TINKER on your Computer** |

The TINKER package is distributed on the Internet via either the web site or the anonymous ftp account on **dasher.wustl.edu** with an IP number of 128.252.208.48. This node is a web and file server located in the Ponder lab at Washington University School of Medicine. The package is available via the web and standard browsers from the TINKER home page at **http://dasher.wustl.edu/tinker/**. Alternatively TINKER can be downloaded by logging into **dasher.wustl.edu** via anonymous ftp (Username: **anonymous**, Password: **"your email address"**) and downloading the software from the /pub/tinker subdirectory. The complete TINKER distributions as well as individual files can be downloaded from this site.

The easiest way to get TINKER running on your machine is to use the self-extracting installation kit for either Linux, Windows, or Macintosh OS X 10.3. The installer will guide you through complete setup of TINKER and the Force Field Explorer (FFE) GUI, and perform all required configuration chores. The installer kits for the three supported systems are **tinker4.2-linux.sh**, **tinker4.2-windows.exe** and **tinker4.2-macosx.sit**. The Linux and Windows kits each contain a private copy of a Java and Java3D run-time environment for use with the package. The Macintosh version requires an OS X 10.3 (Panther) system for installation. The native Java implementation is used on Macs, and the Java3D package must be downloaded from Apple and installed prior to using TINKER with Force Field Explorer.

The TINKER package is also available as compressed Unix tar archives, Windows zip files, and as a complete set of uncompressed source and data files. Binaries are provided for machines running Windows 9X/ME/NT/2000/XP, Linux, and Apple Mac OS X. All of these executables are present in standard compressed formats as individual programs or as complete sets of executables. It is expected that other Unix users and PC users who need specially customized versions, will build binaries for their specific system. Sites with access to the Unix tar, compress and uncompress commands should simply obtain the archive file **tinker.tar.Z**. Alternatively, **tinker.tar.gz** and **tinker.zip** containing identical distributions compressed to GNU gzip and Windows ZIP format are also provided. If you choose to download individual files, you will need at a minimum the contents of the /doc, /source and /params subdirectories. Also required are the compile/build scripts from the subdirectory named for your machine type. Other areas contain test cases and examples, benchmark results, machine-translated C code, and the Force Field Explorer Java GUI for TINKER. The entire TINKER package, after building the executables, will require from about 40 to over 150 megabytes of disk space depending on the components installed and the use of shared libraries in the executables.

The documentation for the TINKER programs, including the guide you are currently reading, is located in the /pub/tinker/doc subdirectory. The documentation was prepared using the Applixware Words and Graphics programs. Portable versions of the documentation are provided as ascii text in **.txt** files and in Adobe Acrobat **.pdf** file formats. Please read and return by mail the TINKER license. In particular, we note that TINKER is not ``Open Source'' as users are prohibited from redistribution of original or modified TINKER source code or binaries to other parties. While our intent is to distribute the TINKER code to anyone who wants it, the Ponder Lab would like to remain the sole distribution site and keep track of researchers using the package. The returned license forms also help us justify further development of TINKER. When new modules and capabilities become available, and when the almost inevitable bugs are uncovered, we will attempt to notify those who have returned a license form. Finally, we remind you that this software is copyrighted, and ask that it not be redistributed in any form.

The compilation and building of the TINKER executables should be easy for most of the common workstation and PC class computers. We provide in the /make area a Unix-style Makefile that with some modification can be used to build TINKER on most Unix machines. As a simpler alternative to Makefiles for the Unix versions, we also provide machine-specific directories with three separate shell scripts to compile the source, build an object library, and link binary executables. Three similar command files are provided for Windows, Macintosh and Open VMS systems. Compilation on Unix workstations should use the vendor supplied Fortran compiler, if available. The public domain GNU g77 Fortran compiler available from **http://gcc.gnu.org/** is also capable of building TINKER on Linux and other Unix-based machines. The Linux executables we provide are built with the Intel Fortran for Linux 8.0 compiler. The Portland Group (PGI) and Absoft ProFortran compilers have also been tested under Linux, both of which generate executables roughly comparable in speed to the Intel compiler. On Linux, the g77 executables tend to exhibit degraded performance compared with executables from commercial compilers. Some benchmark results are provided in a later section of this User's Guide For the Macintosh we distribute executables built under Apple OS X 10.3 with the GNU g77 compiler. TINKER also builds on the Macintosh using the Absoft ProFortran compiler. For PCs running Windows 9X/NT/2000/XP, the distributed TINKER executables are built under the Intel Fortran for Windows 8.0 compiler. Alternative Windows compilers such as Compaq Visual Fortran, Lahey/Fujitsu and The Portland Group compilers, and GNU g77 under Cygwin have been tested and shown to build TINKER correctly. Please see the README files in each of the machine-specific areas for further information.

The first step in building TINKER using the script files is to run the appropriate **compile.make** script for your operating system and compiler version. Next you must use a **library.make** script to create an archive of object code modules. Finally, run a **link.make** script to produce the complete set of TINKER executables. The executables can be renamed and moved to wherever you like by editing and running the ``rename'' script. These steps will produce executables that can run from the command line, but without the capability to interact with the FFE GUI. Building FFE-enabled TINKER executables involves replacing the **sockets.f** source file with sockets.c, and included the object from the C code in the TINKER object library. Then executables must be linked against Java libraries in addition to the usual resources. Sample **compgui.make** and **linkgui.make** scripts are provided for systems capable of building GUI-enabled executables.

Regardless of your target machine, only a few small pieces of code can possibly require attention prior to building. The first two are the system dependent time and date routines found in **clock.f** and **calendar.f** respectively. Next is the **openend.f** routine that facilitates appending data to the end of an existing disk file. Please uncomment the sections of these routines needed for your computer type. Version of these system dependent routines suitable for each system are also provided in the directory for each machine/OS type. The final set of possible source alterations are to the master array dimensions found in the include file **sizes.i**. The most basic limit is on the number of atoms allowed, ``maxatm''. This parameter can be set to 10000 or more on most workstations. Personal computers with minimal memory may need a lower limit, perhaps 1000 atoms, depending on available memory, swap space and other resources. A description of the other parameter values is contained in the header of the file. Note that in order to keep the code completely transparent, TINKER does not implement any sort of dynamic memory allocation or heap data structure. This requires that **sizes.i** dimensioning values be set at least as large as the biggest problem you intend to run. Obviously, you should not set the array sizes to unnecessarily large values, since this can tax your compute resources and may result in performance degradation or overt failure of the executables.

Specific questions about the building or use of the TINKER package should be directed to **tinker@dasher.wustl.edu**. TINKER related questions or comments of more general interest can be sent to the Computational Chemistry List (**http://www.ccl.net/**) run by Jan Labanowski at the University of Notre Dame. The TINKER developers monitor this list and will respond to the list or the individual poster as appropriate.