Installation Manual for the PSI3 Program Package

T. Daniel Crawford,\textsuperscript{a} C. David Sherrill,\textsuperscript{b} and Edward F. Valeev\textsuperscript{b}

\textsuperscript{a}Department of Chemistry, Virginia Tech, Blacksburg, Virginia 24061
\textsuperscript{b}Center for Computational Molecular Science and Technology, Georgia Institute of Technology, Atlanta, Georgia 30332-0400

Version of: January 11, 2003
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

- Various GNU utilities: www.gnu.org
  - autoconf
  - 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

The following series of steps will configure and build the PSI3 package and install the executables in.

Renaming your PSI3 source directory: the PSI3 source will usually go into a place like /usr/local/psi3. It should not be named /usr/local/psi, because 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 there is a bug in dvips which will cause the compilation of documentation to fail if any periods appear in the source path.

1. cd $PSI3 (your top-level PSI3 source directory)
2. autoconf
3. mkdir objdir
4. cd objdir
5. ../configure
6. make

7. make install

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. The default search order for compilers is: gcc, cc, ccc. (NB: On AIX systems, the search order is cc_r, cc, gcc, xlC.)

- **--with-cxx=compiler** — Use this option to specify a C++ compiler. The default search order for compilers is: g++, c++, cxx. (NB: On AIX systems, the search order is xlC_r, xlC, c++, g++.)

- **--with-fc=compiler** — Use this option to specify a Fortran-77 compiler. The default search order for compilers is: g77, f77, fc, f2c. (NB: On AIX systems, the search order is xlF_r, xlF, g77, f77, fc, f2c.)

- **--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=5) 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=4) 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=3) 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=4) 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.

• **--with-parallel=sgi** — This option turns on automatic parallelization available with some SGI systems. Since none of the primary developers of PSI3 actually *uses* SGI systems at present, it seems likely that this option will not work.

### 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` 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, followed by the executable modules, and finally documentation (if LaTeX is available).

#### 3.3 Step 3: 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`, ba-
sis set data and other control structures $prefix/share$, and documentation in $prefix/doc$.

### 3.4 Step 4: Testing

An ever-growing number of test cases is available in $PSI3/tests$. A simple automated system is now available to execute all the current test cases and compare to the available reference data. Simply run the executable perl script, `driver.test.pl`, and the result of each test case will be printed in the text file, `test-case-results`. The PSI3 User’s Manual provides detailed instructions for running calculations. Report bugs to psimaster@sirius.chem.vt.edu.

### 3.5 Step 5: Cleaning

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