

Installation Manual for the PSI3 Program Package

T. Daniel Crawford,^a C. David Sherrill,^b and Edward F. Valeev^b

^a*Department of Chemistry, Virginia Tech, Blacksburg, Virginia 24061-0001*

^b*Center for Computational Molecular Science and Technology,
Georgia Institute of Technology, Atlanta, Georgia 30332-0400*

PSI3 Version: 3.2.1 (stable)

Created on: April 23, 2004

1 Compilation Prerequisites

The following external software packages are needed to compile PSI3:

- A well-optimized basic linear algebra subroutine (BLAS) library for vital matrix-matrix and matrix-vector multiplication routines. We recommend the excellent ATLAS package developed at the University of Tennessee. math-atlas.sourceforge.net
- The linear algebra package (LAPACK), also available from netlib.org. PSI3 makes use of LAPACK's eigenvalue/eigenvector and matrix inversion routines. www.netlib.org/netlib
- POSIX threads (Pthreads) library
- Perl interpreter (version 5.005 or higher)
- Various GNU utilities: www.gnu.org
 - `autoconf` (version 2.52 or higher)
 - `make`
 - `flex`
 - `bison`
 - `fileutils` (esp. `install`)
- For documentation:
 - LaTeX
 - LaTeX2html (v0.99.1 or 1.62, including the patch supplied in `psi3/misc`)

2 Basic Configuration and Installation

A good directory for the PSI3 source code is `/usr/local/src/psi3`. The directory should *not* be named `/usr/local/psi`, as that is the default installation directory unless changed by the `--prefix` directive (see below). It should also not have any periods in the path, e.g., `/usr/local/psi3.2`, because of a bug in `dvips` which will cause the compilation of documentation to fail.

The following series of steps will configure and build the PSI3 package and install the executables in `/usr/local/psi/bin`:

1. `cd $PSI3` (your top-level PSI3 source directory)
2. `autoconf`
3. `mkdir objdir`
4. `cd objdir`

5. `../configure`
6. `make`
7. `make tests` (optional, but recommended)
8. `make install`
9. `make doc` (optional)

There is also a perl script, `INSTALL.pl`, in the top-level `$PSI3` source directory which provides an interactive interface for installation.

You may need to make use of one or more of the following options to the `configure` script:

- `--prefix=directory` — Use this option if you wish to install the PSI3 package somewhere other than the default directory, `/usr/local/psi`.
- `--with-cc=compiler` — Use this option to specify a C compiler. One should use compilers that generate reentrant code, if possible. The default search order for compilers is: `gcc`, `cc`. (NB: On AIX systems, the search order is `cc_r`, `gcc`.)
- `--with-cxx=compiler` — Use this option to specify a C++ compiler. One should use compilers that generate reentrant code, if possible. The default search order for compilers is: `g++`, `c++`, `cxx`. (NB: On AIX systems, the search order is `xlC_r`, `c++`, `g++`.)
- `--with-fc=compiler` — Use this option to specify a Fortran-77 compiler. One should use compilers that generate reentrant code, if possible. The default search order for compilers is: `g77`, `f77`, `fc`, `f2c`. (NB: On AIX systems, the search order is `xlF_r`, `g77`, `f77`, `fc`, `f2c`.)
- `--with-ld=linker` — Use this option to specify a linker program. The default is `ld`.
- `--with-ranlib=ranlib` — Use this option to specify a `ranlib` program. The default behavior is to detect an appropriate choice automatically.
- `--with-ar=archiver` — Use this option to specify an archiver. The default is to look for `ar` automatically.
- `--with-ar-flags=options` — Use this option to specify archiver command-line flags. The default is `r`.
- `--with-perl=perl` — Use this option to specify a Perl interpreter. The default is to look for `perl` automatically.
- `--with-incdirs=directories` — Use this option to specify extra directories where to look for header files. Directories should be specified prepended by `-I`, i.e. `-Idir1 -Idir2`, etc. If several directories are specified, enclose the list with single right-quotes, e.g., `--with-incdirs='-I/usr/local/include -I/home/psi3/include'`.

- **--with-libs=libraries** — Use this option to specify extra libraries which should be used during linking. Libraries should be specified by their full names or in the usual `-l` notation, i.e. `-lm /usr/lib/libm.a`, etc. If several libraries are specified, enclose the list with single right-quotes, e.g., `--with-libs='-lcompat /usr/local/lib/libm.a'`.
- **--with-libdirs=directories** — Use this option to specify extra directories where to look for libraries. Directories should be specified prepended by `-L`, i.e. `-Ldir1 -Ldir2`, etc. If several directories are specified, enclose the list with single right-quotes, e.g., `--with-libdirs='-L/usr/local/lib -L/home/psi3/lib'`.
- **--with-blas=library** — Use this option to specify a BLAS library. If your BLAS library has multiple components, enclose the file list with single right-quotes, e.g., `--with-blas='-lf77blas -latlas'`.
- **--with-lapack=library** — Use this option to specify a LAPACK library. If your LAPACK library has multiple components, enclose the file list with single right-quotes, e.g., `--with-lapack='-llapack -lcblas -latlas'`.
- **--with-max-am-eri=integer** — Specifies the maximum angular momentum level for the primitive Gaussian basis functions when computing electron repulsion integrals. This is set to g -type functions (AM=4) by default.
- **--with-max-am-deriv1=integer** — Specifies the maximum angular momentum level for first derivatives of the primitive Gaussian basis functions. This is set to f -type functions (AM=3) by default.
- **--with-max-am-deriv2=integer** — Specifies the maximum angular momentum level for second derivatives of the primitive Gaussian basis functions. This is set to d -type functions (AM=2) by default.
- **--with-max-am-r12=integer** — Specifies the maximum angular momentum level for primitive Gaussian basis functions used in r_{12} explicitly correlated methods. This is set to f -type functions (AM=3) by default.
- **--with-debug=option** — This option turns on debugging options. If the argument is omitted, “-g” will be used by default.
- **--with-opt=options** — This option may be used to select special optimization flags, overriding defaults.

3 Detailed Installation Instructions

This section provides detailed instructions for compiling and installing the PSI3 package.

3.1 Step 1: Configuration

First, we recommend that you choose for the top-level `$PSI3` source directory something other than `/usr/local/psi`; your `$HOME` directory or `/usr/local/src/psi3` are convenient choices. Next, in the top-level `$PSI3` source directory you've chosen, first run `autoconf` to generate the configure script from `configure.in`. It is best to keep the source code separate from the compilation area, so you must choose a subdirectory for compilation of the codes. A simple option is `$PSI3/objdir`, which should work for most environments. However, if you need executables for several architectures, choose more meaningful subdirectory names.

- The compilation directory will be referred to as `$objdir` for the remainder of these instructions.

In `$objdir`, run the configure script found in the `$PSI3` top-level source directory. This script will scan your system to locate certain libraries, header files, etc. needed for complete compilation. The script accepts a number of options, all of which are listed above. The most important of these is the `--prefix` option, which selects the installation directory for the executables, the libraries, header files, basis set data, and other administrative files. The default `--prefix` is `/usr/local/psi`.

- The configure script's `--prefix` directory will be referred to as `$prefix` for the remainder of these instructions.

3.2 Step 2: Compilation

Running `make` (which must be GNU's 'make' utility) in `$objdir` will compile the PSI3 libraries and executable modules.

3.3 Step 3: Testing

To automatically execute the ever-growing number of test cases after compilation, simply execute "make tests" in the `$objdir` directory. This will run each (relatively small) test case and report the results. Failure of any of the test cases should be reported to the developers at `psi3@psicode.org`. By default, any such failure will stop the testing process. If you desire to run the entire testing suit without interruption, execute "make tests TESTFLAGS='-u -q'". Note that you must do a "make testsclean" in `$objdir` to run the test suite again.

Testing PSI3 from the source directory, which was possible in prerelease version of PSI3 (`rc1` and `rc2`), is no longer recommended.

3.4 Step 4: Installation

Once testing is complete, installation into `$prefix` is accomplished by running `make install` in `$objdir`. Executable modules are installed in `$prefix/bin`, libraries in `$prefix/lib` and basis set data and other control structures `$prefix/share`.

3.5 Step 5: Documentation

If your system has the appropriate utilities, you may build the package documentation from the top-level `$objdir` by running `make doc`. The resulting files will appear in the `$prefix/doc` area.

3.6 Step 6: Cleaning

All compilation-area object files and libraries can be removed to save disk space by running `make clean` in `$objdir`.

3.7 Step 7: User Configuration

After the PSI3 package has been successfully installed, the user will need to add the installation directory into their path. If the package has been installed in the default location `/usr/local/psi3`, then in C shell, the user should add something like the following to their `.cshrc` file:

```
setenv PSI /usr/local/psi3
set path = ($path $PSI/bin)
setenv MANPATH $PSI/doc/man:$MANPATH
```

The final line will enable the use of the PSI3 man pages.

4 Miscellaneous architecture-specific notes

- AIX 4.3/5.x in 64-bit environment: if IBM VisualAge C++ and IBM XL Fortran are used, one has to manually specify the `-q64` compiler flag that enables production of 64-bit executables. The following configure options should be used: `--with-cc='cc_r -q64 -DFCLINK=2' --with-cxx='xlc_r -q64 -DFCLINK=2' --with-fc='xlf_r -q64'`. Note that the reentrant versions of the compilers are used. Also, we haven't had much luck using `xlc_r` because of its handling of functions with variable argument lists, use `cc_r` instead.
- Compaq Alpha/OSF 5.1: default shell (`/bin/sh`) is not POSIX-compliant which causes some PSI3 makefiles to fail. Set environmental variable `BIN_SH` to `xpg4`.
- Mac OS 10.x:
 - The compilation requires a developer's toolkit from apple.com.

- You need the `libcompat` library. It can be obtained from Apple's website at <http://www.opensource.apple.com/>. Then add `-lcompat` to the configure flag `--with-libs`.
- If you are using compilers from the developer's kit then for BLAS and LAPACK, use the configure options:

```
--with-blas='-altivec -framework vecLib'
```

If you compiled compilers yourself from GNU source code then Apple-specific extensions will not work and you will have to specify the location of `vecLib` manually:

```
--with-blas='/System/Library/Frameworks/vecLib.framework/vecLib'
```

- The Fortran compiler in GCC version 3.3 and higher requires the latest assembler, `as`. It can be obtained as a part of `cctools` from <http://www.opensource.apple.com/>. Mac OS X 10.3 (Panther) should come with `cctools` recent enough to compile `PSI3`.
- Certain `PSI3` codes require significant stackspace for compilation. Increase your shell's stacksize limit before running `'make'`. For `cs`, for example, this is done using `'unlimit stacksize'`.

- SGI IRIX 6.x:

- MIPSpro C++ compilers prior to version 7.4 require a command-line flag `'-LANG:std'` in order to compile `PSI3` properly.
- Use command-line flag `'-64'` in order to produce 64-bit `PSI3` executables with MIPSpro compilers. The following is an example of appropriate configure options:

```
--with-cc='cc -64' --with-cxx='CC -64 -LANG:std' --with-fc='f77 -64'
```

- Under IRIX configure will attempt to detect automatically and use the optimized SGI Scientific Computing Software Library (SCSL).