Installation Guide

Schrödinger Software Release 2015-2

Schrödinger Press

Installation Guide Copyright © 2015 Schrödinger, LLC. All rights reserved.

While care has been taken in the preparation of this publication, Schrödinger assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Canvas, CombiGlide, ConfGen, Epik, Glide, Impact, Jaguar, Liaison, LigPrep, Maestro, Phase, Prime, PrimeX, QikProp, QikFit, QikSim, QSite, SiteMap, Strike, and WaterMap are trademarks of Schrödinger, LLC. Schrödinger, BioLuminate, and MacroModel are registered trademarks of Schrödinger, LLC. MCPRO is a trademark of William L. Jorgensen. DESMOND is a trademark of D. E. Shaw Research, LLC. Desmond is used with the permission of D. E. Shaw Research. All rights reserved. This publication may contain the trademarks of other companies.

Schrödinger software includes software and libraries provided by third parties. For details of the copyrights, and terms and conditions associated with such included third party software, use your browser to open third_party_legal.html, which is in the docs folder of your Schrödinger software installation.

This publication may refer to other third party software not included in or with Schrödinger software ("such other third party software"), and provide links to third party Web sites ("linked sites"). References to such other third party software or linked sites do not constitute an endorsement by Schrödinger, LLC or its affiliates. Use of such other third party software and linked sites may be subject to third party license agreements and fees. Schrödinger, LLC and its affiliates have no responsibility or liability, directly or indirectly, for such other third party software and linked sites, or for damage resulting from the use thereof. Any warranties that we make regarding Schrödinger products and services do not apply to such other third party software or linked sites, or to the interaction between, or interoperability of, Schrödinger products and services and such other third party software.

May 2015

Contents

Document Conventions

In addition to the use of italics for names of documents, the font conventions that are used in this document are summarized in the table below.

Font	Example	Use
Sans serif	Project Table	Names of GUI features, such as panels, menus, menu items, buttons, and labels
Monospace	`$SCHRODINGER/maestro`	File names, directory names, commands, environment variables, command input and output
Italic	*filename*	Text that the user must replace with a value
Sans serif uppercase	CTRL+H	Keyboard keys

Links to other locations in the current document or to other PDF documents are colored like this: Document Conventions.

In descriptions of command syntax, the following UNIX conventions are used: braces { } enclose a choice of required items, square brackets [ ] enclose optional items, and the bar symbol | separates items in a list from which one item must be chosen. Lines of command syntax that wrap should be interpreted as a single command.

File name, path, and environment variable syntax is generally given with the UNIX conventions. To obtain the Windows conventions, replace the forward slash / with the backslash \ in path or directory names, and replace the $ at the beginning of an environment variable with a % at each end. For example, `$SCHRODINGER/maestro` becomes `%SCHRODINGER%\maestro`.

Keyboard references are given in the Windows convention by default, with Mac equivalents in parentheses, for example CTRL+H (⌘H). Where Mac equivalents are not given, COMMAND should be read in place of CTRL. The convention CTRL-H is not used.

In this document, to *type* text means to type the required text in the specified location, and to *enter* text means to type the required text, then press the ENTER key.

References to literature sources are given in square brackets, like this: [10].

Installation Notes

This document gives information on the installation of the Schrödinger Suite 2015-2 software, which comprises Maestro 10.2, BioLuminate 1.9, Canvas 2.4, CombiGlide 3.7, ConfGen 3.2, Core Hopping 2.1, Desmond 4.2, Epik 3.2, Glide 6.7, Impact 6.7, Jaguar 8.8, Liaison 6.7, LigPrep 3.4, MacroModel 10.8, Materials Science Suite 1.8, MCPRO$^+$ 2.0, Phase 4.3, Prime 4.0, QikProp 4.4, QSite 6.7, Schrödinger KNIME Extensions 2.11, SiteMap 3.5, Strike 3.0, and WaterMap 2.3, in addition to Field-Based QSAR, Induced Fit Docking, Ligand & Structure-Based Descriptors, Physics-Based ADME/Tox, Quantum-Polarized Ligand Docking, and Virtual Screening Workflow.

Schrödinger software is supported on Linux platforms, on Mac platforms, and on Windows, with a few exceptions. In this manual, the designation "UNIX", where a broader term is needed, includes Linux.

If you have difficulty with the installation, please contact your system manager or Schrödinger (by phone at (503) 299-1150, or via the Support Center).

1.1 Linux Installation Process Summary

This is a summary of the installation process. For detailed instructions, see the page number provided in each step.

1. Check this guide for:
   - System requirements (page 5)
   - Disk space requirements (page 9)
   - Product-specific installation information (page 9)
2. Mount the DVD (page 15) or download the software from the Schrödinger Download Center. If you download the software, extract the downloaded tar file:

   ```
   tar xvf Schrodinger_Internet_Download.tar
   ```
3. Run the `platform` script to verify that your machine meets the system requirements (page 16).
4. Run the `INSTALL` script to install the products (page 17).
5. Set the `SCHRODINGER` environment variable (page 19).

6. Obtain and install a license for the products (see http://www.schrodinger.com/licensing).
7. Test the Maestro installation by typing `$SCHRODINGER/maestro`.

If jobs will be run on remote hosts:

8. Set up the hosts file (page 47).
9. Set up communication between hosts, if not already done (page 59).
10. *Optional:* Prepare for Batch Queue Submission (page 63).
11. Test the installation and communications (page 92).

1.2 Windows Installation Process Summary

1. Insert the product DVD into the DVD drive, or double-click the installer icon.

   The `setup` program should start.
2. Select an installation directory that does not contain spaces.
3. Select the products and profiles you want to install.
4. Obtain and install a license[1] for the products (see http://www.schrodinger.com/licensing).
5. Test the Maestro installation by double-clicking the Maestro icon on the desktop.

If jobs will be run on remote hosts:

6. Set up the hosts file (page 47).
7. Set up communication between hosts, if not already done (page 60).
8. *Optional:* Prepare for Batch Queue Submission (page 63).
9. Test the installation and communications (page 92).

1.3 Mac Installation Process Summary

1. Open the disk image on the product DVD or in your Downloads folder.
2. Double-click the installer package (`.pkg`).

   The installer should start.
3. Follow the prompts and enter a password when requested.

1. For installation of Maestro only at academic institutions, the license is provided in the same location as the `setup` program, and should be copied to the installation directory.

4. Obtain and install a license[2] (see http://www.schrodinger.com/licensing).
5. Test the Maestro installation by double-clicking the Maestro icon on the desktop.

If jobs will be run on remote hosts:

6. Set up the hosts file (page 47).
7. Set up communication between hosts, if not already done (page 59).
8. *Optional:* Prepare for Batch Queue Submission (page 63).
9. Test the installation and communications (page 92).

2. For installation of Maestro only at academic institutions, the license is provided in the installation.

Hardware and Software Requirements

2.1 General Requirements

This section lists the general minimum system requirements and recommendations for Schrödinger products. If the product-specific requirements differ from those listed below, they are given in Section 2.3 on page 9.

2.1.1 Hardware

2.1.1.1 CPU, Memory, and Disk

- x86_64 compatible processor
- 1 GB memory minimum, 2 GB recommended, per core
- For software installation, 9 GB disk space (see below)
- For computational jobs, 4 GB scratch disk space minimum; 60 GB recommended, at 10000 RPM
- Larger cache size for improved performance of most Schrödinger software
- Network card with a configured network interface

2.1.1.2 Graphics Card and Monitor

Required:

- 16-bit color.

Strongly recommended:

- 1280 x 1024 minimum resolution.
- Graphics card that supports hardware-accelerated OpenGL
- Up-to-date vendor-supplied graphics driver.The driver supplied with the operating system is usually inadequate. If you upgrade the operating system, the driver *must* be reinstalled.
- Monitor with a refresh rate of 100 Hz or more.

For hardware stereo viewing, one of the following requirements must be met in addition to the recommendations:

- A graphics card that supports quad-buffered OpenGL stereo, or
- A monitor that supports interlaced stereo.

On Mac OS X, only interlaced stereo is supported.

Not meeting the recommendations can significantly degrade Maestro performance.

Note: An out-of-date graphics driver is the most common cause of Maestro display problems. An inadequate graphics card is the second most common cause.

2.1.1.3 GPGPU Computing

GPGPU computing with CUDA is supported on Linux only for the following:

- NVidia Tesla K20 or K40 graphics card
- Minimum graphics driver version 331.20

3GB of video memory is required, and a Compute Capability of at least 2.0. Other NVidia cards may work but are not recommended; cards from other vendors are not supported. See https://developer.nvidia.com/cuda-gpus for the full list of available models.

2.1.1.4 Clusters

The following components are recommended for optimizing cluster performance, in addition to the general hardware requirements and recommendations given above:

- A highly capable file server for the external network.
- A high-performance intra-cluster network (especially for Desmond).
- Shared storage for the intra-cluster network, to reduce traffic to and from the external network.
- Fast processors, large memory, and high-quality motherboards and network interfaces, especially on the management nodes.

2.1.2 Linux

Basic requirements:

- One of the following 64-bit operating systems:
  - RHEL 6.4 or higher 6.x, 7.0
  - CentOS 6.4 or higher 6.x, 7.0
  - SUSE SLES 11 SP3, SLED 11 SP3, OpenSUSE 13.1
  - Ubuntu 10.04 LTS, 12.04 LTS, 14.04 LTS
- NFS file locking must be enabled.

Software requirements for documentation:

- Adobe Reader no earlier than 7.0.5 for searching manuals.
- PDF reader other than Adobe Reader for viewing manuals. If using Evince, a version no earlier than 2.0 is required. Okular is recommended, if available.

RedHat/CentOS requirements:

- The `nfs-utils` package must be installed.

```
yum install nfs-utils
```

Ubuntu requirements:

- The `mesa-utils` and `lsb-core` packages must be installed:

```
sudo apt-get install mesa-utils
sudo apt-get install lsb-core
```

2.1.3 Windows

All products and job types are supported for local use on Windows except for Desmond simulations, MCPRO+, Jaguar jobs using multiple selected entries. Remote job submission from Windows is available for Desmond and WaterMap.

The following operating systems are supported. Both 32-bit and 64-bit Schrödinger executables are available.

- Windows 8, and 8.1, 64-bit only, classic desktop only
- Windows 7 SP1, 32-bit and 64-bit
- Windows Vista SP2, 32-bit and 64-bit
- Windows Server 2012, 2012 R2, 64-bit only, classic desktop only
- Windows Server 2008, 2008 R2, 2012, 2012 R2, 64-bit only

Only the classic desktop is supported on Windows 8.x and Windows server 2012. Windows 7 support includes English, French, German, Spanish, and Japanese versions.

In addition, the following software is required:

- Microsoft Visual Visual C++ 2010 SP1 Redistributable Package for both 32-bit (x86) and 64-bit Maestro executables (x64). This package is included with the software distribution, and can be installed as needed. You will be prompted to install it during installation if it is not already installed.
- PDF reader for display of manuals. If Adobe Reader is installed, a version no earlier than 7.0.5 is recommended. Adobe Reader 8.0 is not supported.

2.1.4 Mac

Requirements for Mac computers are as follows:

- Intel processor, 64-bit only
- Mac OS X 10.7, 10.8, 10.9, or 10.10.
- Adobe Reader is required to use the search index for the manuals, and for proper functioning of the facility to open tutorial documentation. Preview does not follow links between the distributed PDF documents.

Note: This is the last release for which OS X 10.7 is supported.

All Schrödinger products except MCPRO$^+$ and Desmond are supported. Remote job submission from Mac is available for Desmond and WaterMap.

Quad-buffered hardware stereo is not supported; interlaced stereo is supported.

If you need to compile Prime third-party programs, you must install XCode, including the command-line tools. This requires an Apple Developer account.

2.1.5 Queueing Systems

The following queueing systems are supported:

- PBS, including PBS Pro
- Grid Engine, including SGE, Open Grid Scheduler, and Univa GE, minimum version 6.2
- LSF, minimum version 7.0.2
- SLURM, minimum version 2.1
- Torque

Information on supported features (license checking and GPU support) is given in Table 2.1. See Section 7.9 on page 90 for more information on setting up license checking.

Table 2.1. Queueing system features.

Queueing system	License checking	Native GPU support version
PBS Pro	Yes	11.0
LSF	Yes	9.1
Univa Grid Engine	Yes	8.1
Sun Grid Engine, Open Grid Scheduler	Yes	None
Torque	No	2.5.4
SLURM	No	2.2

2.2 Disk Space for Installation

Disk space requirements for a full installation of Schrödinger software, including the documentation, are given in Table 2.2. These values do not include the PDB or BLAST databases which require 14 GB and 13 GB. The Pfam database (included) requires less than 1 GB.

Table 2.2. Disk space for a full software installation.

Platform	Disk space
Linux-x86_64	5.5 GB
Windows	6.3 GB
Windows-x64	6.5 GB
Mac OS X	6.4 GB

Note: On Windows, a copy of the entire installation is kept for backup and repair. You must therefore have the space locally to store a copy of the entire installation, in addition to the space for the installed software on the drive to which you are installing, space for the download (if relevant) and for the unpacked MSI files. This may mean you need up to 4 times the disk space listed above to perform the installation.

2.3 Product Notes

This section contains notes specific to each product. These notes give information in addition to the requirements listed above. Only external software dependencies are listed below. Licenses must be obtained for each installed product.

2.3.1 Desmond 4.2

Desmond is available only on Linux as 64-bit (x86_64) executables. Desmond jobs can be submitted from Mac and Windows hosts to Linux hosts.

Desmond can be run in parallel using Open MPI. The mmshare distribution includes Open MPI 1.6.5 for parallel execution. Desmond does not support other MPI implementations, such as MPICH. Parallel execution is supported on SMP hosts as well as clusters.

2.3.1.1 Network Recommendations for MPI Use

A high-performance network is highly recommended for parallel execution. Desmond supports the use of ethernet and Infiniband networks for parallel execution. Use of Infiniband requires a Linux kernel no earlier than 2.6.9, and OFED 1.2.

Myrinet is not supported.

If you are using an Infiniband network, the following installation issues must be addressed:

- The system must be configured to allow unlimited locked memory. You must do at least the following:

  a. Add the following lines to `/etc/security/limits.conf` on all nodes:

  ```
  * soft memlock unlimited
  * hard memlock unlimited
  ```

  b. Add the following lines to the appropriate startup scripts:

  csh, tcsh: `limit memorylocked unlimited`

  bash, ksh: `ulimit -l unlimited`

  You should do this in one of the following locations:

  - `/etc/init.d/sshd`
  - `/etc/profile` (bash) and `/etc/csh.cshrc` (csh/tcsh)
  - `$HOME/.bashrc` (bash) or `$HOME/.cshrc` (csh/tcsh)

  You should also add these lines to the startup scripts for the resource manager daemons for the Torque, PBS, LSF queuing systems.

  This is necessary because the limit may be reset to a lower number by some boot procedures, or resource managers might not start with unlimited locked memory.

  When you have added these lines, you must restart the resource management daemon.

2.3.1.2 Use of Graphics Processors (GPUs)

Desmond can be run in parallel on GPGPUs with CUDA. See Section 2.1.1.3 on page 6 for the supported graphics cards.

2.3.2 Jaguar 8.8

1 GB scratch disk space minimum per process is recommended. Large jobs, such as frequency and LMP2 calculations, can use several gigabytes of scratch disk space.

Use local disks for scratch space. Performance is significantly reduced if an NFS-mounted scratch disk is used.

Jaguar can be run in parallel with MPI on Linux and Mac OS X. By default, Jaguar uses Open MPI 1.6.5, which is included in the Schrödinger software distribution. Jaguar runs on shared-memory architectures in SMP mode, or on distributed architectures and clusters. Jaguar cannot be run in parallel on Windows.

Jaguar can also be run in parallel with OpenMP, on Linux only (not Mac OS or Windows). Both OpenMP and Open MPI can be used in the same job.

To run parallel Jaguar jobs from a batch queue, some queue configuration might be needed—see Section 7.3.3 on page 65 for more information.

2.3.3 KNIME Extensions 2.11

The Schrödinger KNIME Extensions for 2015-2 are built on KNIME 2.4.1. This is the required minimum version. If you add the Schrödinger KNIME Extensions to an existing KNIME installation, you must have this version installed at a minimum.

On Linux, the `libstdc++.so` version installed in `/usr/lib64/` must be no less than 6.0.14. If you cannot upgrade your installation and you are using an external version of KNIME, you must set `LD_LIBRARY_PATH` to include the following path:

KNIME-installation`/plugins/com.schrodinger.knime_`*version*`/lib/Linux-x86_64/`

Schrödinger KNIME Extensions require a full installation of Schrödinger software.

2.3.4 MCPRO+ 3.8

MCPRO$^+$ is only supported on Linux platforms.

2.3.5 Prime 4.0

Prime requirements cover Prime, Induced Fit Docking, and Covalent Docking.

Prime requires the installation of or access to various third-party products. See Section 2.4 on page 11 for more information.

2.4 Prime Third-Party Software and Databases

To use Prime, you must install or have access to the PDB, the BLAST program and associated sequence databases, and the HMMER program and Pfam database. The required third-party programs and databases for web-based searching are provided in the download and on DVD.

Use of PSIPRED is also highly recommended for secondary structure predictions and for GPCR modeling. PSIPRED is not available on Windows. PSIPRED is not distributed as part of the Schrödinger software package—see Section 3.6 on page 20 for information on obtaining and installing this program on Linux.

The databases are available on DVD. If you downloaded Prime and do not have the databases, instructions for obtaining them are provided in Section 3.6 on page 20 for Linux, Section 4.2

on page 31 for Windows, and Section 5.2 on page 35 for Mac OS.

- For Induced Fit Docking, you do not need to install the third-party programs or databases.
- For Homology Modeling, you do not need to install the PDB or BLAST databases if you have access to these databases on the web.
- If you do not intend to identify families for your query sequence, you do not need to install HMMER and Pfam.

Disk space requirements are listed in Section 2.2 on page 9.

If you choose to use your own installations of the third-party software, the following versions are required:

- BLAST 2.2.26
- PSIPRED 2.61
- HMMER 3.0 and Pfam 26.0.

Note: The current version of Pfam is incompatible with previous versions distributed with Schrödinger software. If you have been using a version of Pfam that is installed outside the Schrödinger installation, you should ensure that you set the appropriate environment variables to point to a compatible version—see Table 2.3.

If you install these third party products from the DVDs supplied by Schrödinger, you must run the `INSTALL` script for each DVD. Do not change DVDs while the `INSTALL` script is being executed: if you do, the script will fail.

After you have installed the software, you may need to set environment variables to identify the location of the software, depending on where it is installed:

- If you install *all* of these third-party products and databases into the default location (`$SCHRODINGER/thirdparty`), you do not need to set any extra environment variables. However, when you install a new Schrödinger software release, you must make sure that these products and databases are installed in the new default location.
- If you install *all* of these third-party products and databases in the same, nondefault location, you can set `SCHRODINGER_THIRDPARTY` to the chosen location. This location should have a `database` directory, in which databases are stored and a `bin` directory, in which executables are stored.
- If you install *any* of these third-party products in a location other than the default location, `$SCHRODINGER/thirdparty`, you must set environment variables, listed in Table 2.3, to identify the location of the products that are in a nondefault location
- If you already have copies of the third-party products, you can provide links to them

using the environment variables described in Table 2.3. You do not need to set these environment variables if you are installing Prime from the supplied DVDs.

On Windows, if you are using an installed copy of the PDB to run Structure Prediction jobs, you must set the SCHRODINGER_PDB environment variable in Windows or UNC format. See Appendix A for information on setting environment variables on Windows.

You can set the environment variables for remote hosts in the schrodinger.hosts file (see Section 7.1 on page 47 for more information). .

Table 2.3. Environment variables defining the nonstandard location of third-party software and databases for Prime.

Environment Variable	Description
SCHRODINGER_PDB	PDB distribution directory (contains the data directory). Default: $SCHRODINGER/thirdparty/database/pdb.
PSP_BLASTDB	BLAST database directory (contains directories nr and pdb). Default: $SCHRODINGER/thirdparty/database/blast.
PSP_BLAST_DIR	BLAST executable directory. Default: $SCHRODINGER/thirdparty/bin/*platform*/blast/bin.
PSP_BLAST_DATA	BLAST matrices directory. Default: $SCHRODINGER/thirdparty/bin/*platform*/blast/data.
PSP_HMMER_DIR	HMMER executable directory. Default: $SCHRODINGER/thirdparty/bin/*platform*/hmmer.
PSP_HMMERDB	Pfam database directory. Default: $SCHRODINGER/thirdparty/database/pfam.
PSP_PSIPRED_DIR	PSIPRED installation (contains bin and data directories) Default: $SCHRODINGER/thirdparty/bin/*platform*/psipred.
PSP_PSIPRED_DB	Identity of PSIPRED sequence database. Allowed values are nr and pdb. Default: pdb.
PSP_SSPRO_DB	Identity of SSPRO sequence database. Allowed values are nr and pdb. Default: pdb.

Information on the third-party software and databases can be found at the following locations:

- BLAST: http://www.ncbi.nlm.nih.gov/blast
- HMMER: http://hmmer.wustl.edu
- Pfam: http://pfam.sanger.ac.uk
- PDB: http://www.rcsb.org/pdb
- PSIPRED: http://bioinf.cs.ucl.ac.uk/psipred/

2.5 Documentation Requirements

Manuals are distributed in PDF format and require a PDF reader. As the manuals have links between documents, the PDF reader should support links between documents.

- Adobe Reader is required on all platforms to use the search index and to open tutorial documentation from the Tutorials panel in Maestro.
- On Windows the default PDF reader is Adobe Reader. Adobe Reader 8.0 is not supported.
- On Linux, an alternative reader is required to use the documentation index and to follow links between manuals, and this must be set as your default PDF reader. Adobe Reader is not recommended for display of the manuals from Maestro because links between documents work only intermittently. You should set the file association for PDF documents to the alternative reader. We recommend Okular or Evince 2.0 or later.
- On Mac, Adobe Reader is used if it is installed. Preview does not support links between the distributed PDF files as of Mac OS X 10.7, so Adobe Reader is recommended.

If you have Adobe Reader installed, we recommend that you use a version no earlier than 7.0.5, and that you ensure that it can be used as a plug-in to your browser. This is so that you can make full use of the indexing and hyperlink features in the documentation.

Installing the Products on Linux

Before installing Schrödinger products and documentation, read Chapter 2 for information on hardware and software requirements, including product-specific requirements. These requirements must be met before installation. If you are installing on a cluster, read Section 7.9 on page 90.

3.1 Access to the Software

The software can be obtained on a DVD or downloaded from the Schrödinger web site. You must ensure that it is accessible from each host on which you want to install it. The directory that contains the software for installation will be called the *installer* directory. When you have completed this section, the installer directory (designated *installer-dir*) should be one of the following locations:

- DVD mount directory
- directory on a remote machine containing the copied files
- *download-directory*`/Schrodinger_Suite2015_`*type*

If you obtained the software on a DVD:

The DVD must be mounted. Most computers automatically mount the DVD when it is inserted into the DVD drive.

Note: DVDs are only readable by a DVD-ROM drive or a DVD-R or DVD+/-R Read/Write drive. DVD+R Read/Write drives cannot read the DVDs we supply.

If you want to install the software on multiple hosts, you can either copy the files to a place that is accessible to each host, as described in Step 3 below, or mount the DVD on each host.

If the DVD is not automatically mounted, consult your system administrator.

If you obtained the software on a DVD and the host you want to install on does not have a DVD drive:

The software must be copied to the desired host, as follows.

1. Mount the product DVD on the machine that has a DVD drive.
2. Change to the mount directory and display the DVD contents.

3. Copy the following files to the remote host:
   - product tar files for your platform
   - `INSTALL` file
   - scripts in the top-level directory
   - Maestro and mmshare tar files for your platform
   - data tar files for your product
   - third-party software and databases for Prime
   - documentation tar file
4. Change file names if necessary.

   If you are installing from a DVD drive on a Windows machine, Windows may change the case of the file names. The tar files and `platform` script should be in lower case and the `INSTALL` script and the `README` file should be in upper case. Use the `TRANS.TBL` file to rename the files with the correct case.

If you downloaded the software:

You must extract the files from the archive (tar) file. The directory to which you downloaded the software is represented by *download-directory* in what follows.

```
cd download-directory
tar xvf Schrodinger_Suite2015_type.tar
```

where type identifies the hardware platform that the software will be installed on and the type of institution installing the software. When you have extracted the files, a subdirectory named `Schrodinger_Suite2015_`*type* is created, and the software is in this subdirectory. The installer directory is therefore *download-directory*`/Schrodinger_Suite2015_`*type*. If you want to install the software on multiple hosts, you should either ensure that *download-directory* is accessible to each host, or copy the archive file to suitable locations and extract it, or copy the files described in Step 3 above to suitable locations.

Note: If you do not want to install all modules of the software, the modules that you do not want must be removed from the directory from which you install prior to starting the installation. The installation installs all modules that are in the same directory as the `INSTALL` script.

3.2 Verifying System Compliance

Before starting the installation, verify that the systems that you are installing on satisfy the minimum requirements to run Schrödinger software. You should follow this procedure for each host on which you plan to run the software. When you have verified the system compliance, proceed to the software installation.

1. Log on to the desired host.
2. Change to the installer directory, which contains the `platform` script.

   `cd` *installer-dir*
3. Enter the following command:

   ```
   ./platform -s
   ```

   The script indicates whether your system meets the requirements or needs to be updated. If you receive an error message, postpone installation of your Schrödinger software until you have updated your system. For help obtaining any missing libraries, see the appropriate product-specific section of this guide.
4. *Optional:* To see a summary of the platform information, enter:

   ```
   ./platform -l
   ```

   The script checks the operating system and distribution, CPU type, number of processors, perl version number, and relevant libraries (`glibc` for Linux).

3.3 Installing the Software and Documentation

When you select the locations for installing the software, you must ensure that the software is accessible from all hosts that are used either to submit jobs or to run jobs. This includes individual nodes on a cluster.

Note: The installation process does not replace the `$SCHRODINGER/schrodinger.hosts` or `$SCHRODINGER/license` files, or files in the `$SCHRODINGER/queues` directory. If you want to install new versions of these files when upgrading your software, you must move or remove them first.

1. Change to the installer directory.
2. Enter the command

   ```
   ./INSTALL
   ```

   This script accepts a number of options for providing the required information. For details, enter the command

   ```
   ./INSTALL -h
   ```
3. Enter the information requested by the `INSTALL` script.
   - You can accept the default values for each question by pressing RETURN
   - You can quit the `INSTALL` script at any time by pressing CTRL-C.

- If you realize you have entered incorrect information, simply press RETURN at all of the prompts, then enter `n` at the confirmation screen to start the questions again.

Below are explanations of the questions asked by the script:

SCHRODINGER directory: This is the *installation* directory, where the executables, data files, and other files related to Schrödinger products will be installed. Depending on the type of license you have, we recommend the following installations:

- *Token-based or IP-based license:* Use a shared file system so that you only have to install the software once and all client machines with access can use it.
- *Node-locked license:* Use the local file system of the machine for best performance, or an NFS-mounted file system if local space is limited.

You must install a new release into a new `SCHRODINGER` directory.When the script has located or created the `SCHRODINGER` directory, it asks you to confirm that the selection is correct. Press RETURN to accept.

Hardware/Software platform: In this screen, the `INSTALL` script reports the machine type and operating system. Press RETURN to continue.

SCHRODINGER_THIRDPARTY directory: This is the directory where third-party programs required by Prime are installed. By default they are installed in the software installation in the `thirdparty` subdirectory, but you can choose a different location.

It is recommended that you do not install the PDB or BLAST databases inside the installation.

Scratch directory: This directory is for the large, temporary files generated by computational programs during calculations. We recommend this directory be located on a fast, local drive with at least 4 GB of disk space. The `INSTALL` script checks for existing directories named `/scr`, `/scratch` or `/usr/tmp` and suggests the first of these as the default.

If you decide to use a different directory, you will need to create it first. The `INSTALL` script will not create it for you. Also, make sure each person who wants to run jobs has write access to the scratch directory.

4. Confirm the information you provided.

   When you have finished entering the information, the `INSTALL` script summarizes your choices. If any of the summary information is incorrect, answer "`n`" at the prompt to run through the questions again. Once you are satisfied with your answers, press RETURN to install the software. The installation can take some time.

5. Record the `machid` information.

   When the installation is complete, the `INSTALL` script runs the `machid` program, which generates machine-specific information about the computer on which it is run. You will need this information to request a license for your Schrödinger software. See http://www.schrodinger.com/licensing for full details on how to request a license.

6. Remove temporary installation directories and files. If you copied tar files onto a remote machine, delete those files now.

Repeat this procedure for all locations where you want to use the software. Once you have installed the software, you must obtain a license to run it—see http://www.schrodinger.com/licensing.

3.4 Setting the Environment Variables

Before you can launch Schrödinger software, you must set some environment variables. In addition to those listed below, there may be product-specific environment variables that need to be set—see the requirements section for each product.

`SCHRODINGER`	required for all Schrödinger products
`DISPLAY`	required for Maestro (usually automatically set on login)

To set the `SCHRODINGER` environment variable, enter the following command, replacing *install-directory* with the full installation path (for example, `/software/Schrodinger2015`):

csh, tcsh:	`setenv SCHRODINGER` *install-directory*
bash, ksh:	`export SCHRODINGER=`*install-directory*

To avoid using the wrong installation when a new installation is made or when you use different installations, we recommend that you do *not* add a `SCHRODINGER` definition to your shell startup script file or add `$SCHRODINGER` to your path definition. Instead, you can use aliases to set `SCHRODINGER` to the appropriate location, for example:

csh, tcsh:	`alias schro2015 "setenv SCHRODINGER` *installdir*`"`
bash, ksh:	`alias schro2015="export SCHRODINGER=`*installdir*`"`

You can also set aliases to specific programs that substitute the current `SCHRODINGER`, so that you do not have to type `$SCHRODINGER` to run a program. For example:

csh, tcsh:	`alias maestro \$SCHRODINGER/maestro`
bash, ksh:	`alias maestro=\$SCHRODINGER/maestro`

An alternative to setting the environment variables directly is to use environment management software, such as the Environment Modules package found at http://modules.sourceforge.net.

If you expect either long delays when a program tries to obtain a license token, or competition between programs for license tokens, you can set the time limit for trying to obtain a license token in the `SCHRODINGER_LICENSE_RETRY` environment variable. This environment variable can be set to time values, such as 300s, 10m, 2h, or to an integer value, which is interpreted as a time in seconds. The default is 10 minutes.

To use PyMOL from Maestro or Canvas, you need to add the top-level PyMOL directory to the `PATH` and `PYMOL4MAESTRO` environment variables.

To check that you have set the `SCHRODINGER` environment variable correctly, start Maestro with the following command:

```
$SCHRODINGER/maestro &
```

You can also set environment variables for the running of jobs for each host in the `schrodinger.hosts` file. See Section 7.1 on page 47 for more information. Environment variables that are set in the hosts file are not used for running Maestro, so they must be set in the shell (or globally).

For a list of Schrödinger environment variables, see Appendix B of the *Job Control Guide*.

3.5 Enabling Hardware Stereo Viewing

To run Maestro in hardware stereo mode, you must edit `/etc/X11/xorg.conf` to set the driver in stereo-capable mode. For information on the option that is required, consult the documentation for your graphics card. The required options for Nvidia cards are

```
Option "Stereo" "3"
```

in the Device section, and in the Extensions section

```
Option "Composite" "Disable"
```

3.6 Installing Prime Third-Party Software and Databases from a Download

Prime third-party products are included in the web download and are available on the Prime DVD set. They are installed by default. The Pfam database is also installed by default, as well as data required for web-based BLAST searches. PSIPRED is not included or required, but is highly recommended.

Prime can obtain PDB files and run BLAST searches on the web or on a local database. For web-based searches, the default Prime installation is sufficient. For local database access the PDB and BLAST databases must be installed.

If you already have any of the third-party products installed and want to use the existing installation rather than installing another copy, you can do so by setting the appropriate environment variables, which are described in Section 2.4 on page 11. Likewise, if you want to install any of these products in a location other than the standard location in `$SCHRODINGER/thirdparty`, you should use that location in the relevant steps below, and set the appropriate environment variables. The instructions below assume that you are installing into the standard location.

It is assumed that you have already installed the Prime software and set the `SCHRODINGER` environment variable.

To install the PDB and BLAST databases:

- Run the following scripts:

```
$SCHRODINGER/utilities/update_BLASTDB
$SCHRODINGER/utilities/rsync_pdb
```

The BLAST databases are installed in the first location found in the following list:

- `$PSP_BLASTDB`
- `$SCHRODINGER_THIRDPARTY/database/blast`
- `$SCHRODINGER/thirdparty/database/blast`

Likewise, the PDB is installed in the first location found in the following list:

- `$SCHRODINGER_PDB`
- `$SCHRODINGER_THIRDPARTY/database/pdb`
- `$SCHRODINGER/thirdparty/database/pdb`

If an environment variable in these lists is not defined, the list item is skipped. The environment variables are described in Section 2.4 on page 11.

To compile and install PSIPRED:

1. Create a directory in which to download and build PSIPRED.

   This directory will be referred to as the build directory and labeled *build-dir*.

2. Download `psipred261.tar.gz` into the build directory. from the PSIPRED download site at
   http://bioinfadmin.cs.ucl.ac.uk/downloads/psipred/old/psipred261.tar.gz.

(If this location changes, the PSIPRED home page is located at http://bioinf.cs.ucl.ac.uk/psipred, and contains information about the program, referrals to terms of use and license terms in the README file, and a link to download the program. The link to download PSIPRED is in the section of the page labeled "Software Download". You must download version 2.61 to work with Prime 4.0.)

3. Change to the build directory and unpack the source code:

   ```
   cd build-dir
   tar xzvf psipred261.tar.gz
   ```

4. Compile the code:

   ```
   cd src
   make
   make install
   cd ..
   ```

5. Create a directory in `$SCHRODINGER` for the software (if it does not already exist):

   ```
   mkdir -p $SCHRODINGER/thirdparty/bin/Linux-x86_64/psipred
   ```

6. Copy the `bin` and `data` directories from the build directory:

   ```
   cp -r bin/ $SCHRODINGER/thirdparty/bin/Linux-x86_64/psipred/
   cp -r data $SCHRODINGER/thirdparty/bin/Linux-x86_64/psipred/
   ```

7. Check the final layout:

   ```
   ls -R $SCHRODINGER/thirdparty/bin/Linux-x86_64/psipred/
   ```

   The layout should look as follows:

```
.:
   bin  data

   ./bin:
   pfilt  psipass2  psipred  seq2mtx

   ./data:
   weights.dat   weights.dat3  weights_p2.dat  weights_s.dat2
   weights.dat2  weights.dat4  weights_s.dat   weights_s.dat3
```

3.7 Installing VMD for Desmond

To download VMD:

1. Go to the following web site:

   http://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD

2. In the section Version 1.8.7 (2009-08-01) Platforms, click on LINUX OpenGL, CUDA or on LINUX_64 OpenGL, CUDA.

   The choice depends on whether you want to use the 32-bit version or the 64-bit version. When you click on the link, a login page is displayed. If you do not already have an account you can request one from this page and then log in. When you have logged in, the download starts.

3. Choose a location to download the tar file, `vmd-1.8.7.bin.LINUX.opengl.tar.gz`.

To install and configure VMD:

1. Change to the location of the tar file and extract it:

   ```
   cd location-of-tar-file
   tar -zxvf vmd-1.8.7.bin.LINUX.opengl.tar.gz
   ```

2. Change to the VMD directory:

   ```
   cd vmd-1.8.7
   ```

   This directory contains a README file that has installation instructions. These instructions are included in the following steps.

3. Edit the `configure` file and set the values of the `install_bin_dir` and `install_library_dir` variables to the locations where you wish to install the VMD binary and its supporting libraries.

   `$install_bin_dir` is the location where the startup script `vmd` will be installed. It should be set to a location in the path of anyone wanting to run VMD. `$install_library_dir` is the location of all other VMD files, including the binaries and helper scripts. It should not be in the path.

4. Run the configuration script to generate a Makefile:

   ```
   ./configure
   ```

5. Install VMD:

   ```
   cd src
   make install
   ```

6. Make sure that the directory containing the `vmd` executable (the one that you listed for `install_bin_dir` inside the `configure` file above) is in the path for your shell.

   For instance, if that directory was `/usr/local/bin/vmd-1.8.7` and you are using the bash shell you could use the following command:

   ```
   PATH=$PATH:/usr/local/bin/vmd-1.8.7
   ```

To test VMD, type the command:

```
vmd
```

Two windows should be opened, VMD main and VMD *version* OpenGL Display.

3.8 Setting Up Access to PyMOL

Both Maestro and Canvas can open PyMOL directly. To do so, PyMOL must be installed and the location communicated to the application. The recommended procedure is:

1. Install PyMOL in a location outside `$SCHRODINGER`, using `setup.sh`.

   You must use this script to ensure that the PyMOL installation is configured properly.

2. Add the PyMOL top-level directory to the `PATH` environment variable.
3. Set the `PYMOL4MAESTRO` environment variable to the PyMOL top-level directory.

Canvas requires the PyMOL location to be in the `PATH` environment variable. Maestro does not use `PATH`, but uses the following sources to find an installation of PyMOL, in the order given:

1. The `PYMOL4MAESTRO` environment variable. This environment variable should point to the top directory of the PyMOL installation. This is the preferred method of specifying the location of PyMOL.
2. A PyMOL launch script called `pymol4maestro` in the Schrödinger software installation. This script must run PyMOL.
3. A PyMOL installation in `$SCHRODINGER/pymol`.
4. The standard PyMOL environment variable `PYMOL_PATH`.
5. A launch script called `$SCHRODINGER/pymol`.

3.9 Uninstalling the Software

To uninstall the software for a release, delete the release directory:

`rm -rf` *install-directory*

Installing the Products on Windows

4.1 Installing Schrödinger Software

We recommend that you install Schrödinger software as an administrator, and install for all users (AllUsers mode). However, if you do not have administrator privileges, you can install Schrödinger software in single-user (OnlyForMe) mode. To do so, you must have permission to write in the User folder of the registry and permission to create a directory that does not have spaces in the path (see below).

If Schrödinger software is installed by an administrator in OnlyForMe mode, the mode cannot be changed to AllUsers. Instead, the software must be uninstalled and reinstalled in AllUsers mode.

If you are repairing or removing software, you must ensure that none of the programs that are part of the installation are running at the time, as this will prevent the operation from finishing successfully.

After you have installed the software, you must then obtain a license. If you are an academic user installing only Maestro, the license is included and installed automatically.

To install Schrödinger software:

- If you have a DVD, `setup.exe` starts automatically after the DVD is placed in the drive.
- If you have downloaded the software, first unzip the downloaded zip archive, `Schrodinger_Internet_Download.zip`, then double-click the installer icon or run `setup.exe`.

Note: Do not try to install the `.msi` files directly. They must be installed using `setup.exe`.

4.1.1 Preparing for Installation

Successful installation of Schrödinger Suite 2015-2 may depend on taking action on one or more of the following items.

- If you are using product management software, you should be aware that Schrödinger software includes files with a `.ini` extension that are not Windows `.ini` files. You may have to configure your management software to treat these files as plain files.

- Check that you have enough disk space (see Section 2.2 on page 9). The installer requires space on the local drive to unpack the MSI files and store a backup copy. You must therefore have enough space locally to store a copy of the entire installation, in addition to the space for the installed software on the drive to which you are installing and the space taken by the download.
- Installation log files are written by default to the following folder[1]:

  `%LOCALAPPDATA%\Schrodinger\Installer\2015-2`

  These log files may be useful for troubleshooting. If you want to change the location, set the environment variable `SCHRODINGER_INSTALLER_LOGFILE_DIR` to the desired location, which must exist and have write permission. See Appendix A for information on setting environment variables.
- If you cannot find a network-mounted drive that you want to use, it may be due to settings made by User Access Control, under which access to network shares may use the standard network token instead of the Administrator token. To work around this problem, create a registry key under HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System, named EnableLinkedConnections and of type DWORD, and set its value to 1.

4.1.2 Installation Notes

The following notes describe choices that must be made or conditions that must be met when you install the software:

- If you are installing the software as administrator, and you browse to create a directory and rename it, you must press F5 to force the dialog box to update the directory to the new name, so that it is no longer shown as New Folder.
- The default installation folder is `C:\Program Files\Schrodinger-2015-2`, or `C:\Program Files (x86)\Schrodinger-2015-2` for 32-bit executables on a 64-bit OS. If you want to install into another folder, it cannot be a drive, such as `C:`, and it cannot be the same as a previous Schrödinger software installation. The installer does not allow you to choose the same directory as a previous release installation.
- The software distribution includes Schrödinger products or clients for running products remotely, in addition to Maestro. To install products, you must select the corresponding modules in the installer. If you want to run jobs either on the local Windows host or on a remote Linux host, you must install the product or client on the local Windows host.

1. `%LOCALAPPDATA%` is equivalent to `%USERPROFILE%\AppData\Local`.

Table 4.1. Modules to install for each product.

Product	Modules	Product	Modules
BioLuminate	BioLuminate Desmond Epik MacroModel Prime	LigPrep	MacroModel Epik (optional)
		MacroModel	MacroModel
		Maestro	maestro
Canvas	Canvas		
CombiGlide	CombiGlide Impact MacroModel QikProp	P450 Site of Metabolism Prediction	Epik Impact Prime
		Phase	Phase MacroModel
ConfGen	MacroModel		
		Prime	Prime
Core Hopping	CombiGlide Impact MacroModel Phase	PrimeX	Prime Impact Epik MacroModel
Desmond	Desmond		
		QikProp	QikProp
Epik	Epik		
		QM-Polarized Ligand Docking	Impact Jaguar
Glide	Impact		
Impact	Impact	QSite	Impact Jaguar
Induced Fit Docking	Impact Prime	SiteMap	Impact
Jaguar	Jaguar	Strike	Maestro
KNIME Extensions	Knime	Virtual Screening Workflow	Impact Epik MacroModel Prime QikProp
Liaison	Impact		
Ligand & Structure-Based Descriptors	Impact MacroModel Prime QikProp	WaterMap	WaterMap Desmond

- If you want to use Prime or BioLuminate, you must have access to the PDB and BLAST databases. If you do not have web access, you must install them—see Section 4.2 on page 31 for instructions.
- You can choose to create shortcuts for selected Maestro profiles. These profiles present Maestro with different customizations for different user bases.
- To install custom profiles, copy them to the `apps\customize\profiles` folder in the unzipped installer folder before you install. These profiles are presented along with the standard profiles when the screen for selecting profiles is displayed by the installer.
- To install an existing hosts file rather than create a default hosts file, copy the desired `schrodinger.hosts` file to the `apps\customize` folder in the unzipped installer folder before you install.
- To install the license file (`license.txt`) automatically, copy it to the `apps\customize` folder in the unzipped installer folder before you install.
- If your virus checker flags any executable in the distribution as a risk, make an exception for that executable. `SCHROD.exe` is flagged by some virus checkers, for example.

Documentation for all products is installed and uninstalled automatically, along with the shared libraries, `mmshare`.

4.1.3 Installing From the DOS Command Line

If you want to install the software from the command line, without using the graphical interface, you can run the installer in a DOS shell in silent mode, as follows:

```
setup-silent.exe [/install|/remove|/repair] options
```

The options are described in Table 4.2.

To check the progress of the installation you can look at the log files, which are located in `%LOCALAPPDATA%\Schrodinger\Installer\2015` by default, and contain the following information:

`schrodingerSetup.log`	Overall progress.
`schrodingerInstaller.log`	Information for the product installer that was last run or is currently running.

Table 4.2. Options for setup-silent.exe

Option	Description
`/install`	Install products. This is the default, and includes all products.
`/installdir:`*directory*	Installation directory. If mmshare is already installed, the directory in which it is installed is used and this directory is ignored; the change is noted in the log file. The default is `C:\Program Files\Schrodinger-2015-2`, or `C:\Program Files (x86)\Schrodinger-2015-2` for 32-bit executables on a 64-bit OS.
`/logpath:`*directory*	Directory for log files. Default is `%USERPROFILE%\SchrodingerInstaller\2015`
`/plist:`*filename*	Full path to text file containing list of "products" to process. The file must contain one product name per line. The product names are case-sensitive. The product name can be obtained from the initial string, up to the version string (`-v`*x*`.`*y*), on the product line in the `GUIDs.ini` file, e.g. maestro, phase-client. If this option is omitted, all products are processed for the specified action (install, remove, repair).
`/profiles:{all|`*profiles*`}`	Install the specified profiles, given as a comma-separated list. Valid standard profiles are Maestro, BioLuminate, MaterialsScience, Elements, Elements-Basic, Elements-Intermediate, FirstTimeUser, and Jaguar. Custom profiles can be installed if they are added to `apps\customize\profiles` prior to installation. The values are case-sensitive. By default all profiles are installed (including custom profiles). Only valid with `/install`.
`/remove`	Remove products rather than install them. Default is to remove all, including documentation.
`/repair`	Repair the existing installation. The default is to install.
`/interactive_mode [:on|:off]`	Display output from the installer after it finishes if set to `on`, close console window immediately if set to `off`. Default: `on`.
`/user[:all|:current]`	Install for all users or for the current user. Default is all users.

4.1.4 Troubleshooting

- If the installation fails with the following error message, it may be that some of the software required for the installation is not registered on your machine.

  Error 1720. There is a problem with this Windows Installer package. A script required for this install to complete could not be run. Contact your support personnel or package vendor.

  To provide us with information on your registry, you can download RegDllView from http://www.nirsoft.net/utils/registered_dll_view.html, run it and submit a request for support at http://www.schrodinger.com/supportcenter, and attach the output to the web form.

- If you see a warning that there is not enough disk, you will have to clean up your primary disk before proceeding, even if the disk that you are installing to has plenty of space. This is because Windows caches the installers to the primary disk before installing, so you must have enough space on your primary disk. You can remove the download zip file once you have extracted it, if you downloaded it to your local disk. You can also remove the download zip files and folders for previous releases to free some disk space.

- If you need more log information when you install the software, you can set the environment variable `SCHRODINGER_INSTALLER_LOG_VERBOSE` before running the installer. Verbose logging can significantly increase the time for installation. For information on setting environment variables, see Appendix A.

- If you are editing the `schrodinger.hosts` file and cannot save it, this usually means that the editor application is running without administrator privilege. Windows does not allow you to modify files under Program Files without administrator privilege. You can save the file in a location where you have write permission, then move it to the installation. When editing files in the installation, you should start your editor with administrator privilege, for example by right-clicking and choosing Run as Administrator.

- If Maestro fails to start and the DOS startup window shows the following error:

  ```
  The system cannot execute the specified program.
  ```

  the Visual C++ libraries are probably not installed. These libraries need to be installed by a user with administrator privileges, and are available in the software distribution.

If you have problems with the installation, please submit a request for support at http://www.schrodinger.com/supportcenter, and attach the following files, which are in the directory `%LOCALAPPDATA%\Schrodinger\Installer\2015`:

```
schrodingerInstaller.log
SchrodingerPreInstallReg.reg.txt
schrodingerSetup.log
SchrodingerPostInstallReg.reg.txt
```

You should also send the file `schrodinger_machid.txt`, which is on your desktop. If you have installed software versions for more than one release, there will be multiple copies of this file, named `schrodinger_machinfo-`*N*`.txt`, where *N* is a number. In this case you should check that you send the correct version of the file (which will usually be the latest version).

If Maestro fails to start, submit a request for support at http://www.schrodinger.com/support-center, describing the circumstances, and attach the file `maestro_error_`*pid*`.txt`. If Maestro fails after startup, attach this file and the file `maestro.exe_`*pid*`_`*timestamp*`.dmp`. These files can be found in the following folder:

```
%LOCALAPPDATA%\Schrodinger\appcrash
```

When submitting a request for support, please include the following information:

- All relevant user input and machine output
- Schrödinger software purchaser (company, research institution, or individual)
- Primary Schrödinger software user
- Computer platform type
- Operating system with version number
- Version numbers of products installed
- mmshare version number

4.2 Installing Prime Third-Party Software and Databases

Prime allows you to do both web-based searches of the BLAST and PDB databases and searches of a local copy of these databases. The required Prime third-party software is included in the Prime installer, and is installed into the `thirdparty` folder of the installation by default. The databases needed for web-based searches are also included in the Prime installer, and are installed by default. When you do a default installation of Prime, you need not do anything more to enable web-based searches (for finding homologs, secondary structure prediction, and alignments). If you have a web proxy server, you may have to configure it—see Appendix C.

If you want to use local databases, you must install the databases and set the appropriate environment variables.

The required Prime third-party databases can be installed directly from the web. They are also available from the Schrödinger FTP site as a set of zip files, or on DVDs. When you install the databases, you should consider installing them to a network-mounted disk, because of the space required and so that they can be used by multiple users. If you do so, you must set environment variables to identify these databases—see Table 2.3 on page 13. You should not install the databases into the Schrödinger software installation.

The databases from the FTP site or DVDs should be updated to obtain the most recent changes, using the instructions below.

Note: PSIPRED is not supported on Windows.

To update or download the BLAST and PDB databases:

1. Choose Start → All Programs → Schrodinger-2015-2 → Schrodinger Command Prompt.

   A DOS command prompt window opens, in which you can type the required commands.

2. Set the path to the third-party databases by typing the following command:

   `set SCHRODINGER_THIRDPARTY=`*thirdparty-location*

   If you installed the third-party databases in the default location, or if the `SCHRODINGER_THIRDPARTY` environment variable is already set, skip this step.

3. Run the `rsync_pdb` utility by typing the following command:

   ```
   rsync_pdb
   ```

4. Run the `update_BLASTDB` utility by typing the following command:

   ```
   update_BLASTDB -b
   ```

The download starts, and the progress of the process is printed in the shell window. When it is finished, you can close the command prompt window.

(Alternative) To download and install the PDB and BLAST databases (and software):

1. Go to the following site:

   ftp://ftp.schrodinger.com/support/hidden/prime/Windows_Suite2015/

   If you double-click on this link, it should open up a file explorer that lists all the files available at this location.

2. Download the following zip files:

   `psp-blast-thirdparty-database-`*M*`of`*N*`.zip`
   `psp-pdb-thirdparty-database-`*M*`of`*N*`.zip`

   The number of files may change, as we update the databases periodically. The download could take several hours, because each zip file is about 600MB. You only need the last zip file if you are planning to install the software outside the Schrödinger installation.

3. Unzip the zip files into the desired location.

   We recommend that you do not use the Schrödinger installation folder to install the databases, because they are very large, and you would have to move them for each release.

The extraction process could take an hour or two. When it finishes, you should have the databases in a `thirdparty` folder in the location you chose.

Some programs for unzipping files, such as 7-Zip, allow you to select multiple zip files to extract into a single folder.

4. Set the required environment variables—see Section 2.4 on page 11 and Appendix A.

   If you are only installing the databases, set `SCHRODINGER_PDB` and `PSP_BLASTDB`. If you are installing the software and the databases in the same location, set `SCHRODINGER_THIRDPARTY`.

To install the databases from DVD, you need only complete Step 3 and Step 4.

Note: The copy of the PDB database installed as described here does not include an `all` folder. While this does not affect the running of Schrödinger software, its absence might affect the running of other software that relies on the presence of this folder.

4.3 Setting Up Access to PyMOL

Both Maestro and Canvas can open PyMOL directly. To do so, PyMOL must be installed and the location communicated to the application. The recommended procedure is:

1. Install PyMOL in a location outside the Schrödinger software installation.
2. Add the PyMOL location to the `PATH` environment variable.
3. Set the `PYMOL4MAESTRO` environment variable to the PyMOL location.

See Appendix A for instructions on setting environment variables. Canvas requires that the PyMOL location is in the `PATH` environment variable. Maestro does not use `PATH`, but uses the following sources to find an installation of PyMOL first, in the order given:

1. The registry. If PyMOL is installed on Windows with an installer, this version is used.
2. The `PYMOL4MAESTRO` environment variable. This environment variable should point to the PyMOL installation. This is the preferred way of specifying the location of PyMOL.
3. A PyMOL launch script called `pymol4maestro` in the Schrödinger software installation. This script must run PyMOL.
4. A PyMOL installation in the Schrödinger software installation in the `pymol` folder.
5. The standard PyMOL environment variable `PYMOL_PATH`.
6. A launch script called `pymol` in the Schrödinger software installation.

4.4 Uninstalling Schrödinger Software

Before uninstalling Schrödinger software, ensure that all Schrödinger applications and supporting programs are not running: this includes utilities, Maestro, Schrodinger Command Prompt or SchrodingerShell windows, and any other Schrödinger applications. If remote jobs are running, you should wait until they finish before uninstalling Maestro, otherwise the results will not be copied back to the working directory or incorporated into the project.

If you want to kill all Schrödinger applications and supporting programs, navigate to the `mmshare-v`*version*`\bin\WIN32-x86` folder in your Schrödinger software installation and run `SchrodingerProcKill`.

You cannot use Add/Remove Programs to uninstall the software: this has been explicitly disabled, because `setup.exe` performs some tasks that are not performed with Add/Remove Programs. You must therefore use `setup.exe` or its equivalent, as explained below.

For releases prior to Schrödinger Suite 2009 Update 2, you *must* use the `setup.exe` program to uninstall the software. If you no longer have this program for the particular release, you can download it again from the Downloads page of the Schrödinger web site.

From Schrödinger Suite 2009 Update 2 on, you can use one of the following methods:

- Run the uninstaller by going to Start → All Programs → Schrodinger-*year* → Uninstall.
- Run the `uninstall.exe` program, which is located in the `installer` folder of your Schrödinger software installation (`%SCHRODINGER%\installer`).
- Run the `setup.exe` program that you used to install the software.
- Run `uninstall-silent.exe` from a DOS window (*not* a Schrodinger Command Prompt window). This program is also in the `installer` folder of your Schrödinger software installation. It accepts the same arguments as `setup-silent.exe`, given in Table 4.2 on page 29, except that the default and only allowed action is `/remove`, so you do not need to specify the action.

When you uninstall Schrödinger software you can choose whether to uninstall Maestro and the various applications. Selecting Maestro does not uninstall other applications, which must be selected separately. To uninstall the entire software package, you should ensure that you uninstall the other applications as well as Maestro.

If you have installed the third-party databases required by Prime into the Schrödinger installation, the uninstaller does not remove them, and you must remove them manually from the installation. You should do this before running the uninstaller, otherwise the installation is not entirely removed.

Installing the Products on a Mac

This chapter covers installing and uninstalling Schrödinger software on Mac OS X. Once the software is installed, you must obtain and install a license. If you want to run jobs on other hosts, you must also configure access to those hosts, using the information in Chapter 7.

5.1 Installing Schrödinger Software

To install Schrödinger software:

1. Click the disk image (`.dmg`) in your Downloads folder or insert the DVD.
2. Double-click the installer package (`.pkg`).

   The installer starts.
3. Follow the prompts in the installer.

   The installation path is `/opt/schrodinger/suites2015-2`. You can use a different disk by clicking Change Location.
4. Enter the required password to install the software.

   The software is installed, and a folder named SchrodingerSuites2015-2 is added to the Applications folder. This folder contains icons for various Schrödinger applications, which you can drag to the dock if you wish.

5.2 Installing Prime Third-Party Databases

Prime requires access to the PDB and BLAST databases. If these databases are not installed locally, you can use web access to send queries and retrieve the results. If you want to use a local installation instead (for security reasons, for example), use the following instructions.

1. Check that you have at least 20 GB disk space available.
2. Open a terminal window.
3. Set the `SCHRODINGER_THIRDPARTY` environment variable to a suitable location that is *not* inside your software distribution, e.g.

   ```
   export SCHRODINGER_THIRDPARTY=/opt/schrodinger/thirdparty
   ```

This choice ensures that the database is not installed inside your software installation, as it would be removed when you uninstalled the software.

Indexing these databases causes severe performance issues, so you should turn off indexing of this directory before starting the installation.

4. Open System Preferences and click on Spotlight.
5. In the Privacy tab, add the location that you chose for the third-party software above.

Now you can perform the installation.

6. In the terminal window, run the following commands:

```
$SCHRODINGER/utilities/update_BLASTDB
$SCHRODINGER/utilities/rsync_pdb
```

Due to the size of these databases, the commands may take some time to run.

If you want to use PSIPRED, you must compile it in a terminal window using the Linux instructions in Section 3.6 on page 20. To do so, you must have XCode installed, including the command-line tools (Preferences → Downloads → Command-Line Tools in XCode, and click Install). The command-line tools require an Apple Developer account.

5.3 Uninstalling Schrödinger Software

Schrödinger software comes with an application for uninstalling the software. This application performs all the necessary actions to completely remove the software from your Mac. You should not use any other method for uninstallation.

To uninstall Schrödinger software:

1. Double-click the Uninstall.command application.

   This application is in the SchrödingerSuite2015-2 folder in the Applications folder. You can also use the copy of this application in the original `.dmg` package that you used to install the software.

   You are prompted to confirm the uninstallation.

2. Type `yes` and press Enter.

   A password prompt is displayed.

3. Enter the administrator password.

   The software is removed from disk and the SchrodingerSuites2015-2 folder is removed from the Applications folder.

4. Close the command window for the uninstaller.

If you do not use the uninstaller, and try to install software, the applications will not run. You must then use the uninstaller to uninstall the software, and then reinstall it.

Installing and Updating KNIME Extensions

The Schrödinger KNIME Extensions are available with the software distribution, either on DVD or the software download page, or from the Schrödinger KNIME update web site. The software distribution includes KNIME itself as well as the extensions.

- To install KNIME and the Schrödinger extensions from the software distribution, select the KNIME Extensions when you install the Schrödinger software (Windows only). KNIME and the extensions are automatically installed on Linux and Mac.
- If you already have a compatible version of KNIME installed and want to use this version of KNIME instead, you should install from the Schrödinger KNIME update web site, using the instructions in Section 6.1. An archived copy of this site is included in your Schrödinger software download. The update site should be in the folder you extracted the download into, with the name `SchrodingerKNIMEUpdateSite_`*version*`.zip`.
- If you want to update the Schrödinger KNIME Extensions, use the update site or a downloaded archive of this site (see the previous paragraph). See Section 6.2.

6.1 Installing into an Existing Version of KNIME

If you want to install the Schrödinger extensions into an existing version of KNIME, you can do so from the KNIME interface. If the version of KNIME does not meet the minimum requirements, you must update your version of KNIME first. The installation requires access to the Schrödinger KNIME update web site or to a local archived copy of this site.

You must also install the KNIME Base Chemistry Types and Nodes from the KNIME update site before you install the Schrödinger extensions, if you have not done so already.

To use the Schrödinger KNIME extensions, you must identify the Schrödinger installation to KNIME. The nodes will not work without access to a Schrödinger installation.

The installation thus has three stages, instructions for which are given below:

- Installing and updating KNIME itself.
- Installing the Schrödinger nodes
- Identifying the Schrödinger software installation to KNIME.

To install the KNIME Base Chemistry Types and Nodes:

1. Start KNIME.
2. Choose Help → Install New Software.

   The Install Available Software panel opens. If you already have the Base Chemistry Types and Nodes installed, you can skip to Step 2. If you are not sure, proceed with the next step to check.
3. From the Work with option menu, choose the KNIME Update Site.

   The URL for this site should begin with http://www.knime.org. There may be a delay while the list of extensions is loaded from the web site, and the table shows "Pending" in the Name column next to the check box. When the extensions are loaded, they are displayed in the table, with a tree structure so you can show and hide the individual extensions.

   To check whether you have the required extensions installed, ensure that Hide items that are already installed is selected, then click the turner (+ button) for the KNIME & Extensions and KNIME Labs Extensions items to display the list of extensions.
4. Ensure that KNIME & Extensions and KNIME Labs Extensions are checked.

   Checking these items selects all the extensions for these items, which you can verify by expanding the tree. These extensions include more than the Base Chemistry Types and Nodes, but it can be useful to install them all.
5. Click Next.

   The Install Details screen is displayed, listing the extensions you selected.
6. Click Next.

   License agreements are displayed.
7. Accept the license agreements and click Next.

   The extensions are installed. This may take some time. When the installation finishes, a dialog box informs you that you must restart KNIME for the installation to finish, and offers several.
8. Click Restart KNIME Now.

   KNIME restarts, and you can proceed to the installation of the Schrödinger extensions.

To install the Schrödinger extensions:

1. Start KNIME, if it is not already running.
2. Choose Help → Install New Software.

   The Install Available Software panel opens.
3. In the Available Software tab, click Add.

   The Add Repository dialog box opens.
4. Specify the location of the update site:

   If you are using the web site directly:

   a. Enter the following URL in the Location text box:

      https://support.schrodinger.com/releases/knime/extensions

      You can also enter a name to identify the site.

   b. Click OK in the Add Repository dialog box.

   c. Enter your user name (email address) and password for the Schrödinger web site.

   If you downloaded the archived site:

   a. Click Archive.

   b. Navigate to and select the downloaded zip file.

   c. Click OK in the file selector.

   d. Click OK in the Add Repository dialog box.

   The available software from this site is listed in the center of the panel.
5. Select the newly created tree for the update site.
6. Ensure that Show only the latest versions of available software and Hide items that have already been installed are both selected.
7. Select the Schrödinger nodes you want to install, by clicking the check boxes.

   The Schrodinger Nodes for Knime core extensions feature must be selected to use any Schrödinger node or renderer. The 2D renderers are optional; the renderer core feature is required to use any of the renderers (Maestro, SD and SMILES).
8. Click Next.

   License agreements are displayed.

9. Accept the license agreements and click Next.

   The installation begins. Because the Schrödinger plugins are signed, you will be asked to accept this information.

10. Click Restart KNIME Now.

    KNIME restarts, and you can proceed to the identification of the Schrödinger software installation to KNIME.

To identify the Schrödinger software installation to KNIME:

1. Start KNIME, if it is not already running.
2. Choose File → Preferences.

   The Preferences panel opens.

3. Under KNIME, choose Schrodinger.

   The Schrodinger preferences are displayed. There are two environment variables that you can set: `SCHRODINGER`, and `PYMOL4MAESTRO`. If either of the fields for these environment variables is unavailable (grayed out), then KNIME has picked up the variable from the environment, and you do not need to set it. There should be a message telling you that KNIME has done this. If you want to set it in the KNIME preferences, you should unset it in your environment and then start KNIME and set it.

4. Enter the path to the Schrödinger installation in the SCHRODINGER text box.

   The default paths are:

   - **Windows:** `C:\Program Files\Schrodinger-2015-2` (or, for 32-bit software on a 64-bit OS, `C:\Program Files(x86)\Schrodinger-2015-2`)
   - **Mac:** `/opt/schrodinger/suites2015-2`

5. If you want to start PyMol from KNIME, enter the path to the PyMol installation in the PYMOL4MAESTRO text box.
6. Restart KNIME to start using the Schrödinger nodes.

6.2 Updating Existing Schrödinger Extensions

If you have direct access to the web, you can update the existing Schrödinger KNIME extensions to new versions using the update mechanism in KNIME, as follows.

1. Start KNIME.
2. Choose Help → Check for Updates.

   The Contacting Software Sites panel opens, and checks for updates to the software, by contacting the web sites defined in your Available Software Sites preferences (see File → Preferences → Install/Update). You are prompted to log on to the Schrödinger web site.
3. Enter your email address and password (as you normally do to log on to the Schrödinger web site).

   If there are updates, the Available Updates window opens, listing the updates that are available. This list includes updates to all installed KNIME components.
4. Review the list of updates, and deselect any items that you do not want to update.

   If you only want to update the Schrödinger extensions, deselect all other updates.

   Note: If you update the KNIME Desktop or Eclipse, we do not guarantee that the Schrödinger extensions will continue to work.
5. Click Next.

   The detailed list of items to be upgraded is shown, including any dependencies.
6. Click Next.

   License agreements are displayed.
7. Accept the license agreement, and click Finish.

   The download begins. As the Schrödinger plugins are signed, you will be asked to accept this information.
8. Restart KNIME when prompted.

If you do not have direct access to the web, you can update existing extensions from a downloaded archive of the update site by following the procedure for adding extensions.

6.3 Adding New Schrödinger Extensions

Adding new Schrödinger extensions uses a very similar mechanism to installing the Schrödinger extensions into an existing KNIME installation. You can do this from the Schrödinger KNIME update web site or from a downloaded archive of this web site.

1. Start KNIME.
2. Choose Help → Install New Software.

   The Install Available Software panel opens.
3. Ensure that Show only the latest versions of available software is selected.
4. If you are installing the new extensions from the web site, do the following:

   a. Open the Schrödinger site.

      The URL should be https://support.schrodinger.com/releases/knime/extensions. This URL is automatically added when you install KNIME as part of the Schrödinger software.

      You are prompted to log on to the Schrödinger web site.

   b. Enter your email address and password (as you normally do to log on to the Schrödinger web site).

   Otherwise, if you are installing the new extensions from a downloaded archive of the site, do the following:

   a. In the Available Software tab, click Add.

      The Add Repository dialog box opens.

   b. Click Archive.

   c. Navigate to and select the downloaded zip file.

   d. Click OK in the file selector.

   e. Click OK in the Add Repository dialog box.

   The available extensions should now be displayed under the URL.
5. Select the Schrödinger nodes you want to install.
6. Click Next.

   License agreements are displayed.

7. Accept the license agreement, and click Next.

   The installation begins. As the Schrödinger plugins are signed, you will be asked to accept this information.

8. Restart KNIME when prompted.

You can also use this panel to update or install nodes from sources other than Schrödinger.

6.4 Installing Knime Workflows

KNIME workflows that have been wrapped to run from Maestro can be installed for the use of all users into the installation. The standard location for these workflows is in the `mmshare-v`*version*`/python/workflows` directory of the installation. The installation can be performed with one of the following commands:

```
$SCHRODINGER/run installscripts.py -k -c -z workflowzipfile.zip
$SCHRODINGER/run installscripts.py -k -c -d workflowsourcedir
```

The zip file *workflowzipfile*`.zip` or the directory *workflowsourcedir* should contain a set of Python scripts named *workflow*`.py` and directories named *workflow*`_dir`.

When Maestro is started, these workflows should appear on the Workflows menu.

If you do not want to install the workflows into the installation, you can simply place them in a directory and set the `MAESTRO_WORKFLOW_LOCATION` environment variable to point to this directory.

If you are using an external KNIME installation rather than the one in the Schrödinger installation, you must make sure that the external installation has the Schrödinger nodes installed and everything under KNIME & Extensions and KNIME Labs Extensions for the current version of KNIME.

In addition, you must add a `knime` setting in the hosts file for the hosts on which you plan to run Maestro, or to the `localhost` entry. See Section 7.1 on page 47 for information on the hosts file. This setting will have to be updated if you update the external KNIME and the path to the KNIME startup script (or executable) changes. Note that this setting is applied if you try to run the Schrödinger KNIME instance via the icon or the startup script: the external KNIME will start instead of the Schrödinger KNIME.

6.5 Installing A Web Service for Node Execution

Some KNIME nodes can be run through a web service. The instructions for setting up the web service and the necessary configuration are given below.

To set up the web server:

1. Install the Apache Tomcat web server on the machine where the Schrödinger software is installed. If you already have the web server on this machine, stop the web server.
2. Copy the `SchrodingerExecCommand.war` file from your Schrödinger software installation into the `webapps` folder of your Apache Tomcat server location. This file is in the `knime-v`*version*`/webservices` folder (directory) of your Schrödinger software installation.
3. On Linux, set the `SCHRODINGER` and `SCHRODINGER_TMPDIR` environment variables,
4. Start the Apache Tomcat server.

There are two ways to set up the client. You can use Schrödinger Job Control to send the job to the server from your node, or you can use a generic web service client node. To set up the client using Job Control, you must add an entry to the hosts file, `schrodinger.hosts`, with the following settings:

`name:` *server-alias*
`host:` *server-host-name*
`port:` *server-port*
`processors:` *maximum-server-cores*

See Section 7.1 on page 47 for more information on the hosts file and its entries.

Instructions for configuring the nodes are given in Section 4.2 of the *KNIME Extensions User Manual*.

Preparing for Job Submission

Schrödinger products use a common Job Control facility, which allows the user to submit, monitor, suspend and terminate jobs. The Job Control facility is described in detail in the *Job Control Guide*. To run jobs on the local host only, no additional configuration is needed, but you should test your installation as described in Section 7.10 on page 92.

To run jobs on remote hosts, run distributed jobs, run jobs on remote hosts, or submit jobs to batch queues you must first set up the *hosts file*, `schrodinger.hosts`. This task is described in Section 7.1. Next, you must enable access to these remote hosts using the `ssh` command (or equivalent) without specifying a password (see Section 7.2). For batch queues, additional configuration is needed (see Section 7.3), and for some queueing systems, it is also possible to set up license checking (Section 7.9).Finally, this chapter provides information on configuring clusters (Section 7.9) and checking the installation and connections (Section 7.10).

Note: The remote host must be a Linux host. Submitting jobs to a remote Windows or Mac host is not supported.

7.1 The Hosts File

The Job Control facility obtains information about the hosts on which it will run jobs from the *hosts file*. The default name for this file is `schrodinger.hosts`. Maestro also uses the hosts file to set up the menus in the Start dialog box.

This file must contain information on all hosts to which jobs can be submitted, and a copy of this file must be accessible on all hosts from which jobs will be submitted. The copy of the file that is stored in the installation directory provides the default settings for all users and all hosts. To customize job submission for a user, copy `schrodinger.hosts` to the directory `$HOME/.schrodinger` on UNIX, or `%USERPROFILE%\Schrodinger` on Windows, and edit it. A user copy is necessary if the user has a different user name on any host on which Schrödinger products will be run. You can also copy it to a location of your choice and set the `SCHRODINGER_HOSTS` environment variable to point to this file.

The hosts file consists of one or more *entries*, each of which describes a configuration for running jobs on a given host. Each entry consists of a number of settings, one per line. Default values for all hosts are taken from the `localhost` entry, which defines the settings for the current host. Settings are described in more detail in the following subsections.

Note: If you are editing this file in your installation directory on Windows, you should start the editor with administrator privilege, for example by right-clicking and choosing Run as Administrator. Otherwise you will not be allowed to save the file.

To add entries to the hosts file:

1. Open the `schrodinger.hosts` file in a text editor.

   When the software is installed, there is a copy of this file in the installation directory.

2. Create or modify an entry for each remote host using the keywords in Table 7.1.
   - The syntax for the settings is *keyword*: *value*.
   - Keywords are case-insensitive.
   - Each entry must begin with a `name` setting.
   - Comments can be included by beginning a line with a # sign.
   - Multiple entries can be included for a given host with different settings (e.g. different scratch directories).
   - Entries for batch queue submission (including queues on clusters) must specify a temporary directory that is available on all compute nodes, and is writable by all valid users. This directory can be on a common file system shared by the nodes, or it can be identically-named local storage space on each node.
   - Individual nodes in a cluster must be included unless they are only used as part of a properly-configured batch system.
   - If the host has a batch queueing system, you can add settings for the batch queue—see Section 7.3 on page 63 for more information.

3. Save and close the file.

You can test the connections to the hosts in the `schrodinger.hosts` file by using the Diagnostics panel, described in Section 7.10.1 on page 92, or the `installation_check` command, described in Section 7.10 on page 92.

Table 7.1. Keywords for schrodinger.hosts file settings.

Keyword	Description
`base`	Name of an entry (the *base* entry) that is the basis for the current entry. All the keywords from the base entry are inherited by the current entry, and new keywords may be added, in any order. A `base` entry can include another `base` entry.
`env`	Environment variables to be set on the host. The syntax for the environment variables is *variable*=*value*, regardless of the shell used. List each environment variable on a separate `env` line.

Table 7.1. Keywords for schrodinger.hosts file settings. (Continued)

Keyword	Description
gpgpu	Specify a graphics processor (GPU) to use on the host. Multiple instances can be used for multiple GPUs. The specification is in the form *id*, *description*, where *id* is the numerical GPU id, usually starting from 0, and *description* is the description of the GPU, for example GTX-580.
host	Host name. This entry is only needed if it is different from the name setting or if the queueing software is only available on a particular host.
include	Name of an auxiliary hosts file to be included in the current hosts file. The inclusion is done by replacing the include line with the contents of the specified file.
knime	Path to an external KNIME installation (i.e. an installation other than the one in the Schrödinger installation).
name	Name of the host entry or batch queue. For a host this is usually the host name. This name is displayed in the Start dialog box. The name must not contain spaces. The value localhost is a special name that means the host on which the job is launched.
nodelist	List of entry names, used to define a multiple-host entry. A name may be followed by a colon and a number of processors. Can be combined with a host setting.
parallel	Specify whether the host supports MPI parallel jobs or not. The value can be specified as yes or no, true or false, 1 or 0.
port	Server port to use when sending jobs to a server (Used by KNIME only).
processors	Number of processors available on the host. If the host is part of a cluster, this number should be the total number of processors available on the cluster. For multicore processors, the number should be the total number of cores available. The default is 1, except for the localhost entry, where the default is the number of available processors (or cores).
processors_per_node	Number of processors (cores) per node available to a batch queue. This setting is used by applications that support threaded parallel execution (OpenMP).
proxyhost	Host on which to run jproxy. This setting should be made when the host from which a job is launched cannot open a socket connection on the host on which the job is actually run. By default, jproxy is run on the host specified by the host keyword, and is only run when using a queuing system. This setting is only needed in cases where using the default is impossible or impractical. Only valid when the host entry also contains a queue setting.

Table 7.1. Keywords for schrodinger.hosts file settings. (Continued)

Keyword	Description
proxyport	Specify the port or range of ports that jproxy may use. Ports can be specified as comma or colon-separated lists without spaces. Ranges can specified with a dash, for example, 5987:5989-5992:5994. Only valid when the host entry also contains a queue setting.
qargs	Arguments to be used when submitting jobs to a batch queue. These arguments should specify any parameters that define the queue.
queue	Queuing system name. PBS, SGE, LSF, and SLURM are the supported systems. Must be set to the subdirectory of $SCHRODINGER/queues that contains the support files for the queuing system.
recoverjobs	Disable recovery of failed jobs if set to no. Use this setting only for jobs where job recovery might not be possible (such as on the cloud).
schrodinger	The path to the Schrödinger software installation on the host.
tmpdir	Base directory for temporary or scratch files, also called the scratch directory. The file system on which this directory is mounted should be large enough for the largest temporary files, should be mounted locally, and should be writable by the user. Do not use symbolic links, as these can cause some programs to fail. The actual directory created for scratch files is */tmpdir/username/jobname*, where *tmpdir* is the directory defined here and *username* is the user name. Multiple tmpdir settings can be added for a given host and are used by Maestro, but the first setting is used otherwise.
user	User name to use on the host. This should never be set in the hosts file in the installation directory. It is required if the user has a different user name on the defined host than on the host on which the job is launched.

A sample schrodinger.hosts file is shown below.

```
# Schrodinger hosts file
#
name:         localhost
tmpdir:       /scr
#
name:        larry
name:        curly
name:        moe
#
name:        server
schrodinger: /usr/local/schrodinger2015
tmpdir:      /big_scr
```

```
processors:  8
#
name:        cluster
host:        manager
queue:       PBS
qargs:       -lwalltime=1000:00:00
schrodinger: /sw/schrodinger2015
env:         SCHRODINGER_THIRDPARTY=/fast/disk
processors:  16
tmpdir:      /storage/TMPDIR
#
# End of Schrodinger hosts file
```

7.1.1 The name and host Settings

The `name` setting must be the first line for each entry. This is the name that is used to select the host (or batch queue) with the configuration specified in the following settings. It is displayed in the list of hosts in the Start dialog box Host menu or table. Usually, *entry-label* is the name of a host that can be used to run a calculation. If it is not, you must include a `host` setting that supplies the host name. The `host` setting is only needed if the name line does not give the host address. You might, for example, want to provide an alias in the `name` setting and define the host name in a `host` setting if the host name is long. Another use of multiple entries for a single host is to specify different settings on a host, such as different scratch directories or different software installations. You can also use the `name` and `host` settings to specify a batch queue name and the host on which the batch system is available. Yet another use of the `name` setting is to specify a base entry, that defines defaults for other entries. The `name` setting must not contain spaces.

The host name does not need to be the fully qualified domain name: it can be any name that can be resolved by the domain name server (DNS). You will probably need the full name if the host on which you plan to run (the *execution* host) is not on the same local network as the host from which you plan to submit jobs (the *submission* or *launch* host). For a hosts file on Windows, you must use the same name for the host that you used when setting up remote access—see Section 7.2.2.

The value `localhost` is a special `name` setting that means the host from which the job was submitted. In addition to this function, the settings for the `localhost` entry are used as the default values for all other entries. In the `schrodinger.hosts` file example above, the host entries `ahost` and `bhost` inherit the `tmpdir` setting from the `localhost` entry. The `localhost` entry may not have a `schrodinger` setting, however, as the local software installation is defined by the one that you are using and cannot be overridden.

If you run jobs from the command line, the `name` setting is what you should use with the `-HOST` option to select the hosts to run the job.

7.1.2 The user Setting

If you have different user names on the submission and execution hosts, you can include a `user` setting for the execution host in the hosts file on the submission host. The `user` setting should never be added to entries in the hosts file in the installation directory, because this would prevent other users from using those entries. If a `user` setting is required, the user should copy the hosts file to `$HOME/.schrodinger` (Linux or Mac) or `%USERPROFILE%\Schrodinger` (Windows) on the job submission hosts and add the `user` settings to this copy.

7.1.3 The tmpdir Setting

The `tmpdir` setting specifies the scratch directory, where temporary files can be written. Examples are `/scr` or `/temp`. The file system on which this directory is mounted should be large enough for the largest temporary files, should be mounted locally, and should be writable by the user. The actual directory created for scratch files (the job directory) is *tmpdir*`/`*username*`/`*uniquename*, where *tmpdir* is the directory defined here and *username* is the user name.

If you do not specify `tmpdir` for a host, the `tmpdir` setting from the `localhost` entry is used, if there is one. Otherwise, the scratch directory is set to `$HOME/.schrodinger/tmp` on Linux and Mac OS X, and to `%LOCALAPPDATA%\Schrodinger\tmp` on Windows. The use of the home file system for large temporary files is discouraged in most places, so you should always ensure that `tmpdir` is defined for the hosts you run jobs on, if the job requires temporary storage.

You can override the `tmpdir` setting in the `schrodinger.hosts` file by setting the `SCHRODINGER_TMPDIR` environment variable or using the `-TMPDIR` command-line option—see Section 2.3 of the *Job Control Guide* for more information. For example, if the directory designated by `tmpdir` becomes full with files that you don't have permission to delete, you can set `SCHRODINGER_TMPDIR` to a different directory and continue to run jobs.

7.1.4 The processors and processors_per_node Settings

For hosts with multiple processors that are not running a queueing system, set `processors` to the number of processors on the host. For batch queues, set `processors` to the number of processors available to the queue.

Here, "processors" refers to hardware units capable of running a job, which for most recent hardware means "cores".

For applications (such as Jaguar) that use OpenMP for threaded parallel execution, set `processors_per_node` to the number of processors available on each node available to a batch queue. You should ensure that the host for the queue does not have nodes with different numbers of processors.

Within the `qargs` setting, you can use `%PPN%` as a variable that has the value of the `processors_per_node` setting, and `%NPROC/PPN%` for the number of nodes (total number of processors divided by the number of processors per node).

7.1.5 The parallel Setting

The `parallel` setting allows you to designate hosts that do or do not support MPI parallel job execution. By default, it is assumed that a host supports MPI parallel jobs: thus, the main need for this setting is to designate hosts that do not support MPI parallel jobs. The setting allows applications to detect a host selection that does not support MPI parallel jobs, and post a warning before the job is submitted to the host (or queue), thereby avoiding job failure. The setting is merely a flag: there is no checking of the actual host to verify its support of MPI.

7.1.6 The gpgpu Setting

To make use of GPU cards for Desmond calculations, the hosts file must be configured for each host. For example, on a local workstation, you could use an entry like this:

```
# Local workstation
name: myhostname
host: localhost
processors: 4
tmpdir /usr/tmp
gpgpu: 0, GTX 680
```

This entry indicates that the local workstation has a single GeForce GTX 680 card available at device index 0. For a remote host you could use an entry like this:

```
# Remote host
name: otherhost1
host: remote1.mycorp.com
processors: 8
schrodinger: /path/to/schrodinger/
tmpdir: /usr/tmp
gpgpu: 0, Tesla K20
gpgpu: 1, Tesla K20
```

Here, the remote workstation has two Tesla K20 cards, residing at device indices 0 and 1.

7.1.7 The schrodinger Setting

The `schrodinger` setting specifies the directory in which your Schrödinger software is installed on this host (the *installation* directory). If this setting is present, it is used by Job Control as the location of the software installation on the remote machine. If it is not present, the path defined by the software installation that you are using on the local host is used as the path to the software installation on the remote host.

A `schrodinger` setting may not be added to the `localhost` entry, as the location of the software on the local host for submitting or running a job is determined by the software that you are actually using to submit or run the job (set either implicitly or explicitly in your environment).

If you want to submit jobs from a Windows host to a Linux host, you *must* add `schrodinger` settings to ensure that the software is located. Because the default is to use the path to the local installation, the Windows installation path will never be located on Linux.

Note: The `schrodinger` settings in the hosts file should always point to an installation for the current release.

For example, suppose the software is installed on an NFS installation on a Linux host at `/nfs/schrodinger2015-2`. It is cross mounted on a user's local workstation, a remote workstation and a remote cluster. The hosts file could look like the following:

```
name: localhost
tmpdir: /scr

name: bob
host: bob.company.com
processors: 8

name: cluster
host: headnode.company.com
queue: SGE
qargs: -q big.q
processors: 256
```

Here `schrodinger` is not set in any of the host entries. If `SCHRODINGER` is set to `/nfs/schrodinger2015-2` in the environment on the user's local workstation, jobs submitted to either of the remote hosts will look for the installation at `/nfs/schrodinger2015-2`. As this location is mounted on each host, the jobs will run.

Now suppose the software is installed on a Windows laptop, to the default location (`C:\Program Files\Schrodinger-2015-2`). On the laptop, the equivalent hosts file would look like:

```
name: localhost
tmpdir: /scr

name: bob
host: bob.company.com
schrodinger: /nfs/schrodinger2014
processors: 8
name: cluster
host: headnode.company.com
schrodinger: /nfs/schrodinger2014
queue: SGE
qargs: -q big.q
processors: 256
```

Here, `schrodinger` must be set for the remote hosts because on Windows, the default location is a Windows path, which by definition cannot exist on the remote Linux machines.

7.1.8 The env Setting

The `env` setting specifies an environment variable that is to be set on this host when any job is started. The syntax of the setting is *variable=value* (regardless of the UNIX shell used), where *variable* is the environment variable and *value* is its value. For example,

```
env: SCHRODINGER_THIRDPARTY=/software/databases
```

To set multiple environment variables, include one `env` setting for each variable. Environment variables set in the hosts file take precedence over any that are set in your UNIX shell, either on the local host or the remote host.

For products that use OpenMP multithreaded execution, you should set the environment variable `OMP_NUM_THREADS` to the number of processors or cores on a host that you want to use for multithreaded execution.

For Desmond, it is useful to set the environment variable `OMPI_MCA_btl` to `self,sm,tcp` if the host does not have an Infiniband network and is used for parallel execution. This setting suppresses warnings that are generated by Desmond, which uses Infiniband if it can when running in parallel.

For Open MPI parallel use, it is advisable to set one of the environment variables `TMPDIR`, `TEMP`, or `TMP` to a local file system, in order to avoid performance issues with temporary files. See Section 7.3.3 on page 65 for more information.

7.1.9 The include and base Settings

The include and base settings offer tools for maintaining a hosts file that can be used across multiple installations of the software.

The include setting permits you to include a hosts file in another hosts file. This allows you, for example, to make release-independent settings in one hosts file, and use those settings in a hosts file that contains settings for a particular release. The syntax is

include *filename*

where the file name must be an absolute path.

The base setting can be used to include settings from another entry that are used as defaults, and can be overridden by settings in the current entry. This allows you to specify settings that are common to a number of entries, without having to duplicate them for each entry. The syntax is

base *entry-name*

The examples below illustrate the use of the include and base settings. Suppose you define a hosts file named cluster_queues.hosts, that contains settings for several queues on a cluster. These settings use a base to pick up any common settings for the cluster, such as the location of the software installation and the queue host. The file might look like this:

```
# Small queue on cluster1
name:         small1
base:         cluster1
processors:   16

# Medium queue on cluster1
name:         medium1
base:         cluster1
processors:   64

# Large queue on cluster1
name:         large1
base:         cluster1
processors:   256
```

Note that the base setting refers to an entry that is not in this file. Now you define your main hosts file, schrodinger.hosts, with the settings that are common to the queues and also the software installation location, which will vary from release to release. This file has the base definition for the queues, and the cluster queue definitions are imported from the above file:

```
# Local host definition
name:         localhost

# Base settings for cluster1
name:         cluster1
host:         cluster1.mynetwork.com
schrodinger:  /cluster1/schrodinger/schrodinger2014
queue:        SGE
tmpdir:       /var/tmp

include /opt/schrodinger/cluster_queues.hosts
```

When this hosts file is read, `cluster_queues.hosts` is imported and the `base` settings are resolved. Note that `cluster_queues.hosts` is stored in a release-independent location. The hosts file itself is stored in the usual location in the software installation, on your workstation or wherever you plan to submit jobs from. This approach makes it easy to upgrade to a new software version, as you only have to edit the main hosts file that contains the base settings, change the path to the software installation, and save it in the new installation directory.

Entries used as a base can also refer to another base. So, for example, you could have entries that just defined the installation directory in your main `schrodinger.hosts` file, and use them as a base for everything else, which would be imported by an `include` setting:

```
# Local host entry
name:         localhost

# NFS-mounted installation
name: nfs-installation
schrodinger: /nfs/schrodinger2014-4

# Cluster installation
name: cluster-installation
schrodinger: /software/schrodinger2014-4

include /opt/schrodinger/cluster.hosts
include /opt/schrodinger/workstations.hosts
```

The included files could contain both the base settings for the cluster and the settings for the queues, like this (for `cluster.hosts`):

```
# Base setting for cluster1
name:         cluster1
base:         cluster-installation
host:         cluster1.mynetwork.com
```

```
queue:        SGE
tmpdir:       /var/tmp

# Small queue on cluster1
name:         small1
base:         cluster1
processors:   16
...
```

The base entry for the cluster also has a base setting, for the release-dependent `schrodinger` setting in the main hosts file. When you install a new release, you only need to copy the main hosts file to the new release directory and edit the `schrodinger` settings.

7.1.10 Customizing the Hosts File

You can copy and edit the `schrodinger.hosts` file from the installation directory to customize its settings. The edited hosts file should be placed in your Schrödinger user resources directory, which is `$HOME/.schrodinger` on Linux and Mac, and `%USERPROFILE%\Schrodinger` on Windows. You usually do not need to do this unless you have different user names on different hosts. If you have installed Schrödinger products on multiple hosts, you may need to edit the `schrodinger.hosts` file on each host to add entries for the other hosts.

7.2 Setting Up Access to Remote Hosts

To run jobs on remote hosts, you must set up access to these hosts by allowing use of the `ssh` command without specifying a password. If you have a firewall, you must allow access across the firewall. Setting up access is described in the subsections below.

Remote jobs can be submitted to Linux and Mac hosts, but not to Windows hosts.

- To be able to run jobs on remote hosts, Schrödinger products must be installed on both the local (job launch) host and the remote hosts, or on a file system that is accessible to both.
- If access has been set up previously, you do not need to do so again.
- If you plan to run parallel or distributed jobs that involve more than one host, you must set up passwordless access between these hosts. The special case of clusters is discussed in Section 7.9 on page 90.

The domain name server (DNS) must be enabled to run remote jobs from Windows.

7.2.1 Setting Up Access To and From Linux and Mac Hosts

To use passwordless SSH, the hosts to which you want to connect must be configured to satisfy the following requirements:

- An `sshd` server must be running.
- RSA public key authentication must be enabled and empty passphrases must be allowed in the `sshd` configuration.

  Note: Public key authentication is enabled in OpenSSH by default.

The following steps allow you to use `ssh` between computers that share your login directory without specifying a password.

1. Open a terminal window.
2. Generate a public/private RSA key pair on a host of your choice, whose home directory is shared with the remote hosts that you want to run jobs on:

   ```
   cd ~/.ssh
   ssh-keygen -t rsa
   ```

   Note: When asked for a passphrase *do not* enter one; just press ENTER. If you specify a passphrase it defeats the purpose of configuring passwordless `ssh`.

3. Add your public key to the list of keys allowed to log in to your account:

   ```
   cat id_rsa.pub >> authorized_keys
   cat id_rsa.pub >> authorized_keys2
   ```

   The two separate files are necessary to support both OpenSSH 1.5 and OpenSSH 2.0 protocols. Some versions use just one or the other of these files.

4. Remove your `known_hosts` file:

   ```
   rm known_hosts*
   ```

   This is necessary so that the new RSA key-pair mechanism is used for every host. Otherwise, hosts to which you previously connected using passwords might not use the new system automatically.

5. Make sure your home directory cannot be written by anyone but you:

   ```
   chmod go-w ~
   ```

   This is required before SSH will allow passwordless access to your account.

For each home directory that is *not* shared with that of the chosen host, run the following command:

`ssh-copy-id` *user@remote-host*

This command copies the keys and performs the necessary setup and permissions changes. If for some reason this does not work, you can do the setup manually as follows:

1. Copy the public and private keys to that home directory:

   `scp ~/.ssh/id_rsa*` *other-host*`:~/.ssh/`

2. Connect to the host on which that home directory is mounted and change to the `.ssh` directory:

   `cd ~/.ssh`

3. Repeat Step 3 through Step 5 above.

4. Ensure that `id_rsa` (the private key) is readable and writable only by the user:

   `chmod go-rwx ~/.ssh/id_rsa`

To test that passwordless SSH has been set up properly, you can execute a command on a remote host for which it is set up as follows:

`ssh` *username@remotehost* `date`

You should see a single line of output reporting the date on the remote host. If you are prompted for a password, the setup has not been done correctly. You should review the steps above to ensure that you have completed them all, especially the permissions.

7.2.2 Setting Up Access from a Windows Host to Linux Hosts

On Windows, passwordless SSH is handled using `plink.exe`, which is provided in the Schrödinger software distribution. To set up passwordless SSH, you can use the Remote Login Configuration tool to guide you through the process, which involves generating and saving keys, then configuring your remote hosts. This application is available from the Start menu, under Schrodinger-2015-2. You can also run the setup process manually—see the instructions in Appendix E.

Before you start, you should ensure that your hosts file, `schrodinger.hosts`, contains settings for all the hosts that you want to use. This file is used in the second stage of the setup. The `schrodinger.hosts` file used is the first one found in the following locations:

- The startup directory of the configuration tool.
- `%USERPROFILE%\Schrodinger`.
- The installation (`%SCHRODINGER%`).

If you want to use a `schrodinger.hosts` file from a particular directory, right-click on the shortcut for the configuration tool, and change the Start in directory.

The first panel is labeled Remote Access Setup. A summary of the steps is given in the panel; these steps are given in detail below.

1. Click Generate Keys.

   The default key generation is for SSH-2 RSA protocol and 1024 bit encryption. You should not need to change these choices, but if you do need to change them, click Options. The keys are written to `%USERPROFILE%`. The private key is stored as *user*`.ppk` and the public key is stored as `publickey.txt`.

2. Click Initialize Host Access.

   A second panel opens. In this panel, the available hosts are listed in two tables. In the first, you can edit the table cells to enter a host name and a user name. The second table contains a list of all hosts in the hosts file in the installation, `schrodinger.hosts`.

3. Select a host from the second list.

   A host does not need to be initialized if (a) it has a home drive that is shared with the host on which you are running the configuration tool, or (b) it has already been initialized.

4. Click Initialize.

   You will be prompted to enter your password for the host (unless it is already initialized). A connection is opened to this host, and the public key is appended to the authorized keys file on the host. Progress and results of the connection are displayed in the Status area.

   A dialog box opens, asking if you want to add this host to the list of known hosts.

5. Click Yes.

6. Repeat Step 3 and Step 5 for each host that you want to initialize access to.

You can only initialize access to one host at a time, so you must select each host and initialize access to it. If several hosts share a common home file system, you only need to initialize access to one of those hosts, but you should test all of them. Likewise, you only need to initialize access to a queue host for one of the hosts file entries for the queues on that host.

- To test access to a host, select a host in either table and click Test.

  Results are shown in the Status area.

If you want to initialize access to a host that is not in the hosts file (for example, before adding it to the hosts file) you can add it to the User Specified hosts table, select it and click Initialize.

7.2.3 Configuring the Firewall

If you have a firewall, some configuration needs to be made in order to run remote jobs. If you do not have a firewall, this section is not applicable.

7.2.3.1 Linux and Mac Firewalls

For remote job submission, you must open ports above 1024 on the remote (Linux) host.

7.2.3.2 Windows Firewalls

With Windows firewalls you will often see a popup dialog box asking you if you wish to block or unblock a specific application. Below are the scenarios where you need to unblock certain applications, and the list of applications in each case. Unless noted below these instructions apply to all versions of Windows operating systems. These application lists are followed by requirements for various firewalls.

Launching jobs:

- `perl.exe` (`%SCHRODINGER%\mmshare-v`*version*`\bin\`*platform*`\perl.exe`)
- `python.exe` (`%SCHRODINGER%\mmshare-v`*version*`\bin\`*platform*`\python.exe`)
- `sh.exe` (`%SCHRODINGER%\unxutils\sh.exe`)

Running remote jobs:

- `plink.exe`

McAfee

Add the remote machine to the Trusted IP list using the settings. You can configure the firewall to trust all the machines in your LAN as well, *but* make sure the subnet mask is properly set.

Norton

Should work when the "internet access" prompts to unblock the packets are answered. But make sure you "trust" your LAN in this case too.

7.3 Preparing for Batch Queue Submission

Schrödinger supplies support for the PBS, SGE (Sun Grid Engine, now Grid Engine, GE; here it will continue to be referred to as SGE), LSF, SLURM, and Torque queueing systems in the standard software installation. Below are links to information about these queueing systems:

- PBS: http://www.openpbs.org, http://www-unix.mcs.anl.gov/openpbs
- SGE: http://sourceforge.net/projects/gridscheduler, http://gridengine.org, http://www.univa.com/products/grid-engine.php
- LSF: http://www.platform.com/products/wm/LSF
- SLURM: https://computing.llnl.gov/linux/slurm/slurm.html
- Torque: http://www.adaptivecomputing.com/products/open-source/torque/

Enabling batch queue submissions to a supported queueing system for simple job submission only requires the addition of a few lines to the `schrodinger.hosts` file and the specification of the queueing system and the queue name in a configuration file. These additions are described in the next subsections. You can also make settings for use of OpenMPI, GPUs, and license checking.

The following subsections describe the general setup procedures and provide information on settings common to all queueing systems. Details for each queueing system are given in the following sections.

7.3.1 Setting Up the Hosts File for Batch Queues

To enable job submissions to a batch queue on a supported queueing system, you must add host entries that define the available queues to the `schrodinger.hosts` file. The command syntax is described in Table 7.1 on page 48. There are two settings that define the queue: the `Queue` setting and the `Qargs` setting. A sample of the host entries to be inserted into the hosts file is shown below:

```
# Batch submission to 'bigjobs' queue under PBS
Name: bigq
Host: cluster
Queue: PBS
Qargs: -q bigjobs
processors: 128
tmpdir: /storage/TMPDIR
#
# Batch submission to 'shortjobs' queue under PBS
Name: shortq
Host: cluster
Queue: PBS
```

```
Qargs: -q shortjobs
processors: 16
tmpdir: /storage/TMPDIR
```

This example defines two entries named `bigq` and `shortq` to which jobs can be sent on the host `cluster`.

The Job Control facility distinguishes batch queues from hosts by the presence of the `Queue` setting, which specifies the queueing system. The `Queue` setting must be set to the subdirectory of `$SCHRODINGER/queues` that contains the support files for the queueing system. The subdirectories for the supported queueing systems are `PBS`, `SGE`, `SLURM`, and `LSF`. Because of incompatible changes in versions of some queueing systems, the directories for the more recent versions have a version number appended, e.g. `PBS10.4`. The `Qargs` setting specifies command line arguments for the queueing system's job submission command; for SGE, for instance, this is the `qsub` command.

You must also include a `host` setting because the `name` setting is used to specify the queue. Like normal remote host entries, host entries for batch queues inherit settings made in the `localhost` entry of the `schrodinger.hosts` file. If the queueing software is available to all hosts to which users have access, you can set `host` to `localhost`. Otherwise you should set it to the host that runs the queueing software. The host that you use for the `host` setting must meet the following requirements:

- It must be able to submit, cancel and query jobs for the queueing system.
- It must have a single, static host name that can be resolved from the execution hosts (cluster nodes, for example) and the job submission hosts. Job Control must be able to reference this host using a single host name.
- The host name must always be associated with the same physical machine, so that connections using this host name go to the same machine (such as a cluster head node).

Batch queue entries can also have any of the other settings that host entries have, such as `schrodinger` and `tmpdir`. For queues on clusters, the `tmpdir` setting is required and should refer to a directory that is available to all the nodes and writable by all users who will use that queue. On shared memory machines, the `tmpdir` setting is optional. You should also add a `processors` setting for submission of distributed jobs.

You should consider adding an `env` setting for the `SCHRODINGER_LICENSE_RETRY` environment variable, particularly if there is likely to be a communication delay in obtaining a license or the license pool is oversubscribed. See Appendix B of the *Job Control Guide* for the syntax.

Note: You should *not* add `-cwd` to the `Qargs` setting for SGE, as this causes jobs to fail.

7.3.2 Setting Up Access to a Queueing System

Support for queueing systems is provided through a set of scripts and a template that contain the specific syntax required to submit jobs to a queue. These files are installed in a subdirectory of the `$SCHRODINGER/queues` directory that is named after the queue for each of the supported queueing systems. The name of this subdirectory is used as the name of the queueing system in the hosts file, as described above.

The Schrödinger software installation must be available to any host on which queue commands can be executed (the *queue host*).

To set up access to a supported queueing system, you must edit the `config` file in the `$SCHRODINGER/queues/`*queue* directory, and set the `QPATH` keyword to the path to the queueing software (the `bin` directory). If you need to set up the environment for the queuing system, you can set the `QPROFILE` keyword to the absolute path on the queue host of a configuration file. This file will be sourced to set up the environment to use the queue.

You should not need to edit any of the other files, as they are already tailored to the specific queueing system.

If you want to set up support for an unsupported queueing system or change the configuration for a supported queueing system, you can do so with the instructions in Appendix D.

7.3.3 Setting Up Queuing Systems for Open MPI Parallel Execution

Desmond, Impact, and Jaguar parallel execution use Open MPI 1.6.5, and can operate with a number of queuing systems. Open MPI provides tight integration that is compatible with many queuing systems via the PLS (Process Launch Subsystem) and RAS (Resource Allocation Subsystem) components. Loose integration, in which the queuing system is only responsible for allocating resources and dispatching the jobs, is also possible.

Instructions and requirements for the supported queuing systems are listed in the following subsections.

Note: The queues that are set up using the instructions below should *only* be used for jobs that run under MPI such as Desmond and Jaguar parallel jobs. They should *not* be used for distributed computing jobs, such as distributed Glide, LigPrep, and Prime jobs.

Open MPI can create large temporary files, which are written in the location defined by `TMPDIR`, `TEMP`, `TMP`, with a fallback to `/tmp`. To avoid performance problems, you should ensure that these files are written to a local file system with sufficient space, by setting one of these environment variables in the hosts file. For example,

`env: TEMP=/`*mylocaldisk*

```
env: SCHRODINGER_MPIRUN_OPTIONS=-x TEMP
```

For efficient queuing, it may be necessary to specify the number of processors (cores) per node as well as the total number of processors available. As Jaguar can also use OpenMP threads for parallelization in addition to MPI, you may also need to specify the maximum number of threads to use per MPI process. The `Qargs` setting in the hosts file can define these values with three basic variables:

- `%NPROC%`—the total number of CPU cores requested for the job
- `%TPP%`—the number of threads per process requested by the job
- `%PPN%`—the number of CPU cores (processors) per node

The values of `%NPROC%` and `%TPP%` are obtained from the command used to launch the job; whereas `%PPN%` is obtained from the `processors_per_node` setting in the hosts file.

General arithmetic is supported with these variables. For example,

```
%NPROC/TTP%
%NPROC/2%
%NPROC/2+1%
%TPP*4%
```

Integer division is rounded down, so `%NPROC/8%` evaluates to 0 if `%NPROC%` is less than 8. Instructions and examples are given in the sections below for each queueing system.

7.3.4 Setting Up Queuing Systems for GPU Use

Queuing systems must assign the appropriate resources to the GPU. This can be a bit challenging, as not all queueing systems fully manage GPU resources. Desmond MD jobs run on a GPU require 1 CPU per GPU. The `%NPROC%` placeholder will be substituted in the hosts file for the number of GPUs requested by the user at run time. Therefore, the same placeholder can be used for both (see Section 7.3.3 on page 65 for further examples of the `%NPROC%` placeholder in the hosts file). Details of the process are given in the sections below.

7.3.5 Setting Up License Checking for Queueing Systems

On large clusters, it is possible for jobs to fail because there are too few licenses for all the jobs to run. A mechanism has been provided to ensure that a queued job will not run unless all the licenses it requires are available. This facility is available for the SGE 6.0, PBS Pro 10.1, and Platform LSF 6.2 queueing systems. The configuration process is described in detail in the sections below, for each of the queueing systems that support license checking.

7.4 Setup for SGE Queueing Systems

First, set up the `config` file in your Schrödinger software installation:

1. Edit the file `$SCHRODINGER/queues/SGE/config`.
2. Set the `QPATH` keyword to the full path to the `bin` directory of the queueing software installation.
3. Set the `QPROFILE` keyword in the `config` file to the full path to the configuration file that sets up the environment for the queuing software.

   This file is called `settings.sh` (Bash) or `settings.csh` (C shell), and is in the `default/common` subdirectory of your queueing software installation.

Note: Do *not* use the `-cwd` argument to `qsub` in the `config` file, the hosts file, or SGE queue aliases, as this causes jobs to fail.

No instructions are given here for installing and general configuration of the queueing system. For information on this process, you should use the documentation provided with the queueing system, or go to the relevant web site:

- http://sourceforge.net/projects/gridscheduler
- http://gridengine.org
- http://www.univa.com/products/grid-engine.php

The instructions in the sections below are for the use of specific features: MPI, GPU use, and license checking. You only need to use these instructions if you intend to use these features.

7.4.1 OpenMPI Configuration for SGE

Any SGE queue to which parallel jobs are submitted must be configured to support multi-processor jobs. The `schrodinger.hosts` file must include an entry for each queue, which should look like the following:

```
Name: my-queue-name
Queue: SGE
Qargs: -q SGE-queue-name -pe pe %NPROC%
Host: my-cluster-name
processors: processors-in-queue
```

where *pe* is the name of the parallel environment. The value to use for *pe* depends on the queueing system configuration, and it may be necessary to consult the documentation for the queueing system to determine how to select a suitable value. The command `qconf -spl`

provides a list of defined parallel environments. An example of the output of this command is as follows:

```
lam
mpi
mpich
```

These names are merely labels and may have no particular significance—for example, choosing the `mpich` environment does not mean that you must run MPICH. You can find out more information about a defined particular parallel environment using the command:

```
qconf -sp pe
```

The SGE man page for `sge_pe` documents the output from this command.

The number of processors to specify is the total number of cores available to the queue. It is generally recommended that you use "fill-up" scheduling for the queues. The use of "round-robin" scheduling has a larger risk of failure on startup that may be related to an SGE bug (http://gridengine.sunsource.net/issues/show_bug.cgi?id=2393). You can set the allocation rule to `$fill_up` either in the QMON interface (started with the `qmon` command) or by editing the parallel environment with the command

```
qconf -mp pe
```

This command opens a text editor, in which you can change or add the `allocation_rule` setting. Regardless of the allocation rule, it is advisable to use a high-quality network for Desmond, and, if possible, to separate MPI traffic from other I/O traffic.

Below are some examples of host file entries:

```
# Request %NPROC% slots for the job using SGE parallel environment pe
Name: label
Queue: SGE
Qargs: -q SGE-queue -pe pe %NPROC%
Host: submit-node
Processors: processors-in-queue

# Parallel Jaguar: Hybrid OpenMPI/OpenMP.
# Request %NPROC% total threads, with %TPP% threads per node.
Name: label
Queue: SGE
Qargs: -q SGE-queue -pe orte_rr %NPROC/TPP% -l dedicated=%TPP%
Host: submit-node
Processors: processors-in-queue
```

For the second example, some configuration of the queue is required, in order to force SGE to distribute the job in a specific cpu topology. First, make sure your parallel environment (`orte_rr` in the example) has your desired allocation_rule (e.g. `$round_robin`, `$fill_up`). Next, create a complex resource representing the number of cores in each node. In this example, we call the resource `dedicated`. To set up the complex, run the following command and fill in the table with the row given below:

```
qconf -mc
```

```
#name               shortcut   type        relop requestable consumable default  urgency
#----------------------------------------------------------------------------------------
dedicated           ded        INT         <=    YES         YES        1        0
```

Finally, add the appropriate complex value to each execution host, with the value of `dedicated` equal to the number of slots on the host

```
qconf -me computehost-0-3
hostname              computehost-0-3.local
load_scaling          NONE
complex_values        dedicated=12,slots=12
```

7.4.2 GPU Configuration for SGE

Here is an example that can be used for GPU jobs in a very simple scheduling environment:

```
# SGE
name: sge-gpu
host: headnode.mycluster.com
queue: SGE
qargs: -q myqueue.q -pe smp %NPROC%
tmpdir: /usr/local/tmp
schrodinger: /opt/bin/schrodinger2015-2
gpgpu: 0, Tesla C2050
gpgpu: 1, Tesla C2050
gpgpu: 2, Tesla C2050
```

Theses entries have several assumptions and characteristics:

- GPUs are not managed explicitly. We assume that the queue called `myqueue` only includes hosts with GPUs.
- The nodes with GPUs are have the same number and type of GPUs (e.g. 3 Tesla C2050's)
- The job does not span nodes.

The drawback to these configurations is that they make reservations based on CPUs, and there may be more CPUs (or more properly, slots) available on each node than GPUs. Since the GPU resources are not being requested directly, then for these options, you should set the number of

slots in each node equal to the number of GPUs. That will ensure that the node is not overloaded by GPU tasks. Alternatively, you could arrange for the resource request to reserve the entire node outright.

If this type of scheduling proves inadequate, you will need to manage the GPU resources explicitly. To do this, define a consumable resource (named `gpus`, for example) and configure each host in the cluster to have the appropriate number of `gpus`. Then the hosts file entry should request not only the proper number of CPUs, but also the proper number of GPUs, as in the following example:

```
# SGE
name: sge-gpu
host: headnode.mycluster.com
queue: SGE
qargs: -q myqueue.q -pe smp %NPROC% -l gpus=1
tmpdir: /usr/local/tmp
schrodinger: /opt/bin/schrodinger2015-2
pgpu: 0, Tesla C2050
gpgpu: 1, Tesla C2050
gpgpu: 2, Tesla C2050
```

Note that you set `gpus=1` not `gpus=%NPROC%`. The reason for this is that SGE interprets the resource request as: "for every slot requested, give me `'-l gpus=N' gpus` resources". So, by setting `-l gpus=1` you are effectively asking for `%NPROC% gpus` resources as well.

7.4.3 License Checking for SGE

On large clusters, it is possible for jobs to fail because there are too few licenses for all the jobs to run. A mechanism has been provided to ensure that a queued job will not run unless all the licenses it requires are available. This facility is supported for the SGE queueing system from version 6.0 on. The configuration process consists of the following steps:

1. Define a resource representing each license in the queueing system configuration.
2. Configure the load sensor script, which reports how many tokens are available for each license, so that it can find the license file and the FLEXlm utility that it uses to query the license server. It can also be configured to reserve licenses for non-queue use.
3. Install this script in the appropriate queueing system directory.
4. Configure Job Control to specify license requirements when launching jobs.

Each of these steps is described in detail in the sections below.

7.4.3.1 Adding License Attributes

A resource attribute representing each license must be defined in the queueing system configuration. The utility `licutil` can be used to parse your license file or query the license server and generate the configuration text that needs to be added to the queueing system configuration files. The syntax is

`$SCHRODINGER/utilities/licutil -sgeconf [-f` *license-file*`]`

The `-f` *license-file* option is not required if your license file is in the default location, `$SCHRODINGER/license`, or if it is already specified by one of the environment variables `SCHROD_LICENSE_FILE` or `LM_LICENSE_FILE`. If you want to query the server directly, you must *not* use the `-f` option, and the environment variable you use must be set to the server location.

Note: If you specify multiple servers, only the first is queried. Licenses on other servers will therefore not be used.

The output from `licutil` should consist of lines like the following:

```
IMPACT_MAIN     IMPACT_MAIN    INT     <=     YES          YES          0    0
MMLIBS          MMLIBS         INT     <=     YES          YES          0    0
```

There should be one line for each license in your license file.

To add the new license attributes:

1. Copy the existing configuration to a text file:

   ```
   qconf -sc > complex.txt
   ```

   where `qconf` is the SGE queue configuration command.

2. Append the new configuration lines to `complex.txt`:

   ```
   $SCHRODINGER/utilities/licutil -sgeconf >> complex.txt
   ```

3. Load the new configuration:

   ```
   qconf -Mc complex.txt
   ```

This last command should produce a number of messages like

user@host `added "IMPACT_MAIN" to complex entry list`
user@host `added "MMLIBS" to complex entry list`

To verify that the attributes have been added, display the complex configuration with

```
qconf -sc
```

and make sure the new license attributes appear.

7.4.3.2 Configuring the Load Scripts

A script is provided that queries the FLEXlm license server and reports the number of tokens available for each license in a format that the queueing software can understand. To make sure the `lmutil` utility and license file can be found when the script is run by the queueing software, you need to add the locations of the program and the license file to the script, by editing the configuration section at the top of the script. You must have permission to change this file.

1. Use a text editor to edit the file `$SCHRODINGER/utilities/flexlm_sensor.pl`.
2. Edit the line beginning `my $LICENSE` so that the full path to the license file is between the quotation marks:

   `my $LICENSE="`*full-path-to-license-file*`";`

   You can specify the license server instead of the license file as follows:

   `my $LICENSE="\@`*license-server*`";`

   or

   `my $LICENSE="`*port*`\@`*license-server*`";`

   The backslash is necessary as Perl interprets @ as a special character.
3. Edit the line beginning `my $LMUTIL` so that the full path to `lmutil` is between the quotation marks:

   `my $LMUTIL="`*full-path-to-lmutil*`";`
4. To reserve licenses for non-queue use, edit the line beginning `my $reserved` to set the value to the fraction or percentage of licenses to reserve. For example, either of the following lines reserves 20% of the licenses for non-queue use.

   ```
   my $reserved = 0.2;
   my $reserved = "20%";
   ```
5. Save the modified script and close the editor.

Follow the instructions below to verify that the script is configured properly. If you test the script before and after specifying reserved licenses, you should see the number of licenses reported decrease by the specified percentage.

1. Enter the following command:

   ```
   perl $SCHRODINGER/utilities/flexlm_sensor.pl
   ```
2. Press the ENTER key.

You should get a block of output like

```
begin
global:IMPACT_MAIN:20
global:MMLIBS:20
end
```

showing the number of tokens available for each license. Pressing the ENTER key again updates the list, and the script waits for further input until you close the input stream by typing CTRL+D.

The log output from the script is sent to the system logging facility (syslog), so you can examine the syslog to find information about the actions performed by the script. More verbose information is sent to the `$logfile` which is set manually (and optionally) inside the script.

7.4.3.3 Installing the Load Script

To install `flexlm_sensor.pl`, you need to add a `load_sensor` attribute to the host configuration of one of the hosts managed by SGE. The load sensor is executed periodically on that host and feeds information about license availability back into the queueing system. You can get a list of the execution hosts using the command:

```
qconf -sel
```

To add the load sensor to the host configuration for the chosen host:

1. Enter the command

   `qconf -mconf` *hostname*

2. In the resulting edit window, add the line

   `load_sensor` *installation*`/utilities/flexlm_sensor.pl`

   where *installation* is the installation directory, `$SCHRODINGER`. You must specify the path explicitly—you cannot use environment variables in the SGE configuration files. If there is already a `load_sensor` line, replace it with the above line.

3. Save the modified file and close the editor.

4. Verify that your changes were accepted by displaying the host configuration with the command:

   `qconf -sconf` *hostname*

SGE starts the load sensor script automatically after you have made the `load_sensor` setting, and also restarts it automatically if you modify or update the script `flexlm_sensor.pl`. The command

```
qhost -F -h hostname
```

displays all of the resource attributes for *hostname* If the load sensor is running and working properly, you should see your licenses in this list.

For example,

```
gl:IMPACT_MAIN=20.000000
gl:MMLIBS=20.000000
```

SGE should be now configured to handle Schrödinger licenses properly.

7.4.3.4 Turning On License Checking in Job Control

The final step is to instruct Job Control to specify license requirements when submitting jobs to the queue. This step must be done after the steps above, otherwise jobs submitted to the queue will fail because the license software will assume that the required licenses are not available.

1. Open the file `$SCHRODINGER/queues/SGE/config` in a text editor.
2. Change the `LICENSE_CHECKING` line to read

   ```
   LICENSE_CHECKING=yes
   ```

3. Save the modified file and close the editor.

7.5 Setup for PBS Pro Queueing Systems

First, you must set up the config file in your Schrödinger software installation:

1. Edit the file `$SCHRODINGER/queues/PBS/config` (or the file in the relevant versioned subdirectory).
2. Set the `QPATH` keyword to the `bin` directory of your PBS queueing software installation.

No instructions are given here for installing and general configuration of the queueing system. For information on this process, you should use the documentation provided with the queueing system, or go to the relevant web site:

- http://www.openpbs.org
- http://www-unix.mcs.anl.gov/openpbs

The instructions in the sections below are for the use of specific features: MPI and license checking. You only need to use these instructions if you intend to use these features.

7.5.1 OpenMPI Configuration for PBS Pro

The TM API is used to allocate slots and launch processes in Open MPI for all of the family of PBS queuing systems. For more information of running jobs under PBS Pro, check the OpenMPI FAQs at http://www.open-mpi.org/faq/?category=tm.

Tight integration for PBS Pro 9.1 should work without any configuration. It might also work for older version of PBS Pro. If the bundled components do not work on your queuing system, you can remove those components and use loose integration:

```
cd $SCHRODINGER/mmshare-v version/lib/platform/openmpi/lib/openmpi
rm mca_plm_tm.la mca_plm_tm.so mca_ras_tm.la mca_ras_tm.so
```

If loose integration is used, however, your parallel jobs will not be the under control of PBS Pro, and in the case of a failure, the slave processes might not be terminated by MPI.

Here are some examples of host file entries:

```
# Request %NPROC% processors on a single node
Name: label
Queue: PBS10.4
Qargs: -q PBS-queue -l select=1:ncpus=%NPROC%:mpiprocs=%NPROC%
Host: submit-node
Processors: processors-in-queue

# Request %NPROC% processors, one on each node
Name: label
Queue: PBS10.4
Qargs: -q PBS-queue -l select=%NPROC%:ncpus=1:mpiprocs=1
Host: submit-node
Processors: processors-in-queue
# Request %NPROC% processors, spread across
# nodes, using 2 cores per node
Name: label
Queue: PBS10.4
Qargs: -q PBS-queue -l select=%NPROC/2%:ncpus=2:mpiprocs=2
Host: submit-node
Processors: processors-in-queue

# Parallel Jaguar: Hybrid OpenMPI/OpenMP.
# Request %NPROC% total threads, with %TPP% threads per node.
Name: label
Queue: PBS10.4
Qargs: -q PBS-queue -l nodes=%NPROC/TPP%:ncpus=%TPP%:mpiprocs=%TPP%
Host: submit-node
```

`Processors:` *processors-in-queue*

7.5.2 License Checking for PBS Pro

On large clusters, it is possible for jobs to fail because there are too few licenses for all the jobs to run. A mechanism has been provided to ensure that a queued job will not run unless all the licenses it requires are available. This facility is supported for the PBS Pro queueing system from version 10.1 on. The configuration process consists of the following steps:

1. Define a resource representing each license in the queueing system configuration.
2. Configure the load sensor script, which reports how many tokens are available for each license, so that it can find the license file and the FLEXlm utility that it uses to query the license server. It can also be configured to reserve licenses for non-queue use.
3. Install this script in the appropriate queueing system directory.
4. Configure Job Control to specify license requirements when launching jobs.

Each of these steps is described in detail in the sections below, for each of the three queueing systems that support license checking.

7.5.2.1 Adding License Attributes

A resource attribute representing each license must be defined in the queueing system configuration. The utility `licutil` can be used to parse your license file or query the license server and generate the configuration text that needs to be added to the queueing system configuration files. The syntax is

`$SCHRODINGER/utilities/licutil -pbsconf [-f` *license-file*`]`

The `-f` *license-file* option is not required if your license file is in the default location, `$SCHRODINGER/license`, or if it is already specified by one of the environment variables `SCHROD_LICENSE_FILE` or `LM_LICENSE_FILE`. If you want to query the server directly, you must *not* use the `-f` option, and the environment variable you use must be set to the server location.

Note: If you specify multiple servers, only the first is queried. Licenses on other servers will therefore not be used.

The output from `licutil` should consist of lines like the following:

```
Copy the text between the dashed lines below into your
<PBS_HOME>/server_priv/resourcedef file.
------------------------------------------------------------------------
IMPACT_MAIN  type=long
MMLIBS  type=long
------------------------------------------------------------------------
```

```
Add the text between the dashed lines below to the value of
the 'resources' parameter in your <PBS_HOME>/sched_priv/sched_config file.
E.g., if the 'resources' line in the file was
resources: "ncpus, mem, arch"
and the text between the lines was
feature1, feature3, feature5
then the new 'resources' line should be
resources: "ncpus, mem, arch, feature1, feature3, feature5"
---------------------------------------------------------------------
IMPACT_MAIN, MMLIBS
---------------------------------------------------------------------

Copy the text between the dashed lines below into your
<PBS_HOME>/sched_priv/sched_config file.
---------------------------------------------------------------------
server_dyn_res: "IMPACT_MAIN !/usr/bin/schro09/utilities/pbs_lic_sensor.pl IMPACT_MAIN"
server_dyn_res: "MMLIBS !/usr/bin/schro09/utilities/pbs_lic_sensor.pl MMLIBS"
---------------------------------------------------------------------
```

In the first and third parts of this output, there should be one line for each license in your license file; in the second part, one word for each license. To add the new license attributes, follow the instructions in the output. The files that you need to modify are `$PBS_HOME/server_priv/resourcedef` and `$PBS_HOME/sched_priv/sched_config`, where `$PBS_HOME` is the PBS home directory, often something like `/var/spool/PBS` on the head node of your cluster.

For the changes to take effect, you must restart PBS. The command for doing this is typically the following:

```
/etc/init.d/pbs restart
```

Refer to your *PBS Administrator Guide* for more information about restarting PBS.

7.5.2.2 Configuring the Load Script

A script is provided that queries the FLEXlm license server and reports the number of tokens available for each license in a format that the queueing software can understand. To make sure the `lmutil` utility and license file can be found when the script is run by the queueing software, you need to add the locations of the program and the license file to the script, by editing the configuration section at the top of the script. You must have permission to change this file.

1. Use a text editor to edit the file `$SCHRODINGER/utilities/pbs_lic_sensor.pl`.
2. Edit the line beginning `my $LICENSE` so that the full path to the license is between the quotation marks:

   `my $LICENSE="`*full-path-to-license-file*`";`

   You can specify the license server instead of the license file as follows:

   `my $LICENSE="\@`*license-server*`";`

   or

```
my $LICENSE="port\@license-server";
```

The backslash is necessary as Perl interprets @ as a special character.

3. Edit the line beginning `my $LMUTIL` so that the full path to `lmutil` is between the quotation marks:

```
my $LMUTIL="full-path-to-lmutil";
```

4. To reserve licenses for non-queue use, edit the line beginning `my $reserved` to set the value to the fraction or percentage of licenses to reserve. For example, either of the following lines reserves 20% of the licenses for non-queue use.

```
my $reserved = 0.2;
my $reserved = "20%";
```

5. Save the modified script and close the editor.
6. Verify that the script is configured properly by entering the following command:

```
perl $SCHRODINGER/utilities/pbs_lic_sensor.pl MMLIBS
```

The script should print a single number, which is the number of available licenses of the type specified as the command-line argument. You can replace `MMLIBS` with any other license type.

If you test the script before and after specifying reserved licenses, you should see the number of licenses reported decrease by the specified percentage.

The log output from the script is sent to the system logging facility (syslog), so you can examine the syslog to find information about the actions performed by the script. More verbose information is sent to the `$logfile` which is set manually (and optionally) inside the script.

7.5.2.3 Installing the Load Script

The location of the load sensor script, `pbs_lic_sensor.pl`, is specified in the `sched_priv/sched_config` file (see Section 7.5.2.2 on page 77). The configuration produced by `licutil` points to the location of this script within the Schrödinger installation, `$SCHRODINGER/utilities/pbs_lic_sensor.pl`. If this location is acceptable, no further action is necessary. If the script needs to be installed elsewhere, you must copy it to the required location and edit each line of `sched_priv/sched_config` that refers to the script and replace the path with the new path.

7.5.2.4 Turning On License Checking in Job Control

The final step is to instruct Job Control to specify license requirements when submitting jobs to the queue. This step must be completed after the steps above, otherwise jobs submitted to the

queue will fail because the license software will assume that the required licenses are not available.

1. Open the file `$SCHRODINGER/queues/`*queueing-system*`/config` in a text editor, where *queueing-system* is `SGE`, `PBS`, or `LSF`.
2. Change the `LICENSE_CHECKING` line to read

   ```
   LICENSE_CHECKING=yes
   ```
3. Save the modified file and close the editor.

7.6 Setup for Torque Queueing Systems

First, you must set up the `config` file in your Schrödinger software installation:

1. Edit the file `$SCHRODINGER/queues/Torque/config`.
2. Set the `QPATH` keyword to the `bin` directory of your Torque queueing software installation.

No instructions are given here for installing and general configuration of the queueing system. For information on this process, you should use the documentation provided with the queueing system, or go to the web site: http://www.clusterresources.com/products/torque-resource-manager.php.

The instructions in the sections below are for the use of specific features: MPI and GPU use. You only need to use these instructions if you intend to use these features.

7.6.1 OpenMPI Configuration for Torque

The TM API is used to allocate slots and launch processes in Open MPI for all of the family of PBS queuing systems. For more information of running jobs under Torque, check the OpenMPI FAQs at http://www.open-mpi.org/faq/?category=tm.

Due to a naming conflict, components for Torque have been placed under `$SCHRODINGER/mmshare-v`*version*`/lib/`*arch*`/openmpi/disabled_lib/openmpi`. To use the Torque queuing system, you should copy those components to the standard location:

```
cd $SCHRODINGER/mmshare-vversion/lib/arch/openmpi/disabled_lib/openmpi/
cp -rf mca_plm_tm.la mca_plm_tm.so mca_ras_tm.la mca_ras_tm.so \
   $SCHRODINGER/mmshare-vversion/lib/arch/openmpi/lib/openmpi
```

Note: If you are currently running multiple queuing systems from the same installation you may need to create two installations, one with these changes, and one without them.

The bundled Torque components depend on the `libtorque.so.2` library from Torque 2.2.1. If you do not have a compatible `libtorque.so.2` library on your system, you may also need to copy it:

```
cd $SCHRODINGER/mmshare-vversion/lib/arch/openmpi/disabled_lib/
cp libtorque.so.2 $SCHRODINGER/mmshare-vversion/lib/arch/openmpi/lib
```

Here are some examples of host file entries:

```
# Request %NPROC% processors on a single node
Name: label
Queue: Torque
Qargs: -q Torque-queue -l nodes=1:ppn=%NPROC%
Host: submit-node
Processors: processors-in-queue

# Request %NPROC% processors, one on each node
Name: label
Queue: Torque
Qargs: -q Torque-queue -l nodes=%NPROC%:ppn=1
Host: submit-node
Processors: processors-in-queue

# Request %NPROC% processors, spread across nodes, 4 cores per node
Name: label
Queue: Torque
Qargs: -q Torque-queue -l nodes=%(NPROC+3)/4%:ppn=4
Host: submit-node
Processors: processors-in-queue
# Parallel Jaguar: Hybrid OpenMPI/OpenMP.

# Request %NPROC% total threads, with %TPP% threads per node.
Name: label
Queue: Torque
Qargs: -q Torque-queue -l nodes=%NPROC/TPP%:ppn=%TPP%
Host: submit-node
Processors: processors-in-queue

# Request %NPROC% processors, spread across nodes using %PPN% cores per
# node. Illustrates the use of the processors_per_node field and the
# corresponding %PPN% placeholder
Name: label
Queue: Torque
Qargs: -q Torque-queue -l nodes=%(NPROC+PPN-1)/PPN%:ppn=%PPN%
```

`Host:` *submit-node*
`Processors:` *processors-in-queue*
`Processors_per_node:` *processors-per-node*

7.6.2 GPU Configuration for Torque

Here is an example that can be used for GPU jobs in a very simple scheduling environment:

```
# Torque
name: torque-gpu
host: headnode.mycluster.com
queue: Torque
qargs: -q myqueue -l nodes=1:ppn=%NPROC%
tmpdir: /usr/local/tmp
schrodinger: /opt/bin/schrodinger2015-2
gpgpu: 0, Tesla C2050
gpgpu: 1, Tesla C2050
gpgpu: 2, Tesla C2050
```

Theses entries have several assumptions and characteristics:

- GPUs are not managed explicitly. We assume that the queue called `myqueue` only includes hosts with GPUs.
- The nodes with GPUs are have the same number and type of GPUs (e.g. 3 Tesla C2050's)
- The job does not span nodes.

The drawback to these configurations is that they make reservations based on CPUs, and there may be more CPUs (or more properly, slots) available on each node than GPUs. Since the GPU resources are not being requested directly, then for these options, you should set the number of slots in each node equal to the number of GPUs. This will ensure that the node is not overloaded by GPU tasks. Or, you could arrange for the resource request to reserve the entire node.

If this type of scheduling proves inadequate, you will need to manage the GPU resources explicitly. To do this, define a consumable resource in your queueing system (named `gpus`, for example) and configure each host in the cluster to have the appropriate number of `gpus`. Then the hosts file entry should request not only the proper number of CPUs, but also the proper number of GPUs, as in the following examples:

```
# Torque
name: torque-gpu
host: headnode.mycluster.com
queue: Torque
qargs: -q myqueue -l nodes=1:ppn=%NPROC%:gpus=%NPROC%
tmpdir: /usr/local/tmp
schrodinger: /opt/bin/schrodinger2015-2
gpgpu: 0, Tesla C2050
gpgpu: 1, Tesla C2050
gpgpu: 2, Tesla C2050
```

7.7 Setup for LSF Queueing System

First, you must set up the `config` file in your Schrödinger software installation:

1. Edit the file `$SCHRODINGER/queues/LSF/config`.
2. Set the `QPATH` keyword to the full path to the `bin` directory of your LSF queueing software installation.
3. Set the `QPROFILE` keyword in the `config` file to the full path to the configuration file that sets up the environment for the queuing software.

   This file is called `profile.lsf` (Bash) or `cshrc.lsf` (C shell), and is in the `conf` subdirectory of your queueing software installation.

No instructions are given here for installing and general configuration of the queueing system. For information on this process, you should use the documentation provided with the queueing system, or go to the web site: http://www.platform.com/products/wm/LSF.

The instructions in the sections below are for the use of specific features: MPI, GPU use, and license checking. You only need to use these instructions if you intend to use these features.

7.7.1 OpenMPI Setup

Loose integration for LSF should work without change. Tight integration is supported by LSF from version 7.0.2 on. Some examples are given below.

To simply select `%NPROC%` processors:

```
qargs: -q queue_name -n %NPROC%
```

To tile the job across several nodes, using exactly 8 processors per node:

```
qargs: -q queue_name -n %NPROC% -R \"span[ptile=8]\"
```

Here are some examples of host file entries:

```
# Request %NPROC% processors with an arbitrary cpu topology
Name: label
Queue: LSF
Qargs: -q LSF-queue -n %NPROC%
Host: submit-node
Processors: processors-in-queue
```

```
# Request %NPROC% processors spread evenly over two nodes
Name: label
Queue: LSF
Qargs: -q LSF-queue -n %NPROC% -R \"span[hosts=2]\"
Host: submit-node
Processors: processors-in-queue

# Request %NPROC% processors, spread across
# nodes, using 2 cores per node
Name: label
Queue: LSF
Qargs: -q LSF-queue -n %NPROC% -R \"span[ptile=2]\"
Host: submit-node
Processors: processors-in-queue
# Parallel Jaguar: Hybrid OpenMPI/OpenMP.
# Request %NPROC% total threads, with %TPP% threads per node.
Name: label
Queue: LSF
Qargs: -q LSF-queue -n %NPROC% -R \"span[ptile=%TPP%]\"
Host: submit-node
Processors: processors-in-queue
```

7.7.2 GPU Setup for LSF

Here is an example that can be used for GPU jobs in a very simple scheduling environment:

```
# LSF
name: lsf-gpu
host: headnode.mycluster.com
queue: LSF
qargs: -q myqueue -n %NPROC% -R \"span[hosts=1]\"
tmpdir: /usr/local/tmp
schrodinger: /opt/bin/schrodinger2015-2
gpgpu: 0, Tesla C2050
gpgpu: 1, Tesla C2050
gpgpu: 2, Tesla C2050
```

Theses entries have several assumptions and characteristics:

- GPUs are not managed explicitly. We assume that the queue called `myqueue` only includes hosts with GPUs.
- The nodes with GPUs are have the same number and type of GPUs (e.g. 3 Tesla C2050's)
- The job does not span nodes.

The drawback to these configurations is that they make reservations based on CPUs, and there may be more CPUs (or more properly, slots) available on each node than GPUs. Since the GPU resources are not being requested directly, then for these options, you should set the number of slots in each node equal to the number of GPUs. That will ensure that the node is not overloaded by GPU tasks. Alternatively, you could arrange for the resource request to reserve the entire node outright.

If this type of scheduling proves inadequate, you will need to manage the GPU resources explicitly. To do this define a consumable resource in the queueing system (named `gpus`, for example) and configure each host in the cluster to have the appropriate number of `gpus`. Then the hosts file entry should request not only the proper number of CPUs, but also the proper number of GPUs, as in the following examples:

```
# LSF
name: lsf-gpu
host: headnode.mycluster.com
queue: LSF
qargs: -q myqueue -n %NPROC% -R \"span[hosts=1] rusage[gpus=%NPROC%]\"
tmpdir: /usr/local/tmp
schrodinger: /opt/bin/schrodinger2015-2
gpgpu: 0, Tesla C2050
gpgpu: 1, Tesla C2050
gpgpu: 2, Tesla C2050
```

7.7.3 License Checking for LSF

On large clusters, it is possible for jobs to fail because there are too few licenses for all the jobs to run. A mechanism has been provided to ensure that a queued job will not run unless all the licenses it requires are available. This facility is supported for the Platform LSF queueing system from version 6.2 on. The configuration process consists of the following steps:

1. Define a resource representing each license in the queueing system configuration.
2. Configure the External Load Information Manager (ELIM) script, which reports how many tokens are available for each license, so that it can find the license file and the FLEXlm utility that it uses to query the license server. It can also be configured to reserve licenses for non-queue use.
3. Install this script in the appropriate queueing system directory.
4. Configure Job Control to specify license requirements when launching jobs.

Each of these steps is described in detail in the sections below.

7.7.3.1 Adding License Attributes

A resource attribute representing each license must be defined in the queueing system configuration. The utility `licutil` can be used to parse your license file or query the license server and generate the configuration text that needs to be added to the queueing system configuration files. The syntax is

`$SCHRODINGER/utilities/licutil -lsfconf [-f` *license-file*`]`

The `-f` *license-file* option is not required if your license file is in the default location, `$SCHRODINGER/license`, or if it is already specified by one of the environment variables `SCHROD_LICENSE_FILE` or `LM_LICENSE_FILE`. If you want to query the server directly, do *not* use the `-f` option, but set the environment variable to the server location.

Note: If you specify multiple servers, only the first is queried. Licenses on other servers will therefore not be used.

The output from `licutil` should consist of lines like the following:

```
Copy the text between the dashed lines below into the "Resource" section of
your lsf.shared file. (Don't, however, copy the RESOURCENAME line below if
there's already such a line there.)
------------------------------------------------------------------------
RESOURCENAME             TYPE      INTERVAL  INCREASING  DESCRIPTION
IMPACT_MAIN              Numeric         10           N  (IMPACT_MAIN license)
MMLIBS                   Numeric         10           N  (MMLIBS license)
------------------------------------------------------------------------

Copy the text between the dahsed lines below into the "Parameters" section
of your lsf.cluster.<your_cluster_name> file:
------------------------------------------------------------------------
LSF_ELIM_DEBUG=y
------------------------------------------------------------------------

Copy the text between the dahsed lines below into the "ResourceMap" section
of your lsf.cluster.<your_cluster_name> file.  (Don't, however, copy the
RESOURCENAME line below if there's already such a line there.)
------------------------------------------------------------------------
RESOURCENAME             LOCATION
IMPACT_MAIN              [all]
MMLIBS                   [all]
------------------------------------------------------------------------
```

In the first and third parts of this output, there should be one line for each license in your license file. To add the new license attributes, follow the instructions in the output. The files you need to modify, `lsf.shared` and `lsf.cluster.<your_cluster_name>`, are located in the `$LSF_CONFDIR` directory, which is usually something like `/usr/lsf/conf`. If this

directory is not cross-mounted between all of the hosts to which LSF can submit jobs, you will have to make these changes in the configuration files on each host.

To verify that the changes were made correctly, restart LSF with the commands

```
lsadmin reconfig
badmin mbdrestart
```

and then list the resources LSF recognizes using the command

```
lsload -l
```

The output should include a column for each of the new license resources, for example,

```
HOST_NAME               status  <...>   mem  IMPACT_MAIN MMLIBS
host1.mycluster.com         ok  <...>  365M            -      -
host2.mycluster.com         ok  <...>  362M            -      -
```

7.7.3.2 Configuring the Load Script

A script is provided that queries the FLEXlm license server and reports the number of tokens available for each license in a format that the queueing software can understand. To make sure the `lmutil` utility and license file can be found when the script is run by the queueing software, you need to add the locations of the program and the license file to the script, by editing the configuration section at the top of the script. You must have permission to change this file.

1. Use a text editor to edit the file `$SCHRODINGER/utilities/elim.schrodinger`.
2. Edit the line beginning `my $LICENSE` so that the full path to the license is between the quotation marks:

   `my $LICENSE="`*full-path-to-license-file*`";`

   You can specify the license server instead of the license file as follows:

   `my $LICENSE="\@`*license-server*`";`

   or

   `my $LICENSE="`*port*`\@`*license-server*`";`

   The backslash is necessary as Perl interprets @ as a special character.
3. Edit the line beginning `my $LMUTIL` so that the full path to `lmutil` is between the quotation marks:

   `my $LMUTIL="`*full-path-to-lmutil*`";`
4. To reserve licenses for non-queue use, edit the line beginning `my $reserved` to set the value to the fraction or percentage of licenses to reserve. For example, either of the following lines reserves 20% of the licenses for non-queue use.

```
my $reserved = 0.2;
my $reserved = "20%";
```

5. Save the modified script and close the editor.
6. Verify that the script is configured properly by entering the following command:

   ```
   perl $SCHRODINGER/utilities/elim.schrodinger
   ```

The script should print a line like the following every thirty seconds:

```
2 IMPACT_MAIN 10 MMLIBS 15
```

The first number is the number of different licenses you have, followed by the name and token count for each license. Exit the script by typing CTRL+C.

If you test the script before and after specifying reserved licenses, you should see the number of licenses reported decrease by the specified percentage.

The log output from the script is sent to the system logging facility (syslog), so you can examine the syslog to find information about the actions performed by the script. More verbose information is sent to the `$logfile` which is set manually (and optionally) inside the script.

7.7.3.3 Installing the Load Script

1. Copy the edited script into the `$LSF_SERVERDIR` directory on the LSF master host:

   ```
   cp $SCHRODINGER/utilities/elim.schrodinger $LSF_SERVERDIR
   ```

   The `$LSF_SERVERDIR` directory is typically named something like `/usr/lsf/6.2/linux2.6-glibc2.3-x86/etc/` and also contains the program `melim`, which manages ELIMs.

2. Restart LSF with the following commands:

   ```
   lsadmin reconfig
   badmin mbdrestart
   ```

3. Check that the new ELIM is active, using the command

   ```
   lsload -l
   ```

The output should now contain the number of available tokens for each license; for example,

```
HOST_NAME               status  <...>   mem  IMPACT_MAIN MMLIBS
host1.mycluster.com        ok   <...>  365M           10     15
host2.mycluster.com        ok   <...>  365M           10     15
```

LSF should now be configured to handle Schrödinger license requirements.

7.7.3.4 Turning On License Checking in Job Control

The final step is to instruct Job Control to specify license requirements when submitting jobs to the queue. This step must be done after the steps above, otherwise jobs submitted to the queue will fail because the license software will assume that the required licenses are not available.

1. Open the file `$SCHRODINGER/queues/LSF/config` in a text editor.
2. Change the `LICENSE_CHECKING` line to read

   ```
   LICENSE_CHECKING=yes
   ```

3. Save the modified file and close the editor.

7.8 Configuring a SLURM Queueing System

First, you must set up the `config` file in your Schrödinger software installation:

1. Edit the file `$SCHRODINGER/queues/SLURM2.1/config`.
2. Set `QPATH` to the `bin` directory of the SLURM queueing software installation.

No instructions are given here for installing and general configuration of the queueing system. For information on this process, you should use the documentation provided with the queueing system, or go to the web site: https://computing.llnl.gov/linux/slurm/slurm.html.

The instructions in the sections below are for the use of MPI. You only need to use these instructions if you intend to use this feature.

7.8.1 OpenMPI configuration

Below are some examples of SLURM entries in the hosts file.

```
# Request %NPROC% processors with an arbitrary cpu topology
Name: label
Queue: SLURM2.1
Qargs: --partition=SLURM-partition --ntasks=%NPROC%
Host: submit-node
Processors: processors-in-queue

# Request %NPROC% processors, one on each node
Name: label
Queue: SLURM2.1
Qargs: --partition=SLURM-partition --nodes=1 --ntasks-per-node=%NPROC%
Host: submit-node
Processors: processors-in-queue
```

```
# Request %NPROC% processors, spread across nodes, 2 cores per node
Name: label
Queue: SLURM2.1
Qargs: --partition=SLURM-partition --nodes=%NPROC/2% --ntasks-per-node=2
Host: submit-node
Processors: processors-in-queue

# Parallel Jaguar: Hybrid OpenMPI/OpenMP.
# Request %NPROC% total threads, with %TPP% threads per node.
Name: label
Queue: SLURM2.1
Qargs: --partition=SLURM-partition --nodes=%NPROC/TPP% --ntasks-per-
node=%TPP%
Host: submit-node
Processors: processors-in-queue
```

Note that the `Qargs` entry should all be on a single line, with no break.

7.9 Configuring Clusters

The configuration of a cluster to run Schrödinger software must take into account the special issues of communication between the compute nodes, the manager nodes, and the job submission host, and the impact that this communication might have on performance.

7.9.1 Configuration Requirements

Like any other host, each compute node must have access to a license, the software and the job-related files. The requirements can be stated as follows:

1. Schrödinger software installations must be available to all hosts: the job submission host, the manager node, and the compute nodes. These installations must contain the same software versions, but they could be in separate physical installations.

   To reduce network traffic, Schrödinger software should be installed either on each compute node's local disk, or on a file system that is accessible internally to all cluster nodes (that is, one that does not create network traffic through the manager node to the external network).

2. The job submission host and the compute nodes must be able to open socket connections to the FlexLM license server.

   The license file can be stored on the external network, the internal network, or copied to each node. Since this file is small, the location does not matter.

3. Compute nodes must be able to open socket connections to each other and to the manager node. They must also be able to connect to a node or host from which jobs can be submitted.
4. Passwordless SSH must be enabled:
    a. from the job submission host to the manager node;
    b. from the compute nodes to the manager node.
5. The host that runs the queueing software must have a static host name that is resolvable from any host that submits jobs or runs jobs, and the host name must always be associated with the same physical host. Communications addressed to that host name must not be redirected to another physical host.

For MPI parallel execution of Jaguar, the following are required in addition:

- The user's home directory must be mounted on the compute nodes.
- Passwordless SSH must be enabled between compute nodes.

7.9.2 Performance Optimization

To optimize the performance of a cluster for Schrödinger software, we suggest that you consider the following options when purchasing, upgrading, or configuring a cluster:

- Invest in a highly capable file server for the external network.
- Invest in shared storage for the private (intra-cluster) network, to reduce traffic to and from the external network. Shared storage makes installation and maintenance of the code much simpler, and can be used to store large data files, temporarily or permanently.
- Divide services among several management nodes. For example, the queueing system, the private network's shared storage and the routing could all be handled by separate management nodes. Likewise, nodes used as file servers should not run computations. (However, load balancing of these functions among management nodes may cause job failure.)
- Ensure that the management nodes have fast processors, large memory, and high-quality motherboards and network interfaces.
- Run more recent Linux versions, which have better facilities for network address translation (NAT) and related functionality than earlier versions.
- Store large databases on high-performance network-attached storage for efficient generation, management, and screening.
- Run a robust queuing system that is relatively immune to stalling, crashing or bringing

down its host if it is heavily loaded.

If you are likely to have more than 250 actively running jobs on the cluster, it is important to make some system-level adjustments.

- Increase the shell limit on open file handles, to 10240 for example, e.g. on Linux,

  ```
  ulimit -n 10240
  ```

  This setting should be made in the global `.bashrc` and `.cshrc` on each machine in the cluster, so that the user environment limits are always set appropriately.

  It might also be necessary to set this limit in the environment of the queueing system, since the queueing system's daemons are responsible for starting jobs. For SGE, you can set the following in the global configuration:

  ```
  execd_params S_DESCRIPTORS=10240 H_DESCRIPTORS=10240
  ```

- Increase the backlog of TCP SYN requests:

  ```
  echo "10240" > /proc/sys/net/ipv4/tcp_max_syn_backlog
  ```

  This setting is not persistent across reboots, so you will need ensure this comand is executed at boot time, perhaps in an init script.

7.10 Testing the Installations and Connections

Once you have installed the software and set up the hosts file on the desired hosts, you should test the installation. To do this, you can use the Diagnostics panel, or you can run checks from the command line.

7.10.1 The Diagnostics Panel

The Diagnostics panel runs checks on the license server and the hosts file, and can be used to run test applications on hosts to ensure that the connections are working properly.

To open the Diagnostics panel, use one of the following methods:

- **Linux:** Enter `$SCHRODINGER/diagnostics` in a terminal window.
- W**indows:** Choose Start → All Programs → Schrodinger-2015-2 → Diagnostics, or enter `diagnostics` in a Schrodinger Command Prompt window.
- **Mac:** Double-click the Diagnostics icon in the SchrodingerSuite2015 folder of the Applications folder, or enter `$SCHRODINGER/diagnostics` in a terminal window.

This panel can also be opened from the Help menu in Maestro and Canvas. It is opened automatically if Maestro fails to start.

After starting the application, a set of tests is run to verify that the licenses are correctly installed and the license server is running, and to verify that the hosts file is correctly set up. Their progress is reported in the Running Tests information box, which is displayed along with the Diagnostics panel. The results of these tests are then displayed in the panel.

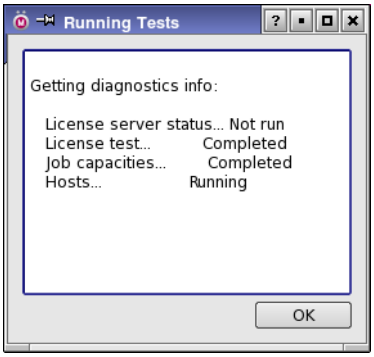

Figure 7.1. The Running Tests dialog box.

The host tests that are run when the panel opens do not include test jobs on remote hosts, only syntax checking on the hosts file. To run a test job on a host, select the row for that host in the Hosts table (Hosts tab), and click Selected Host. To run test jobs on all hosts in the hosts file, click All Hosts. The status of the test is reported in the Result column of the table, with messages in the Results area below the table.

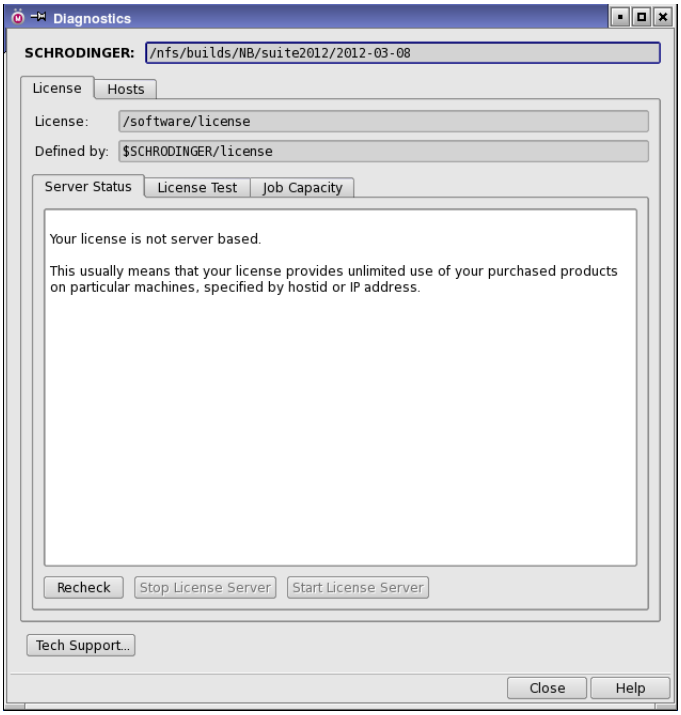

Figure 7.2. The Diagnostics panel.

If the result for a queue host is `OK (SECURE)` then file transfer to that host is secured by the Secure Channel (SSH tunnel) mechanism; if the result does not include `(SECURE)`, then the file transfer is not secured by this mechanism.

To change the reporting level, you can start the application from the command line with the `-l` option, which takes the values `critical`, `error`, `warning`, `info`, `debug`. These values represent different levels of reporting. The default level is `error`.

If you run this application on the license server host, you can use the Stop License Server and Start License Server buttons to stop and start the license server. These buttons are not available otherwise.

The hosts file is identified by the same mechanism for locating the hosts file that is used when you start a job. If you have a custom hosts file in your Schrödinger user resources directory, it uses that file. If you have a custom hosts file in some other location, you can test it from the command line by changing to the directory containing that file and running this application.

The results of the diagnostic tests can be collected so that you can send them to technical support. Click Technical Support, which collects the results into a zip archive and opens a

dialog box that gives the path to the archive. You can upload this file to the web form when requesting technical support (at https://www.schrodinger.com/supportcenter).

7.10.2 Command-Line Tools

You can test the connections to remote hosts with the `installation_check` application. This application reads a hosts file and runs a test job using the host entries defined in the file. Errors in the hosts file and failures in the test jobs are reported and (to the extent possible) recommendations are given for fixing the problems that were uncovered. It also checks the status of the license server and access to licenses.

The syntax of the command is as follows:

```
installation_check [options]
```

To run this command on Windows, open a Schrödinger Command Prompt window (from Start → All Programs → Schrodinger-2015-2) and enter the command. To run this command on Linux or Mac, prepend it with `$SCHRODINGER/`. For a description of the options, run the command with the `-h` option.

In addition to output to the terminal, the following output is generated: a summary, *jobname*`.summary`, a directory, *jobname*, and a zip archive, *jobname*`.zip`, that contains the contents of the directory and the summary.

The `installation_check` application runs a test program, `testapp`, that exercises all the Job-Control-related features of a real application. You can run this program as follows:

```
testapp [options] [jobname|inputfile]
```

For a description of the options, run the command with the `-h` option.

If no input file is specified, either a run time (`-t`) or a number of subjobs (`-n`) needs to be specified. If subjobs are specified, then the job runs until the subjobs all finish, regardless of the specified run time.

The standard Job Control options `-HOST`, `-NOLAUNCH`, `-SAVE`, and `-TMPDIR`, described in Table 2.1 of the *Job Control Guide*, and the extra options `-INTERVAL`, `-LOCAL`, `-NOJOBID`, and `-WAIT`, described in Table 2.2 of the *Job Control Guide*, are also supported.

Distributed computing can be tested with `para_testapp`. The syntax of the command is:

```
para_testapp [options]
```

For a description of the options, run the command with the `-h` option. The `-DRIVERHOST` option can be used to specify the host to run the driver job, otherwise it is the first host specified by `-HOST`.

If the number of subjobs is not specified explicitly with the `-n` option, then only a single subjob is started. Likewise, the number of processors to use must be specified explicitly using the `-HOST` option. For jobs submitted to a batch queue, the `Processors` line from the hosts file entry is used if it is present. `-LOCAL` and `-NOLOCAL` can be used to specify the location of temporary files, and `-LOCALSUBJOB` can be used to run the subjobs with `-LOCAL`. To test the startup, use `-NOLAUNCH` which stops short of actually starting the job.

Setting Environment Variables

This appendix describes how to set an environment variable on the supported platforms.

A.1 Windows

You can create or change environment variables in the Environment Variables dialog box. If you are adding to the `PATH` environment variable, you should separate each field with a semicolon (;).

To open the Environment Variables dialog box:

1. Click Start, then click Control Panel.

   The Control Panel opens.

2. Click User Accounts.

3. Click User Accounts again.

4. In the Task side pane on the left, click Change my environment variables.

   The Environment Variables dialog box opens.

To create a new environment variable:

1. In the User variables section, click New.

   The New User Variable dialog box opens.

2. Enter the name of the variable and its value, and click OK.

   The New User Variable dialog box closes, and the variable is added to the User variables section of the Environment Variables dialog box.

3. Click OK in the Environment Variables dialog box.

To modify an existing environment variable:

1. In the User variables section, select the environment variable you want to modify.

2. Click Edit.

   The Edit User Variable dialog box opens.

3. Change the value of the variable and click OK.

   The Edit User Variable dialog box closes, and the variable is updated in the User variables section of the Environment Variables dialog box.

When you have finished creating or editing environment variables, click OK in the Environment Variables dialog box to save the values. You can then close the Control Panel.

A.2 Mac OSX

To set an environment variable on Mac OSX, first open a terminal window.

If you are setting the environment variable to run jobs from the command line in a terminal window, use the following command:

`export` *variable*`=`*value*

where *variable* is the name of the environment variable (such as `SCHRODINGER`) and *value* is the value you want to assign to the variable, (such as `/opt/schrodinger2015`).You can find out which environment variables have been set with the `env` command.

If you are setting the environment variable globally to use with applications, use the commands given below. The environment variables set by these commands are inherited by any shell or application.

OSX 10.7:

Enter the following command:

`defaults write ~/.MacOSX/environment` *variable* `"`*value*`"`

To find out which environment variables have been set, enter the following command:

```
defaults read ~/.MacOSX/environment
```

OSX 10.8, 10.9:

Enter the following command:

`launchctl setenv` *variable* `"`*value*`"`

A.3 Linux

To set an environment variable on Linux, enter the following command at a shell prompt:

csh/tcsh: `setenv` *variable value*

bash/ksh: `export` *variable*`=`*value*

where *variable* is the name of the environment variable (such as `SCHRODINGER`) and *value* is the value you want to assign to the variable, (such as `/opt/schrodinger2015`).

File and Resource Locations on Windows

This appendix contains information on where various resources or files are kept on Windows. These are relative to the standard application data location (`%APPDATA%`), local application data location (`%LOCALAPPDATA%`), and user location (`%USERPROFILE%`), whose definitions are listed below.

`%USERPROFILE%` is set to `C:\Users\`*username*.
`%APPDATA%` is set to `%USERPROFILE%\AppData\Roaming`.
`%LOCALAPPDATA%` is set to `%USERPROFILE%\AppData\Local`.

Table B.1. Location of files written or used by Schrödinger software.

Main Location	Folder	Files
`%APPDATA%\Schrodinger\`		`web_proxy.json`
	`maestro`*NN*	Maestro resource files, such as `prefer.cmd`, `default.menu`, `rotamer.res`, and so on.
	`scripts`*N*`.`*M*	Command-line scripts
`%LOCALAPPDATA%\Schrodinger\`		
	`Installer\`*year*	Installer log files, (`SCHRODINGER_INSTALLER_LOGFILE_DIR`), `schrodingerRegCure_`*year*`.log`.
	`Installer`	`SchrodingerProcKill.log`
	`tmp`	Maestro temporary files, including scratch projects (`SCHRODINGER_TEMP_PROJECT`, `MAESTRO_TEMP_LOCATION`) Job directories (`SCHRODINGER_TMPDIR`)
	`.jobdb2`	Job database
	`appcrash`	Application crash files, `maestro_error.txt`, `maestro.EXE.dmp`
	`maestro`*NN*	Maestro recent projects list (`mruprojectlist`)
`%USERPROFILE%\Schrodinger`		Customized `schrodinger.hosts`, `license.txt`
	queues	Custom queue definition files.
	product	Customized data files such as `reagentprep.ini` (CombiGlide), solvation files, etc.

Access to the Web via a Proxy

Some Schrödinger applications can retrieve information from the web. If you are using a proxy server for web access, you may have to perform some configuration to allow these applications access to the web.

For two applications, `getpdb` (which retrieves a PDB file from the web) and `checkupdates` (which checks the Schrödinger site for software updates), a script has been provided that creates and stores the necessary information about the proxy server. Using this script should should work for any proxy server that supports digest or basic authentication.

You can run this script with the following command (on Windows, open a Schrodinger Command Prompt window first):

`$SCHRODINGER/run proxy_config.py [`*options*`]`

The options are given in Table C.1.

Table C.1. Options for the proxy_config.py script

Option	Description
`-c[heck]` url	Check a specific URL.
`-f[ile]` *filename*	Specify the file that stores the proxy configuration.
`-h[elp]`	Display usage message and exit.
`-t[est]`	Test a previously configured proxy.
`-v[ersion]`	Display the program version and exit.

This script tests for access to http and https URLs, and if it can access these without problems it does not ask for proxy information. If access to either http or https URLs is blocked, it asks for the web proxy URL.

If the web proxy requires a login and password, the script asks for these to test the configuration, but it does *not* store the login and password. When you run either of these applications, you will be prompted for your login and password.

The script stores the proxy information in a file named `web_proxy.json`. The default location for this file is your Schrödinger user resources directory (`$HOME/.schrodinger` on linux, `%APPDATA%\Schrodinger` on Windows). If you want to store the information in

another location, run the script with `-f[ile]` *filename*. For example, if you want to install this file as a global resource for all users, use `-f $SCHRODINGER/web_proxy.json`.

When the application attempts to contact the web, it looks for the proxy information in the following locations, and uses the first location that it finds:

- The file specified by the environment variable `SCHRODINGER_PROXY_CONFIG`.
- The file `web_proxy.json` file in your user resources directory (`$HOME/.schrodinger` on Linux, `%APPDATA%` on Windows).
- The file `web_proxy.json` in the software installation (`$SCHRODINGER`).

If you do not want to use any of these locations, set `SCHRODINGER_PROXY_CONFIG` to an empty string.

The `getpdb` application requires access to the addresses listed below:

www.rcsb.org
eutils.ncbi.nlm.nih.gov
www.ncbi.nlm.nih.gov
helixweb.nih.gov
www.schrodinger.com

Two other utilities, `rsync_pdb` and `update_BLASTDB`, perform updates of databases from the web. These utilities use `rsync` on Linux and `wget` on Windows. To allow access via a proxy on Linux, you can set the `RSYNC_PROXY` environment variable to *hostname*`:`*port*, where *hostname* is the proxy server name and *port* is the port used. (You must use just the host name: do not prepend it with `http://`.) You can use `-p` *port* when you run these utilities to specify the port.

Adding an Unsupported Queueing System

To add support for an unsupported queueing system, the following files must be created in a subdirectory of `$SCHRODINGER/queues` that is named to identify the queueing system:

- A `submit` script, which is a wrapper for the queueing system's own job submission utility (`qsub` for PBS and SGE).
- A `cancel` script, which is a wrapper for the queueing system's job removal command (`qdel` for PBS and SGE).
- A `template.sh` file, which is a template for the shell script that is actually submitted to the batch queue and used to launch your calculation on the execution host.
- A `config` file, which contains settings for the keywords `QPATH`, `QSUB`, `QDEL`, `QSTAT`, and `QPROFILE`.

The name of the subdirectory that contains these files is used as the name of the queueing system in the hosts file.

D.1 The submit Script

The `submit` script needs to support the command line syntax:

`submit` *job-script* [*qsub-options*]

where *job-script* is the name of a shell script that starts a job on the queue. This is always the first (and possibly only) command-line argument to `submit`. Anything else on the command line must be passed on as arguments to the actual job-submission command.

If job submission is successful, `submit` should extract the batch ID from the output of the underlying job-submission command and report it in its output, in the form:

`BatchId:` *batchid*

If job submission fails for some reason, the script should exit with a non-zero exit code.

If you are creating your own `submit` script to support a new queueing system, you can use the `submit` scripts provided for PBS, SGE, and LSF as templates. Use the `QSUB` variable rather than the actual submission command in your script, and define `QSUB` in the `config` file.

D.2 The cancel Script

The `cancel` script must support the command line syntax:

`cancel` *batchid*

where *batchid* is a batch ID assigned by the queueing system. Job Control keeps track of the batch ID of each submitted job so that the ID can be used for cancelling jobs. The `cancel` script should return a nonzero exit status if the operation fails, for Job Control to be able to detect the failure.

If you are creating your own `cancel` script to support a new queueing system, you can use the `cancel` scripts provided for PBS, SGE, and LSF as templates. Use the `QDEL` variable rather than the actual submission command in your script, and define `QDEL` in the config file.

D.3 The Job Script Template File

The `template.sh` file is a skeleton for the Bourne-shell script that is actually submitted to the batch queue. The Schrödinger job-launching mechanism reads this file and inserts the commands necessary to launch the user's job, and then submits the resulting file to the queueing system using the `submit` command described above.

The following information from the `template.sh` file supplied for the PBS system illustrates how the `template.sh` file works.

```
#!/bin/sh
#PBS -N %PBSNAME%
#PBS -j oe
#PBS -r n
#PBS -l nodes=%NPROC%
#
# Batch-submission script for OpenPBS system
#

PATH=/usr/bin:/bin:/usr/bsd:/usr/sbin:/sbin:/usr/local/bin:$PATH

QPATH=/usr/local/pbs/bin
curdir=`echo $0 | sed -e 's#/[^/]*$##'`
if [ -f "$curdir/config" ]; then
  . $curdir/config
fi

PATH=$QPATH:$PATH
```

```
SCHRODINGER_BATCHID="$PBS_JOBID"
export SCHRODINGER_BATCHID

if [ -n "$PBS_NODEFILE" ]; then
  SCHRODINGER_NODEFILE="$PBS_NODEFILE"
  export SCHRODINGER_NODEFILE
fi

%ENVIRONMENT%

%COMMAND%
```

The `#PBS` lines are directives that are interpreted by PBS. In this case, the first directive sets the job name for this job to the Schrödinger job name, while the fourth specifies the number of processors to use for the job. Most other queueing systems also allow directives to be provided in the initial comment lines of the job submission scripts.

The words delimited by percent signs are variables, which are replaced at job launch time with the actual job name, Schrödinger job ID, etc., for the job you are submitting. Variables that you can put in any new `template.sh` file are listed in Table D.1.

Table D.1. Batch script variables.

Variable	Variable action
`%NAME%`	Schrödinger job name, usually derived from your input file name.
`%DIR%`	Directory from which the job was submitted.
`%HOST%`	Machine from which the job was submitted.
`%USER%`	Name of the user who submitted the job.
`%JOBID%`	Job ID assigned by the Schrödinger job control system.
`%ENVIRONMENT%`	Commands that define the environment variables required for your job to run.
`%PRODUCT%`	Product name (NOT the executable).
`%APP_EXEC%`	The name of the exec variable for the product.
`%VER_ARGS%`	Version arguments.
`%JOBDB%`	The path to the job database.
`%NPROC%`	Number of processors that were requested from the queueing system for the execution of a single program. For distributed jobs, this is set to 1; for MPI parallel jobs, this is set to greater than 1.

Table D.1. Batch script variables. (Continued)

Variable	Variable action
%LOGDIR%	The directory in which log files are written.
%HOME%	Home directory on the submission host.
%COMMAND%	Command that launches the Schrödinger jmonitor program, which sets up, runs, and cleans up after your calculation.

The %ENVIRONMENT% and %COMMAND% lines are the only lines that are absolutely required in this script and they must appear in this order. These variables are assigned by the job control system and are not configurable by the user.

Another important component of this script is the two-line section that sets the SCHRODINGER_BATCHID environment variable to the actual batch ID assigned to this job. The batch ID is usually provided by the queueing system in a special environment variable such as the PBS_JOBID variable used by PBS. The jmonitor program checks for the SCHRODINGER_BATCHID environment variable and saves the batch ID in the job record, where the user can look it up.

If you want to run MPI parallel jobs, the list of host names assigned to the job by the queuing system must be made available in a file, and the SCHRODINGER_NODEFILE environment variable must be set in the script to point to this file. An example of this is shown in the PBS batch script above, in the if block above the %ENVIRONMENT% line.

D.4 The config File

The configuration file must contains settings for the keywords QPATH, QSUB, QDEL, QSTAT, and QPROFILE. These keywords define paths and commands that are used by the queueing software. As an example, $SCHRODINGER/queues/PBS/config contains the settings:

```
QPATH=/usr/local/pbs/bin
QSUB=qsub
QDEL=qdel
QSTAT=qstat
```

The QPROFILE keyword specifies the absolute path on the queue host of a configuration file that needs to be sourced to set up the environment to use the queue. This variable is useful for setting up an environment for the queuing system that does not affect the global environment.

Setting Up Remote Access Manually on Windows

This appendix describes how to manually set up access to remote hosts from Windows, rather than using the Remote Access Setup tool. This process involves running PuTTYgen, which is provided in the Schrödinger software installation in `%SCHRODINGER%\mmshare-v`*version*`\bin\WIN32-x86`, and uses `plink.exe` for connection, which is also provided in the Schrödinger software installation.

1. Double-click `puttygen.exe` in an explorer window or run `puttygen.exe` from the command line.

   The window that is displayed when PuTTYgen starts is shown in Figure E.1.

2. In the Parameters section, select the appropriate type of key to generate.

   Select SSH-1 (RSA) if the remote ssh server only supports `ssh` protocol 1.0. Otherwise, select SSH-2 RSA. Leave the number of bits in a generated key as 1024.

3. Click Generate, then move the pointer around in the blank area of the Key section.

   You **must** move the pointer around, as PuTTYgen uses these movements to generate a random key. When the key is generated, it appears in a noneditable text box at the top of this area, along with some other text boxes (see Figure E.2).

4. Copy and paste the key into the `authorized_keys` file in your `$HOME/.ssh` directory on the UNIX host. (You must copy and paste: do not use Save Public Key.) Append this public key to the `authorized_keys2` file if the `ssh` server supports only SSH 2.0 protocol.

   If your home directory is not cross-mounted on all desired UNIX hosts, you must copy and paste the key for each independent home directory.

   If you prefer, you can use PuTTY to connect to the UNIX host. A version of PuTTY is provided in the software installation in the subdirectory `mmshare-v`*version*`\bin\WIN32-x86`.

5. Make sure you do not have write permissions for group or others on your `$HOME` and `$HOME/.ssh` directory on the UNIX host:

   ```
   chmod go-w $HOME $HOME/.ssh $HOME/.ssh/authorized_keys*
   ```

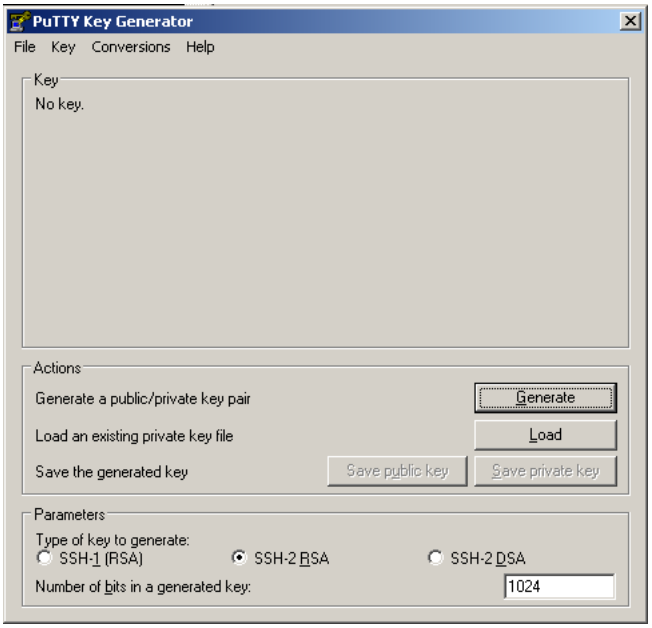

Figure E.1. PuTTYgen window before key generation.

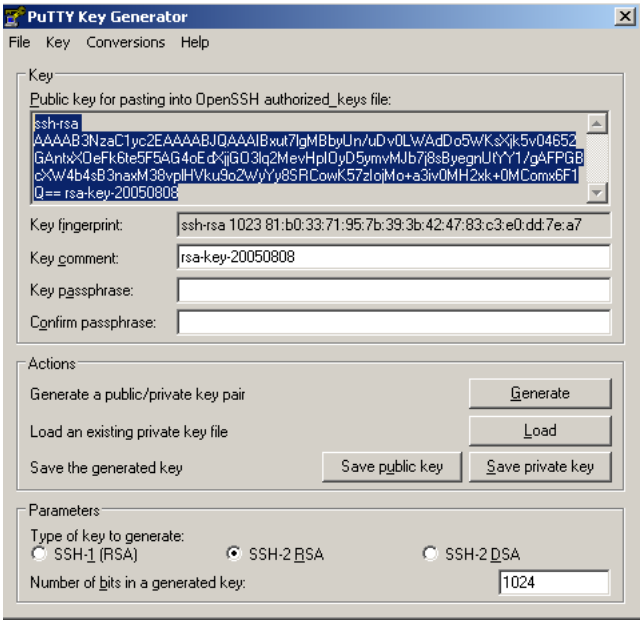

Figure E.2. PuTTYgen window after key generation with key selected.

6. In the PuTTYgen window, leave the Key passphrase and Confirm passphrase fields empty.

   Specifying the passphrase would defeat the purpose of this whole procedure, which is to avoid the password prompt.

7. Click Save private key.

8. Save the private key to `%HOME%\`*unixusername*`.ppk` if `%HOME%` is set, otherwise save it to `%USERPROFILE%\`*unixusername*`.ppk`, where *unixusername* is the user name on the Unix host.

   The extension must be included when you specify the location of the file.

   The default value of the `USERPROFILE` environment variable is `C:\Users\`*username*. To find out what `USERPROFILE` is set to, you can open a DOS window and enter the command `set`. If you do not save the private key to this location, you must set the environment variable `SCHRODINGER_SSH_IDENTITY` to the appropriate path.

   You must have a private key for each user name that you intend to use on remote hosts. It is recommended that you generate a private key for each user name. Repeat these instructions from Step 3 to Step 8 generate a new key for each user name.

9. Open a DOS window and run the following command once for each host to which you want to establish a connection without supplying a password.

   path-to-plink`\plink.exe -ssh` *hostname* `-i "`*path-to-ppk*`" -l` *username* `ls`

   `plink.exe` should be in the same location as `puttygen.exe`. If the ssh server on the remote host supports only SSH 2.0, use the following command instead:

   path-to-plink`\plink.exe -ssh -2` *hostname* `-i "`*path-to-ppk*`" -l` *username* `ls`

   The quotes around the path to the private key are required, and the `.ppk` extension must be included. This path is the one you specified in Step 8.

   If the remote host can be resolved on the network using both its short name and its fully qualified domain name, repeat the `plink` command for each version of the name, in order to cache the fingerprint for each name.

10. If prompted to save the RSA key for that host, choose `yes`.

    This choice ensures that in future the prompt is not displayed. The above command should then list the files in your home directory on the remote host.

For a more detailed explanation, go to the following web site:

http://the.earth.li/~sgtatham/putty/0.58/htmldoc/Chapter8.html#pubkey

Index

F

G

H

I

J

K

L

M

O

P

Q

R

S

T

U

V

W

120 West 45th Street
17th Floor
New York, NY 10036

101 SW Main Street
Suite 1300
Portland, OR 97204

245 First Street
Riverview II, 18th Floor
Cambridge, MA 02142

8910 University Center Lane
Suite 270
San Diego, CA 92122

155 Gibbs St
Suite 430
Rockville, MD 20850-0353

Dynamostraße 13
D-68165 Mannheim
Germany

Zeppelinstraße 73
D-81669 München
Germany

Potsdamer Platz 11
D-10785 Berlin
Germany

Quatro House
Frimley Road
Camberley GU16 7ER
United Kingdom

8F Pacific Century Place
1-11-1 Marunouchi
Chiyoda-ku, Tokyo 100-6208
Japan

No. 102, 4th Block
3rd Main Road, 3rd Stage
Sharada Colony
Basaveshwaranagar
Bangalore 560079, India

SCHRÖDINGER®