PGI® COMPILERS & TOOLS

INSTALLATION GUIDE FOR X86-64 CPUS AND TESLA GPUS

Version 2019

NVIDIA

TABLE OF CONTENTS

Chapter 1. Introduction..... 1
1.1. Product Overview..... 1
1.1.1. PGI Professional Edition..... 1
1.1.2. PGI Community Edition..... 1
1.1.3. PGI in the Cloud..... 1
1.2. Release Components..... 2
1.2.1. Additional Components..... 2
1.2.2. MPI Support..... 2
1.3. Terms and Definitions..... 2
1.4. Supported Processors..... 3
1.4.1. Supported Processors..... 3
1.5. Supported Operating Systems..... 4
1.6. Hyperthreading and Numa..... 5
1.7. Java Runtime Environment (JRE)..... 5
1.8. Product Support..... 5
Chapter 2. Installation Overview..... 7
2.1. Before You Begin..... 7
2.2. PGI Network Installations..... 8
2.3. Cluster Installation Overview..... 9
2.3.1. Cluster Configurations..... 9
2.3.2. Open Source Component Overview..... 10
2.4. Further Considerations..... 11
Chapter 3. Licensing..... 13
3.1. Licensing Terminology..... 13
3.2. Licensing Keys..... 13
3.2.1. Licensing Key Options..... 14
3.2.2. Licensing Key Capabilities..... 14
3.2.3. Licensing Key Comparisons..... 15
3.2.4. Licensing Key Dependencies..... 15
3.3. License File Overview..... 15
3.4. PGI License Setup Tool..... 16
3.5. Bundled License Key..... 16
3.6. PGI Licensing Considerations..... 17
3.7. The FlexNet License Manager..... 18
3.8. License Support..... 19
Chapter 4. PGI Installations on Linux..... 20
4.1. Prepare to Install on Linux..... 20
4.2. Installation Steps for Linux..... 21
4.3. End-user Environment Settings..... 29
4.3.1. PGI Compilers and Tools..... 29

4.3.2. Open MPI access..... 30
4.3.3. MVAPICH access..... 30
Chapter 5. Installations on macOS..... 31
5.1. Preparing to Install on Apple macOS..... 31
5.2. System Configuration to Enable Debugger 'Attach'..... 32
5.3. Installation Steps for macOS..... 33
5.4. End-user Environment Settings on macOS.....36
5.4.1. Use PGI Terminal Settings..... 37
5.4.2. Initialize your Terminal Shell Manually.....37
5.4.3. Use MPICH.....37
Chapter 6. Installations on Microsoft Windows.....38
6.1. Preparing to Install on Windows.....38
6.2. Installation Steps for Windows.....39
6.3. Customizing the Command Window.....42
6.4. PGI Default Installation Directories..... 42
6.4.1. Default Installation Directories..... 42
6.4.2. Tool Default Versions.....43
6.5. PGROUPD_LICENSE_FILE and FLEXLM_BATCH..... 43
6.5.1. PGROUPD_LICENSE_FILE.....43
6.5.2. FLEXLM_BATCH.....44
6.6. Windows Firewall and PGI Debugging.....44
6.6.1. Automatic Additions to Windows Firewall Exceptions List..... 44
6.7. Common Windows Installation Problems..... 44
Chapter 7. Contact Information..... 46

LIST OF FIGURES

Figure 1 PGI Installation Overview 8

Figure 2 Apple macOS Installation Overview 32

Figure 3 Welcome PGI Account Sample 35

LIST OF TABLES

Table 1 Processors Supported by PGI 2019 3

Table 2 License Key Comparisons 15

Table 3 Default Windows Installation Directories 42

Chapter 1. INTRODUCTION

Welcome to Release 2019 of PGI compilers and development tools for 64-bit x86-compatible processor-based workstations, servers, and clusters running versions of the Linux, Apple macOS, and Microsoft Windows operating systems.

This installation information applies to all PGI products.

1.1. Product Overview

PGI Compilers & Tools are available as PGI Professional Edition, PGI Community Edition and via select cloud platforms.

1.1.1. PGI Professional Edition

PGI Professional Edition is a for-fee, perpetual license providing access to the latest releases. Professional Edition offers technical support with frequent updates including feature enhancements, performance improvements, and bug fixes. Licenses are available in node-locked (single-system) and network floating configurations. See Licensing for more information on license types.

1.1.2. PGI Community Edition

PGI Community Edition is a no-cost license to a recent release. Issued one to two times per year, each release is valid for one year from date of release. For more information about Community Edition and differences with Professional Edition, see our Community Edition FAQ and Feature Comparison

1.1.3. PGI in the Cloud

PGI Community Edition for Linux is available on NVIDIA GPU Cloud (NGC) and Amazon Web Services (AWS) Marketplace. Both provide a pre-configured installation ready for use, as a container image on NGC and as an Amazon Machine Image on AWS.

1.2. Release Components

Release 2019 includes the following components:

- PGFORTRAN™ native OpenMP and OpenACC Fortran 2003 compiler.
- PGCC® native OpenMP and OpenACC ISO C11 and K&R C compiler.
- PGC++® native OpenMP and OpenACC ISO C++17 compiler.
- PGI Profiler® OpenACC, CUDA, OpenMP, and multi-thread graphical profiler.
- PGI Debugger® MPI, OpenMP, and multi-thread graphical debugger.
- Open MPI version 3.1.3 for 64-bit Linux including support for NVIDIA GPUDirect. Note that 64-bit linux86-64 MPI messages are limited to < 2 GB size each. As NVIDIA GPUDirect depends on InfiniBand support, Open MPI is also configured to use InfiniBand hardware if it is available on the system. InfiniBand support requires OFED 3.18 or later.
- ScaLAPACK 2.0.2 linear algebra math library for distributed-memory systems for use with Open MPI, MPICH or MVAPICH, and the PGI compilers on 64-bit Linux and macOS for Intel 64 or AMD64 CPU-based installations.
- Microsoft HPC Pack 2012 MS-MPI Redistributable Pack (version 4.1) for 64-bit development environments (Windows only).
- BLAS and LAPACK library based on the customized OpenBLAS project source.
- A UNIX-like shell environment for 64-bit Windows platforms.
- FlexNet license utilities.
- Documentation in man page format and online, pgicompilers.com/docs, in both HTML and PDF formats.

1.2.1. Additional Components

PGI floating license holders may download additional components for Linux from the PGI website including:

- MPICH MPI libraries
- MVAPICH2 MPI libraries

1.2.2. MPI Support

You can use PGI products to develop and debug MPI applications. The PGI debugger is MPI-enabled. PGI node-locked licenses support MPI debugging up to 16 local processes. PGI network floating licenses support MPI debugging up to 256 local or remote processes.

1.3. Terms and Definitions

This document contains a number of terms and definitions with which you may or may not be familiar. If you encounter an unfamiliar term in these notes, please refer to the PGI online glossary located at pgicompilers.com/definitions.

These two terms are used throughout the documentation to reflect groups of processors:

Intel 64
64-bit Intel x86-64 CPUs including Intel Core processors, Intel Xeon Nehalem, Sandy Bridge, Ivy Bridge, Haswell, Broadwell and Skylake processors, and Intel Xeon Phi Knights Landing.

AMD64
64-bit AMD™ x86-64 CPUs including Opteron and EPYC processors.

1.4. Supported Processors

The following table lists the processors on which Release 2019 of the PGI compilers and tools is supported. The table also includes the CPUs available and supported in multi-core versions.

The `-tp <target>` command-line option generates executables that utilize features and optimizations specific to a given CPU and operating system environment. Compilers in a 64-bit PGI installation can produce executables targeted to any 64-bit target, including cross-targeting for AMD64 and Intel 64-bit compatible CPUs.

The last release of PGI products to support 32-bit development was 16.10.

PGI products have the capability to generate binaries optimized for specific Intel or AMD processors. The PGI 2019 compilers can produce PGI Unified Binary™ object or executable files containing code streams fully optimized and supported for both Intel and AMD x64 CPUs and NVIDIA GPUs.

Tip

To determine what CPU you have, use one of the following commands:

```
pgfortran -V
```

or

```
pgcc -V
```

To produce PGI Unified Binary files, use the `-tp` command-line option, where <target> is any of the valid values in the supported processors table:

```
-tp <target1>,<target2>,<target3> ...
```

1.4.1. Supported Processors

Table 1 Processors Supported by PGI 2019

Brand	CPU	Target Processor (-tp <target> [,target...])
AMD		
	Zen architecture (EPYC, Ryzen)	zen

Brand	CPU	Target Processor (-tp <target> [,target...])
	Opteron Piledriver	piledriver
	Opteron Bulldozer	bulldozer
Intel		
	Skylake	skylake
	Knights Landing	knl
	Haswell	haswell
	Sandy Bridge	sandybridge
Generic		
	Generic	px

1.5. Supported Operating Systems

PGI supports Linux, macOS and Windows operating systems.

Linux

Most Linux operating systems with GLIBC 2.12 and newer. PGI tests and officially supports the following distributions; others may or may not work:

- CentOS 6.4 through 7.4
- Fedora 14 through 27
- openSUSE 13.2 through openSUSE Leap 42.3
- RHEL 6.4 through 7.4
- SLES 12 through SLES 12 SP 3
- Ubuntu 14.04, 16.04, 17.04, 17.10

Version differences in some operating system components may cause difficulties, but often these can be overcome with minor adjustments to the PGI software installation or operating system environment.

Apple macOS (formerly OS X)

The compilers, debugger, and profiler are supported on macOS versions 10.10.5 (Yosemite) through 10.13 (High Sierra).

Windows

- Windows Server 2008 R2
- Windows 7
- Windows 8.1

- Windows 10
- Windows Server 2012
- Windows Server 2016

The PGI compatibility page, https://www.pgroup.com/products/index.htm?tab=compat lists any new operating system distributions that may be explicitly supported by the PGI compilers. If your operating system is newer than any of the operating systems in the preceding list, the installation may still be successful.

1.6. Hyperthreading and Numa

Most modern operating systems include support for Intel Hyper-threading (HT). Further, most modern Linux distributions support the *Native Posix Threads Library (NPTL)*. Parallel executables generated using the *OpenMP* and auto-parallelization features of the PGI compilers will automatically make use of NPTL on distributions where it is available. In addition, the PGI debugger is capable of debugging executables built using either NPTL or earlier `pthread` implementations.

Many modern multi-socket AMD Opteron processor-based servers use a *NUMA (Non-Uniform Memory Access)* architecture in which the memory latency from a given processor to a given portion of memory can vary. Newer Linux distributions, including SuSE 11/13 and SLES 10/11, include NUMA libraries that can be leveraged by a compiler and associated runtime libraries to optimize placement of data in memory.

1.7. Java Runtime Environment (JRE)

The PGI debugger and profiler use Java. To use these tools, a version of the Java Runtime Environment (JRE) must be installed.

The tools are compatible with Java 1.6 and later versions. The PGI 19.3 release includes the JRE version 1.8 update 112. On Linux and macOS, this version of the JRE is installed in the same location as the compilers and does not affect the default Java. On Windows, the JRE is installed to Java's default install location.

One can opt out of the JRE installation on both Linux and Windows. Without Java, the debugger and profiler are only available in command-line mode.

1.8. Product Support

All new PGI licenses include free PGI Support Service during the money-back guarantee period.

The PGI Support Service provides access to techincal support and other benefits including:

- Technical support requests may be sent in a number of ways:

- Faxed to +1-503-682-2637
- By using the online support request form located at pgicompilers.com/support-request.

Phone support is not currently available.

- Release upgrades for licensed product(s) at no additional cost, except for any administrative fee that may apply.
- Notification by email when maintenance releases occur and are available for download and installation.
- Full license fee credits on Product upgrades, except for any administrative fee that may apply. "Product upgrades" refer to exchanging one Product license for a more expensive Product license, and is not the same as a Version or Release upgrade previously referenced.
- Full license fee credits on user-count upgrades, except for any administrative fee that may apply.

Important To continue receiving these benefits after the money-back guarantee period days, you can purchase an extension to your PGI Support Service. Extensions are available in yearly increments.

Contact sales@pgroup.com if you would like information regarding the support service for the PGI products you have purchased.

Chapter 2. INSTALLATION OVERVIEW

This chapter provides an overview of the steps required to successfully install PGI compilers and tools. The remaining chapters provide the details of each of the steps. Specifically, the Licensing section describes licensing. PGI Installations on Linux describes how to install PGI in a generic manner on Linux, including how to install and run a FlexNet license daemon on Linux. Installations on Microsoft Windows describes how to install on a Windows system, and Installations on macOS describes how to install on an Apple macOS system.

2.1. Before You Begin

Before you begin the installation, it is advantageous to understand the flow of the installation process. There are three stages of the process:

- Prepare to install—verifying that you have all the required information, that the correct PGI software is downloaded, and any other non-PGI software that is needed is available and/or installed.
- Install the software—installing the software appropriate for your operating system.
- License the software—generating of license keys using the PGI website, installation of the license keys, and starting the license server.

The following illustration provides a high-level overview of the PGI installation process.

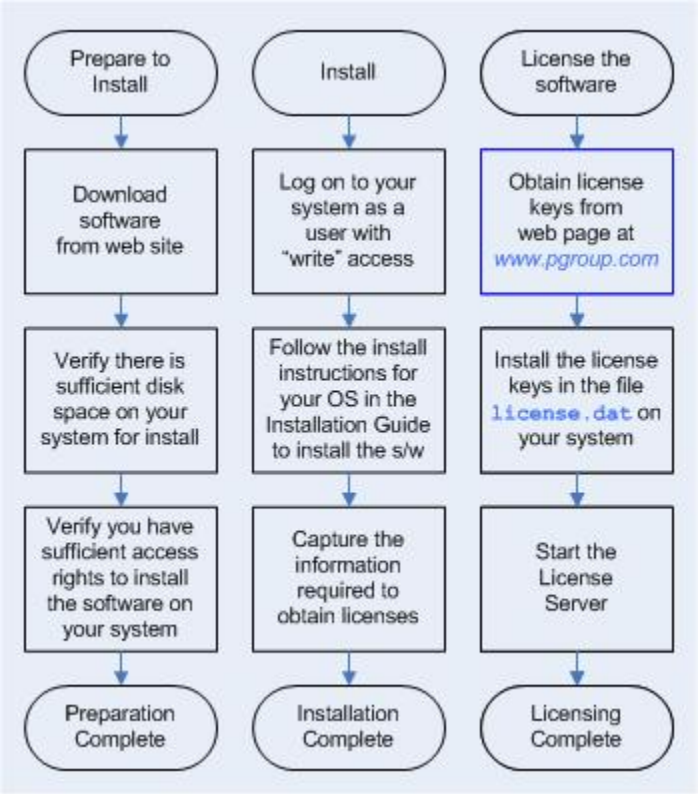

Figure 1 PGI Installation Overview

For more complete information on these steps and the specific actions to take for your operating system, refer to the remainder of this document.

2.2. PGI Network Installations

Linux

On Linux, a PGI network floating license may be installed locally on each machine on a network or it may be installed once on a shared file system available to each machine. If you select the second method, then after the first installation you can run a simple script on each machine in the group to add each new platform to the family of machines using the common compiler installation.

For more information on network installations, refer to PGI Installations on Linux.

Apple macOS and Windows

On Apple macOS and Windows, PGI software must be installed locally on each machine. A network installation using a shared file system is not supported.

2.3. Cluster Installation Overview

The following sections contain the information needed for you to successfully prepare to install the PGI software for operation in a cluster environment.

Recall that a cluster is a collection of compatible computers connected by a network. The PGI software can be installed on a single node, and the node can be treated as if it is a cluster.

Support for cluster programming is limited to clusters combining Intel 64 or AMD64 CPU-based systems.

For multi-process programming of message-passing applications that execute on a cluster, PGI includes a 64-bit set of Open MPI libraries. These libraries implement the MPI inter-process communication standard.

Additional MPI libraries are available from the PGI website at pgicompilers.com/downloads.

2.3.1. Cluster Configurations

Clusters are typically configured with two types of nodes:

- One "master" node from which jobs are launched
- One or more "slave" nodes that are used only for computation

PGI compilers and tools should be installed on the master node, and installed to run from the master node only.

Typically, the master node is accessible from the general-purpose or "public" network and shares a file system with the other computers on your network using NFS. The master node and all of the slave nodes are interconnected using a second "private" network that is only accessible from computers that are part of the cluster.

There are two common cluster configurations:

1. The master node is used only for compilation and job submission, and only the slave nodes are used for computation.
2. All nodes are used for computation, including the master node.

One way to use MPICH in the first configuration is to manage job scheduling. By default, the `mpirun` command uses the master node as one of the computation nodes. It is possible to exclude the master node as a computation node in the second configuration if `mpirun` is invoked with the `-nolocal` option. For more information, refer to the man page for `mpirun`.

Typically, a master node has two network cards to allow communication to the outside network as well as to the cluster nodes themselves, which may be on their own subnet. The installation script will prompt you for the name of the master node. If your cluster has two network cards, then you should enter the name of the network card that is connected to the same network as the cluster nodes.

For MPICH to run correctly, access from each node to every other node must be available via the `rsh` or `ssh` command. For example, if a three-node cluster consists of a master, named *master*, and two slaves named *node1* and *node2*, then from *node1* as a user you should be able to issue the commands:

```
% rsh master date
% rsh node2 date
```

or

```
% ssh master date
% ssh node2 date
```

You can issue similar commands from *node2* using *master* and *node1*.

By default, all of the PGI compilers and tools will be installed on your system. You will select which of the open-source components to install.

At this point, before you start the installation, you must determine:

- Which open-source components—MPICH, MVAPICH2 and/or Open MPI—are best suited for cluster. All three are available.
- The hostnames of all the nodes that will be included in your cluster. You need a list of these during the installation.
- The type of cluster configuration—that is, whether the master node will participate as a compute node or will be strictly a front-end for compilation, job launching, and so on.
- Whether the compute nodes can share files with the master node, which is strongly recommended.
- The location of a commonly mounted disk that all nodes can access via the same pathname. This disk is very helpful for installing runtime libraries and storing executables created.

PGI Installations on Linux describes how to install the PGI Fortran, C and C++ compilers and tools on Linux using the *install_cdk* script from PGI.

The PGI compilers and tools are license-managed, which is described in Licensing. Further, Installation Steps for Linux provides specific information about how to use your personalized account to generate permanent license keys.

2.3.2. Open Source Component Overview

The PGI Compiler User's Guide, www.pgroup.com/resources/docs/19.3/pdf/pgi19ug-x86.pdf describes basic usage of the PGI open-source components, including the MPICH and ScaLAPACK libraries, and the example benchmark programs and tutorials.

MPICH, MVAPICH, Open MPI, and ScaLAPACK are all open-source software packages that are not formally supported by PGI. Support for these products is generally provided by their respective user communities, which you can learn more about at the following URLs:

- MPICH3—http://www.mpich.org

- MVAPICH2—http://mvapich.cse.ohio-state.edu/ contains a wealth of information, including online documentation, tutorials, FAQ files, patch distributions, and information on how to submit bug reports to its developers.

To use the PGI version of MVAPICH2, the prerequisite OpenFabrics (OFED) software must be installed. For details on OFED, refer to http://www.openfabrics.org.

- Open MPI—https://www.open-mpi.org is a source for Open MPI information about building and using the library.
- ScaLAPACK—http://www.netlib.org/scalapack contains FAQ files and current distributions of ScaLAPACK.

2.4. Further Considerations

There are a number of additional things you might consider prior to installation.

- A PGI Account

  PGI recommends that everyone installing or using the compilers create an account on the PGI website, pgicompilers.com/register. An account is required to manage your license, download PGI software, post on the PGI userforum, submit technical support requests, view some web content and other activities.

  Once you have a web account, your account should be tied to any PIN that you will use to license the compilers. A PIN is connected to permanent license, as described in Licensing.

- Downloading the proper package

  When ready, you can download the proper package(s) at pgicompilers.com/download. The file sizes can be large. To help avoid problems, a checksum is displayed for all downloaded files. After downloading `file.tar.gz`, use the command `md5sum file.tar.gz` to display the checksum value. The two values should match.

- Permissions

  You will need root, sudo, or Administrator access to install the compilers and/or to set up the license service for auto restart. File ownership or permission issues are a common source of unexpected behavior.

- Firewall Issues

  Systems with firewalls may encounter two problems. First, the hostname used in the license file may not be mapped properly to the IP address of your machine. Second, the license server (lmgrd) and license daemon (pgroupd) each need an assigned port to allow them access through the firewall. For lmgrd, the default port is 27000. For

pgroupd, there is no default port assigned. With a firewall, you may need to change and/or specifically assign another port number.

- Laptop Installations

  For laptop installations, consider using localhost as the hostname, mapped to the loopback interface 127.0.0.1. Test this by pinging localhost. Also consider looking for hostids (lmutil lmhostid) in all three laptop environments (network, wi-fi, on none), and look for hostids persistent in all three environments.

Chapter 3. LICENSING

The PGI compilers and tools use the FlexNet Publisher (FNP) license management system from Flexera Software.

PGI software licensing uses the FlexNet Publisher (FNP) license management system from Flexera Software.

3.1. Licensing Terminology

The PGI compilers and tools are license-managed. Before discussing licensing, it is useful to have common terminology.

- **License** – the right to use PGI compilers and tools as defined by the End-user License Agreement (EULA), this is a legal agreement between NVIDIA and PGI end-users. PGI Professional (for-fee, perpetual) licenses are identified by a Product Identification Number (PIN - see below). You can find a copy of the EULA on the PGI website , pgicompilers.com/LICENSE, and in the `$PGI/<platform>/<rel_number>/doc` directory of every PGI software installation.
- **License keys** – ASCII text strings that enable use of the PGI software and are intended to enforce the terms of the License. For PGI Professional, License keys are generated by each PGI end-user on the PGI website using a unique hostid and are typically stored in a file called `license.dat` that is accessible to the systems for which the PGI software is licensed.
- **PIN** – Product Identification Number, a unique 6-digit number associated with a PGI Professional license. This PIN is included in your order confirmation. The PIN can also be found in your license key file after **`VENDOR_STRING=`**.
- **PIN tie code** – A unique 16-digit number associated with each license (PIN) that allows others to "tie" that license to their PGI user account, pgicompilers.com/account for administrative purposes. PGI Professional licensees can use their PIN tie code to share license administration capabilies with others in their orgaization.

3.2. Licensing Keys

3.2.1. Licensing Key Options

PGI products on x86-64 use Flexera's FlexNet license management software, and may require FlexNet license daemons running on your local system or a separate license server.

There are three types of license keys: starter, community, and permanent.

Starter License Keys
Time-limited evaluation license keys bundled with most releases or provided on request by PGI. Starter license keys do not use FlexNet daemons.

Community License Keys
A PGI Community license key is bundled with PGI Community Edition releases, valid only for that release and for one year from the date of release. Community license keys do not use FlexNet daemons.

Permanent License Keys
Purchased PGI licenses allow users to create *permanent* license keys. Permanent license keys are perpetual, meaning they remain valid indefinitely with eligible versions of PGI software. Permanent license keys require FlexNet license daemons and are available in either node-locked (local system only) or network floating (server) configurations:

- **Node-locked**, created by a user. FlexNet daemons are installed on the same system as the compilers, and use is restricted to that system. Any user on the system is allowed to use the compilers, but only a single user at a time.
- **Networking floating**, created by a user. FlexNet daemons are installed on a networked license server system, allowing for concurrent use (compilation) by multiple networked client systems, up to the number of licensed seats purchased. The license server for network floating licenses can run on either Linux or Windows.

3.2.2. Licensing Key Capabilities

At the conclusion of the Starter license key evaluation interval, the PGI compilers and tools, along with any executable files generated with them, will cease to function. To retain functionality, any executables, object files, or libraries created using the PGI compilers with starter license keys must be recompiled with either a community or permanent license key. See Licensing Key Comparisons below for details.

Executable files generated using community or permanent license keys are unconstrained, and run on any compatible system regardless of whether the PGI compilers are installed.

3.2.3. Licensing Key Comparisons

Table 2 License Key Comparisons

License Key Type	Starter	Community	Permanent Node-locked	Permanent Network Floating
Limited duration	Yes - 30-90 days	Yes - 1 year	No - Perpetual	No - Perpetual
Time-limited executables	Yes	No	No	No
FlexNet daemons required	No	No	Yes	Yes
Concurrent users	No	No	No	Yes

3.2.4. Licensing Key Dependencies

When using permanent license keys, you must install the PGI software before obtaining your license keys. The license key generation process requires information generated during software installation. Starter and community license keys do not have this dependency.

Important If you change the configuration of your system by adding or removing hardware, permanent license keys may become invalid. You will need to obtain new license keys from pgicompilers.com or using the license setup tool described below.

3.3. License File Overview

A license file is broken down into four sections: **SERVER**, **DAEMON**, **PACKAGE**, and **INCREMENT**. The following snippet gives an overview of each section and its relevant fields. The italicized parts are optional.

```
SERVER <license server hostname> <your hostid> 27000
DAEMON pgroupd "/path/to/pgroupd" PORT=port_number
PACKAGE PGI<release>-<PGI PIN> pgroupd <support end date> COMPONENTS= \
...
INCREMENT PGI<release>-<PGI PIN> pgroupd <support end date> permanent <# of
 seats> \
VENDOR_STRING=<PGI PIN>:16:ws:accel HOSTID=<your hostid>
...
```

The **SERVER** line has three components, the hostname of the license server, the hostid of the license server, and the **PORT** used by lmgrd to process the license requests. You can edit the hostname and **PORT** used (27000 default) by hand without regenerating the license.

The **DAEMON** line has three components, the name of the **DAEMON** used (pgroupd), the optional path to the daemon if not where lmgrd is located (as in `/usr/pgi/daemon/`

pgroupd) and an optional **PORT** which pgroupd uses to communicate. The **PORT** can be any unused integer that the operating system allows, which you can change by hand in the license file. If the **PORT** is unspecified pgroupd will randomly pick one between 1024 and 65535.

The next section begins with a line starting with **PACKAGE**, listing all available component features for the given license. This is followed by a section with a line beginning with **INCREMENT**. Both the **PACKAGE** and **INCREMENT** lines contain a date-formatted number like 2018.1231. This number designates when support service expires and with which releases the license will work. In this case, support service for the license expires on December 31 2018, and all versions released on or prior to that date are supported by this license key. The license key does not need to be updated until support service is renewed with a later date of expiration.

3.4. PGI License Setup Tool

PGI 2019 for Linux x86-64 and Windows includes the PGI License Setup Tool to help automate your permanent license key retrieval and installation process. For Linux, this tool is normally installed at $PGI/<platform>/<rel_number>/bin/pgi_license_tool.

In order for the FlexNet daemons to work properly, on Linux platforms, they require the Linux Standard Base (LSB) linker/loader helper library to exist at /lib64/ld-lsb-x86-64.so.3. The library may already exist on your system, in which case no further action is required. If no such library exists, the license setup tool will search for a compatible linker library and attempt to establish a symlink. Running the script without sufficient privileges (e.g. root access) will cause the script to error out.

You can perform symlink step manually with the following commands:

```
$ sudo ln -s /lib64/ld-linux-x86-64.so.2 /lib64/ld-lsb-x86-64.so.3
```

or

```
$ sudo ln -s /lib/x86_64-linux-gnu/ld-linux-x86-64.so.2 /lib64/ld-lsb-
x86-64.so.3
```

You may alternatively install the full LSB package for your distribution, but know that the LSB package installs other components that are not required by FlexNet or PGI compilers.

- Debian/Ubuntu: [sudo] apt-get install lsb
- RedHat/Fedora/CentOS: [sudo] yum install redhat-lsb
- SLES/OpenSUSE: [sudo] yast --install lsb

For more information on using the PGI License Setup Tool on Windows, refer to Installation Steps for Windows.

3.5. Bundled License Key

Installation may place a temporary license key file named license.dat in the PGI installation directory if no such file already exists.

If you use a separate license server, for example `LM_LICENSE_FILE=`*port@server.domain.com*, that supports this version, it is recommended that you remove or rename the license key file in the installation directory.

3.6. PGI Licensing Considerations

PGI licensing is the common method used by all PGI products to control access. The License Service is made up of two components:

- The `lmgrd` daemon is the common license manager component used by all FlexNet licenses. It oversees the distribution of *license seats*.
- The `pgroupd` daemon is unique to PGI and reads and decodes the PGI license file.

A license is created using the *hostname* and the *hostid* of the license server. These rules apply:

- Only one copy of `lmgrd` runs on the license server, and is used by all FlexNet-type licenses. For both floating or node-locked licenses, only a single license server is required.
- Only one PGI license per license server. You may have FlexNet licenses for other non-PGI software packages, but you may only have one PGI license per server. More than one causes the license server to malfunction.
- The *hostname* used in the license file should be understood by all of the systems using the compilers. Entering the command `ping hostname` should result in the same IP address on all of the systems, including the license server itself.
- The *hostid* is usually the MAC (inet address) of a device (`/sbin/ifconfig` on Linux and macOS, `ipconfig /all` on Windows), which can be detected by the FlexNet utilities. It is usually best to choose the MAC associated with the IP address of the server.
- PGI node-locked licenses are limited to allow the compilers to work only on the machine running the license service.
- PGI network floating licenses allow any machine that can communicate with the license server in a timely manner (i.e. distance relative) to run the compilers.
- On clusters, the PGI installer does not install compilers on the ‘slave’ nodes. The ‘master’ node usually is on both the private and public networks, so the master node is the only node that needs access to the license service. The master node may even be the license server. Multiple clusters can share a single PGI floating license by having all of the master nodes communicate with the license service. A PGI network floating license allows the PGI debugger and the PGI profiler to run distributed across all of the cluster nodes.
- There is a sequence in which the PGI compilers look for environment variables.
  - The PGI compilers look first at the environment variable `$PGROUPD_LICENSE_FILE` for the location of the PGI license. This variable is set in the registry on Windows machines, and is specific to PGI products.

  - The PGI compilers next look at the environment variable `$LM_LICENSE_FILE` for the location of the PGI license. This is a generic variable used by every FlexNet licensed product.
  - The PGI compilers then look at the default location for `$PGI/license.dat`.
- On the license server itself, `$PGROUPD_LICENSE_FILE` and/or `$LM_LICENSE_FILE` must be the full license pathname, as in `/opt/pgi/license.dat` on Linux and macOS, or `C:\Program Files\PGI\license.dat` on Windows.
- On machines other than the license server, `$PGROUPD_LICENSE_FILE` and/or `$LM_LICENSE_FILE` can either be a full pathname to the license file, or *port@host* such as 27000@hostname.
- When changing a license file, take care to ensure the license software can read the new license. On Windows and macOS, this means having Admin privileges or *sudo* capability to copy the license file to its destination. If the license service is already running, you may need to STOP and START it again to make sure the new file is read and not the old file that is stored in cache.
- When the FlexNet license service detects a problem or an event, it records it in either the `flexlm.log` file (Linux) or the `License.log` file (Windows, macOS). If you're having licensing problems, check this file for information.

3.7. The FlexNet License Manager

PGI products on x86-64 use the FlexNet software license management system from Flexera Software.

As part of the installation process, PGI Professional Edition users can install and configure the FlexNet license management software. There are two permanent licensing options using FlexNet licensing.

- **Node-locked** allows any user on a single system to run the software, with access restricted to that prespecified machine.
- **Network floating license** supports a more flexible licensing system.
  - There is one license server on a network and any system on that network with a properly configured version of PGI Software installed can run the software.
  - Multiple users can use PGI Software simultaneously from multiple systems on the network.
  - The limitation on the number of user+system combinations is determined by the number of seats that were purchased with the license.

The instructions in the following sections of this guide describe how to configure license daemons for Linux, Windows, or macOS, including installation and start-up of the license services, and proper initialization of the `LM_LICENSE_FILE` and, for Windows, `FLEXLM_BATCH` environment variables.

3.8. License Support

All new PGI licenses include free PGI Support Service during the money-back guarantee period. For more information about this service and how to extend it, refer to Product Support.

Chapter 4.
PGI INSTALLATIONS ON LINUX

This section describes how to install PGI products in a generic manner on a Linux system. It covers local and network installations.

4.1. Prepare to Install on Linux

Linux installations require some version of the GNU Compiler Collection (gcc) to be installed and in your `$PATH` prior to installing PGI software. For PGI compilers to produce 64-bit executables, a 64-bit gcc compiler must be present. For C++ compiling and linking, ensure the same is true for `g++`. To determine if such a compiler is installed on your system, do the following:

1. Create a `hello.c` program.

```
#include <stdio.h>
int main() {
  printf(" hello\n");
}
```

2. Compile with the `-m64` option to create a 64-bit executable.

```
$ gcc -m64 -o hello_64 hello.c
```

   Run the `file` command on the produced executable. The output should look similar to the following:

```
$ file ./hello_64
hello_64: ELF 64-bit LSB executable, AMD x86-64, version 1 (SYSV), for
GNU/Linux 2.6.9, dynamically linked (uses shared libs), for GNU/Linux
2.6.9, not stripped
```

Any changes to your gcc compilers requires you to reinstall your PGI compilers.

For cluster installations, access to all the nodes is required. In addition, you should be able to connect between nodes using `rsh` or `ssh`, including to/from the same node you are on. The hostnames for each node should be the same as those in the cluster machine list for the system (`machines.LINUX` file).

In a typical local installation, the default installation base directory is `/opt/pgi`.

If you choose to perform a network installation, you should specify:

- A shared file system for the installation base directory. All systems using the compilers should use a common pathname.
- A second directory name that is local to each of the systems where the PGI compilers and tools are used. This local directory contains the libraries to use when compiling and running on that machine. Use the same pathname on every system, and point to a private (i.e. non-shared) directory location.

This directory selection approach allows a network installation to support a network of machines running different versions of Linux. If all the platforms are identical, the shared installation location can perform a standard installation that all can use.

To Prepare for the Installation:

- After downloading the PGI installation package, bring up a shell command window on your system.

  The installation instructions assume you are using csh, sh, ksh, bash, or some compatible shell. If you are using a shell that is not compatible with one of these shells, appropriate modifications are necessary when setting environment variables.
- Verify you have enough free disk space for the PGI installation.
  - The uncompressed installation packages require 1.2 GB of total free disk space.
  - The linux86-64 platform requires up to 1.54 GB of free disk space, depending on the number of packages installed.
- If you plan to create a permanent license, your computer must be able to access the Internet. You also need to know your IP address and whatever hostname is mapped to that IP address.

  If this computer is behind a firewall at your site, make sure it can access the Internet.

  If a proxy server is used, you need this additional information:
  - The address (URL) of the proxy server.
  - Whether the proxy server requires authentication and if so, the username and password.
- To generate license keys, log in to your account on the PGI website at pgicompilers.com/login. From this page, you may also create an account or reset your password.

4.2. Installation Steps for Linux

Follow these instructions to install the software:

1. Unpack the PGI software.

In the instructions that follow, replace `<tarfile>` with the name of the file that you downloaded.

Use the following command sequence to unpack the tar file before installation.

```
% tar xpfz <tarfile>.tar.gz
```

2. Run the installation script(s).

Install the compilers by running **`[sudo] ./install`** from the directory where you unpacked the tar file.

For *PGI CDK*, after the compilers are installed, install the additional CDK components by running the **`install_cdk`** script. While you do not need root-level access permission to install the PGI CDK, you must have permissions to create and copy files on the cluster nodes.

Important The installation script must run to completion to properly install the software.

To successfully run this script to completion, be prepared to do the following:

- Consent to the PGI End-User License Agreement (EULA) available to preview at **`install_components/common/LICENSE.txt`**.
- If you are installing a PGI network floating license, determine whether to perform a local installation or a network installation.
- Determine whether to install the optional components:

  Java Runtime Environment (JRE)
  Open MPI library
- Define where to place the installation directory. The default is `/opt/pgi`.
- Determine whether to use the built-in utility to generate license keys.

Linux users now have the option of automating the installation of the PGI compiler suite without interacting with the usual prompts. This may be useful in a large institutional setting, for example, where automated installation of PGI compilers over many systems could be efficiently done with a script.

To enable the silent installation feature, set the appropriate environment variables prior to running the installation script. These variables are as follows:

`PGI_SILENT`	(required) Set this variable to "true" to enable silent installation.
`PGI_ACCEPT_EULA`	(required) Set this variable to "accept" to indicate that the user has accepted the PGI EULA and all applicable third-party EULAs; these EULAs are located in the **`install_components/common/`** subdirectory in the directory where the PGI installation tar file was uncompressed and untarred.
`PGI_INSTALL_DIR`	(required) Set this variable to a string containing the desired installation location, e.g. **`/opt/pgi`**.
`PGI_INSTALL_TYPE`	(required) Set this variable to select the type of install. The accepted values are "single" for a single system install or "network" for a network install.

PGI_INSTALL_LOCAL_DIR	(required for network install) Set this variable to a string containing the path to a local file system when choosing a network install.
PGI_INSTALL_NVIDIA	(optional) Set this variable to "false" to not install the CUDA Toolkit components. If a supported version of CUDA is already installed, this option can be used to save disk space.
PGI_INSTALL_JAVA	(optional) Set this variable to "true" to enable the installation of the Java Runtime Environment. The JRE is required for the PGI debugger and PGI profiler. If a JRE is already present on the system, it is not necessary to reinstall.
PGI_INSTALL_MPI	(optional) Set this variable to "true" to enable the installation of the Open MPI environment.
PGI_MPI_GPU_SUPPORT	(optional) Set this variable to "true" to enable NVIDIA GPU support in the Open MPI environment.

The PGI installation scripts install all of the binaries for the PGI compilers, tools, and libraries in the appropriate subdirectories within the specified installation directory. This main directory is defined during installation as $PGI and the $PGI environment variable should be set to the path of this directory before using the compilers.

Important If you choose not to generate license keys at installation time, record the FlexNet hostid and hostname that are echoed to the screen. This information is also saved to the file `/opt/pgi/license.info`, where `/opt/pgi` is the installation directory.

The PGI installation script asks if you wish to enable NVIDIA GPU support in Open MPI. If you choose to disable GPU support in Open MPI at installation time, then the installation script writes the following environment variable setting to `/opt/pgi/linux86-64/2019/mpi/openmpi-2.1.2/bin/env.sh`, where `/opt/pgi` is the installation directory:

```
export OMPI_MCA_mpi_cuda_support=0
```

Should you decide at a later time to enable NVIDIA GPU support in Open MPI, you may do so by simply commenting out this line in the file by inserting a **#** character at the beginning of the line.

3. Make PGI products accessible.

   When the installation script has completed, execute the following commands to make the PGI products accessible and to initialize your environment for use by FlexNet.

Each user must issue the following sequence of commands to initialize the shell environment prior to using the PGI compilers and tools.

Assuming you have installed in the default `/opt/pgi` directory, execute the following commands.

In csh, use these commands:

```
% setenv PGI /opt/pgi
% set path=(/opt/pgi/linux86-64/19.3/bin $path)
% setenv MANPATH "$MANPATH":/opt/pgi/linux86-64/19.3/man
```

```
% setenv LM_LICENSE_FILE /opt/pgi/license.dat:"$LM_LICENSE_FILE"
```

In bash, sh, or ksh, use these commands:

```
$ export PGI=/opt/pgi;
$ export PATH=/opt/pgi/linux86-64/19.3/bin:$PATH;
$ export MANPATH=$MANPATH:/opt/pgi/linux86-64/19.3/man;
$ export LM_LICENSE_FILE=$LM_LICENSE_FILE:/opt/pgi/license.dat;
```

Tip You should add these commands to your shell startup files to ensure that you have access to the PGI products in future login sessions.

4. Verify the release number of the installed software. It should say 19.3.

   To verify the release number of the products you have installed, use the `-V` option on any of the compiler commands, as illustrated in the following examples. If you use `-v` instead, you can also see the sequence of steps the compiler uses to compile and link programs for execution on your system.

For Fortran 77, use:	**pgf77 -V x.f**
For Fortran 2003, use:	**pgfortran -V x.f**
For C++, use:	**pgc++ -V x.cpp**
For ANSI C, use:	**pgcc -V x.c**

These commands can be successfully executed even if the files `x.f` or `x.c` do not exist and you have not completed the licensing phase of the installation. Use them to check that you have installed the proper version of the compilers and have initialized your environment to enable access to that version.

5. Generate and install license keys.

This step is necessary only if you chose not to allow the installation script to perform these tasks for you. Alternatively, you may use the PGI License Setup Tool, normally installed at `$PGI/<platform>/<rel_number>/bin/pgi_license_tool`.

Most PGI products are license-managed using FlexNet licensing. This system requires that you possess a valid license key file for the licensed product to operate. Most PGI products include temporary license keys with the installation package. Permanent license keys are available from the PGI website, https://www.pgroup.com/license/pin_manage.php?view=keys.

Other components, such as Open MPI, are open-source products that are not license-managed.

Any temporary license keys included with your package will be installed automatically. You can verify this by confirming the existance of the `license.dat` file in your PGI installation directory. If so, you may skip to the next section.

To obtain your permanent license key, you need the following information:

- An account on the PGI website. You probably created this account when you downloaded the PGI software.

Tip The username (email address) and password required to connect to the pgicompilers.com website are the same ones you used to download the installation software from the web site.

  - If you purchased a license without creating an account, one was created for you when your order was processed. Please check for an activation email from sales-noreply@pgroup.com.
  - If you don't have an account, you can create one at: pgicompilers.com/register.
- The FlexNet *hostid* and *hostname* of the computer on which the software is installed. The installer echoes this information to your screen.

You can also obtain your FlexNet hostid by using the following command after you have installed the products and initialized the environment variables:

```
% lmutil lmhostid
```

You should see a message similar to the following message with one or more hostids displayed.

```
The FlexNet host ID of this machine is "12345678abcd edcba9876543".
```

In this example, you can use either `12345678abcd` or `edcba9876543`, but not both, as the hostid.

Hostids come from configured network cards. If you use your computer in multiple network environments, you may need to run the following command in each environment to see which hostids are configured. Then, to reduce potential license problems, choose the hostid that occurs in all your environments.

Install License Keys

Once you have generated your permanent license key, copy and paste it into the file: `/opt/pgi/license.dat`, substituting the appropriate installation directory path if you have not installed in the default `/opt/pgi` directory.

The `license.dat` depends on the product you are installing. However, each `license.dat` file looks somewhat similar to the following sample file from a PGI installation:

```
SERVER <hostname> <hostid> 27000
DAEMON pgroupd
PACKAGE PGI2017-<PGI_PIN> pgroupd <support end date> \
<...>
6167 7015 3F05 9C37 2315 ACDF 1B73 DAA9 FBAE"
```

In your license file:

- `<hostid>` should match the *hostid* you submitted above when you generated your license keys. You should see it on the SERVER line.

- If necessary, you can enter or edit the <hostname> entry manually, but you cannot edit the <hostid> entry or you will invalidate the license keys.
- <support end date> represents the date your support expires. For example, if your support end date for your PGI PIN (Product Identification Number) is August 1, 2017, then the date in your file is 2017.0801. For information on how to renew your support contract, refer to Product Support.
- <PGI_PIN> represents the six-digit product identification number for this license.

Please include your PIN when contacting PGI for technical support for the products you have purchased. This PIN is also in your order confirmation email.

6. Start the license manager daemon.

Important If you are using a PGI Community Edition license key, evaluating PGI software with a starter license key, or if you used the PGI installation script, you do not need to perform this step and can proceed to Step 7.

Installations in a directory other than the default `/opt/pgi`

Perform the following steps if you installed the compilers in a directory other than `/opt/pgi`:

1. Edit the shell script template `$PGI/linux86-64/19.3/bin/lmgrd.rc`.
2. Substitute the correct installation directory for `/opt/pgi` in the section of the script entitled *Where to find the PGI Software*.
3. Save the file and exit the editor.

Issue the following command to start the license server and `pgroupd` license daemon running on your system:

```
% cd $PGI/linux86-64/19.3/bin/
% ./lmgrd.rc start
```

Tip To use the more restrictive access of invoking `lmgrd` with the `-local` switch, refer to the `lmgrd.rc` file for a description of the `-local` switch and how to use it.

If you wish to stop the license server and `pgroupd` license daemon at a later time, you can do so with the command:

```
% cd $PGI/linux86-64/19.3/bin/
% ./lmgrd.rc stop
```

Start license server upon reboot:

To start the license server and `pgroupd` license daemon each time your system is booted:

1. Log in as root.

You **must** be logged in as root to successfully execute these commands.

2. Verify you have set the PGI environment variable as described in Step 3 of this installation process.
3. Execute the following two commands:

```
% cp $PGI/linux86-64/19.3/bin/lmgrd.rc  /etc/init.d/lmgrd
% ln -s /etc/init.d/lmgrd  /etc/rc.d/rc3.d/S90lmgrd
```

There are two values in this example that may be different on your system:

- Your `rc` files may be in a directory other than the one in the example: `/etc/init.d`. If the `rc` files are in a directory such as `/etc/rc.d/init.d`, then substitute that location in the example.
- Your system's default `runlevel` may be something other than '3', the level used in this example. You can run `/sbin/runlevel` to check the system's runlevel. If the runlevel on your system is different, then you must set the correct subdirectory; use your system's runlevel in place of the '3' in the preceding example.

Start license server upon reboot on Ubuntu

To start the license server and `pgroupd` license daemon each time your system is booted:

1. Log in as root.

You **must** be logged in as root to successfully execute these commands.

2. Verify you have set the PGI environment variable as described in Step 3 of this installation process.
3. Execute the following two commands:

```
% cp $PGI/linux86-64/19.3/bin/lmgrd.rc  /etc/init.d/lmgrd
% ln -s /etc/init.d/lmgrd  /etc/rc.d/rc3.d/S90lmgrd
```

There are two values in this example that may be different on your system:

- Your `rc` files may be in a directory other than the one in the example: `/etc/init.d`. If the `rc` files are in a directory such as `/etc/rc.d/init.d`, then substitute that location in the example.
- Your system's default `runlevel` may be something other than '3', the level used in this example. You can run `/sbin/runlevel` to check the system's runlevel. If the runlevel on your system is different, then you must set the correct subdirectory; use your system's runlevel in place of the '3' in the preceding example.

chkconfig(8) Utility

Most Linux distributions include the **`chkconfig(8)`** utility which manages the `runlevel` scripts. If your system has this tool and you wish to use it, then run the following commands:

```
% cp $PGI/linux86-64/19.3/bin/lmgrd.rc /etc/init.d/
% /sbin/chkconfig --add lmgrd
```

These commands create the appropriate links in the `/etc/init.d` directory hierarchy. For more information on **`chkconfig`**, please refer to the manual page.

Important

PGI Professional users can co-install Release 2019 with older releases and use older releases with the latest versions of `lmgrd` and `pgroupd` serving a Release 2019 license file.

If you use the `lmgrd.rc` file to start `lmgrd` automatically after a reboot of your system, you need to modify your `lmgrd` script in the `/etc/rc.d` or `/etc/init.d` directory to use the latest `lmgrd` daemon, which is now stored in a central directory instead of version-specific bin directories.

For example, your `lmgrd` script may look like this.

```
## Path to master daemon lmgrd
# Commented out previous path to 13.8:
#LMGRD=$PGI/linux86-64/13.8/bin/lmgrd
LMGRD=$PGI/linux86-64/flexlm/lmgrd

## Command to stop lmgrd
#Commented out previous path to 13.8:
#LMUTIL=$PGI/linux86-64/13.8/bin/lmutil
LMUTIL=$PGI/linux86-64/flexlm/lmutil
```

7. Review documentation.

   PGI documentation is available online in both HTML, www.pgroup.com/resources/docs/19.3/x86/index.htm and PDF, www.pgroup.com/resources/docs/19.3/x86/index.htm formats.

8. Complete network installation tasks.

Skip this step if you are not installing PGI network floating license using a network installation.

For a network installation, you must run the local installation script on each system on the network where the compilers and tools will be available for use.

If your installation base directory is `/opt/pgi` and `/usr/pgi/shared/19.3` is the common local directory, then run the following commands on each system on the network.

```
/opt/pgi/linux86-64/19.3/bin/makelocalrc  -x /opt/pgi/linux86-64/19.3 \
     -net /usr/pgi/shared/19.3
```

These commands create a system-dependent file `localrc.machinename` in the `/opt/pgi/linux86-64/19.3/bin` directory. The commands also create the following three directories containing libraries and shared objects specific to the operating system and system libraries on that machine:

```
/usr/pgi/shared/19.3/lib
/usr/pgi/shared/19.3/liblf
```

```
/usr/pgi/shared/19.3/lib64
```

The **makelocalrc** command does allow the flexibility of having local directories with different names on different machines. However, using the same directory on different machines allows users to easily move executables between systems that use PGI-supplied shared libraries.

Installation of the PGI products for Linux is now complete. For assistance with difficulties related to the installation, please contact the PGI technical reporting service, pgicompilers.com/support_request.

The following two sections contain information detailing the directory structure of the PGI installation, and instructions for PGI end-users to initialize environment and path settings to use the PGI compilers and tools.

4.3. End-user Environment Settings

4.3.1. PGI Compilers and Tools

After software installation is complete, each user's shell environment must be initialized to use the PGI compilers and tools. Assume the license file is in `/opt/pgi/license.dat`, and the `lmgrd` license manager is running.

Each user must issue the following sequence of commands to initialize the shell environment before using the PGI compilers and tools.

To make the PGI compilers and tools available:

In csh, use these commands:

```
% setenv PGI /opt/pgi
% setenv MANPATH "$MANPATH":$PGI/linux86-64/19.3/man
% setenv LM_LICENSE_FILE $PGI/license.dat
% set path = ($PGI/linux86-64/19.3/bin $path)
```

Once the compilers are available, you can make the Open MPI commands and Open MPI man pages accessible using these csh commands:

```
% set path = ($PGI/linux86-64/19.3/mpi/openmpi/bin $path)
% setenv MANPATH "$MANPATH":$PGI/linux86-64/19.3/mpi/openmpi/man
```

In bash, sh, or ksh, use these commands:

```
$ PGI=/opt/pgi; export PGI
$ MANPATH=$MANPATH:$PGI/linux86-64/19.3/man; export MANPATH
$ LM_LICENSE_FILE=$PGI/license.dat; export LM_LICENSE_FILE
$ PATH=$PGI/linux86-64/19.3/bin:$PATH; export PATH
```

Once the 64-bit compilers are available, you can make the Open MPI commands and Open MPI man pages accessible using these commands.

```
$ export PATH=$PGI/linux86-64/19.3/mpi/openmpi/bin:$PATH
$ export MANPATH=$MANPATH:$PGI/linux86-64/19.3/mpi/openmpi/man
```

4.3.2. Open MPI access

To access Open MPI and the Open MPI man pages for linux86-64, execute these commands:

In csh, use these commands:

```
% set path = (/opt/pgi/linux86-64/19.3/bin  \
  /opt/pgi/linux86-64/19.3/mpi/openmpi/bin $path)
% setenv MANPATH  "$MANPATH":/opt/pgi/linux86-64/19.3/mpi/openmpi/man
```

In bash, sh or ksh, use these commands:

```
$ export PATH=/opt/pgi/linux86-64/19.3/bin: \
  /opt/pgi/linux86-64/19.3/mpi/openmpi/bin:$PATH
$ export MANPATH=$MANPATH:/opt/pgi/linux86-64/19.3/mpi/openmpi/man
```

4.3.3. MVAPICH access

To access MVAPICH2 and the MVAPICH2 man pages for linux86-64, execute these commands:

In csh, use these commands:

```
% set path = (/opt/pgi/linux86-64/19.3/bin
      /opt/pgi/linux86-64/19.3/mpi/mvapich/bin $path)
% setenv MANPATH  "$MANPATH":/opt/pgi/linux86-64/19.3/mpi/mvapich/man
```

In bash, sh or ksh, use these commands:

```
$ export PATH=/opt/pgi/linux86-64/19.3/bin:
      /opt/pgi/linux86-64/19.3/mpi/mvapich/bin:$PATH
$ export MANPATH=$MANPATH:/opt/pgi/linux86-64/19.3/mpi/mvapich/man
```

Chapter 5. INSTALLATIONS ON MACOS

This section describes how to install PGI software on an Apple computer running Apple macOS.

PGI products for Apple macOS are supported on Intel processor-based computers running OS X version 10.10.5 or greater. PGI products are compatible with Apple Xcode 6.4 through 9.x installed. Xcode is available from the Apple App Store.

The default installation base directory is `/opt/pgi`.

5.1. Preparing to Install on Apple macOS

To prepare for the installation:

- Verify you have enough free disk space.
  - PGI installation requires 1.75 GB of free disk space.

  In addition, the download file requires approximately 600 MB.
- Verify that Xcode 6.4 or greater (through 9.x) is installed.
  - If you know how to run Xcode, start Xcode and click About Xcode to verify the version.
  - If you do not know how to run Xcode or are uncertain if it is installed on your system, try searching for it using Spotlight, which can be found by clicking the magnifying glass in the upper-right corner of the menu bar.

PGI software for macOS requires the Xcode application, which provides several components of the tool chain, including the system assembler, linker, and runtime libraries. However, PGI software runs in Terminal, not in the Xcode IDE, and the PGI debugger and PGI profiler have their own GUIs.

In preparation for installing PGI software on macOS, first review Figure 2, an overview of the installation process.

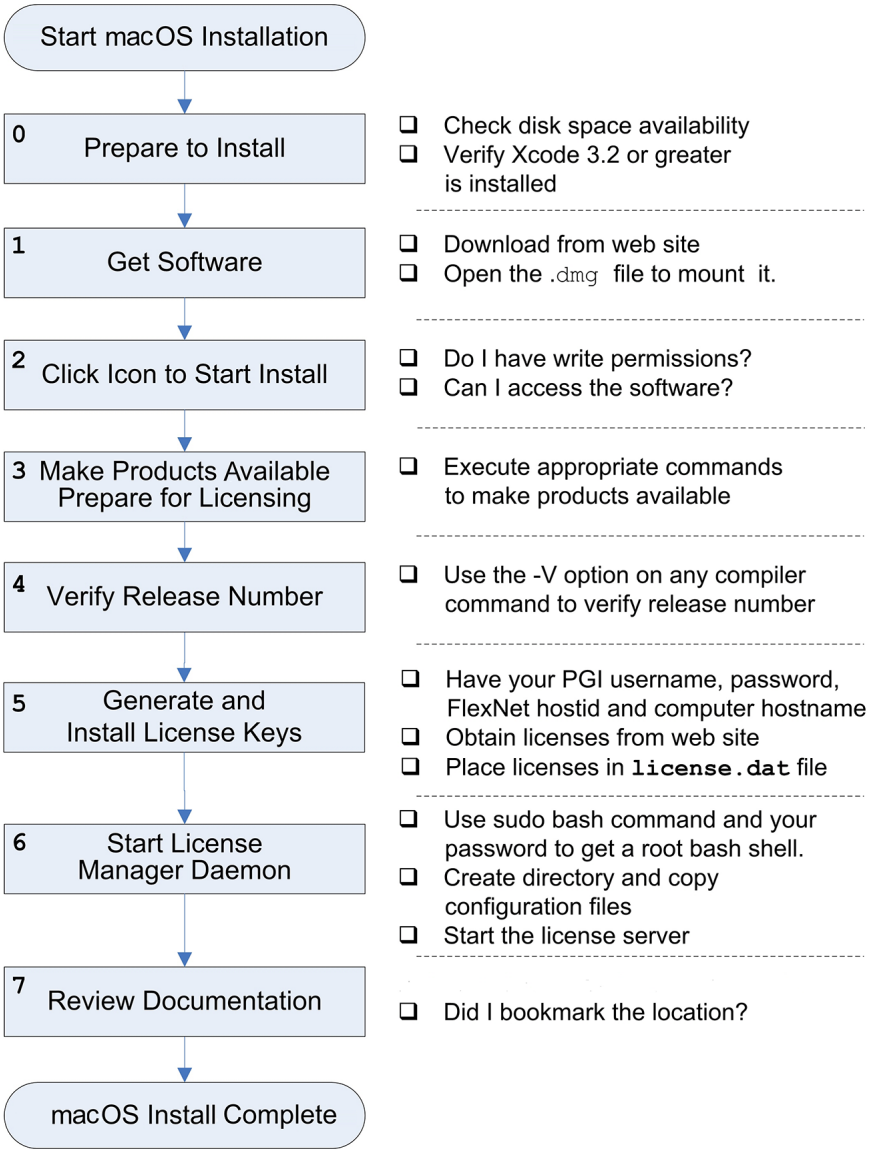

Figure 2 Apple macOS Installation Overview

5.2. System Configuration to Enable Debugger 'Attach'

The PGI debugger can debug programs that are launched by the debugger, either by specifying the program on the debugger's command line, using the `debug` or `load` commands, or by specifying a program in the Connections tab of the debugger GUI.

Apple macOS security features prevent the PGI debugger from attaching to a running program unless some configuration is performed. Users who need to attach to a running program must be able to enter the authentication credentials for a user that is a member of the _developer group. Users can be added to a group using the following command:

```
sudo dseditgroup -o edit -a usernametoadd -t user _developer
```

5.3. Installation Steps for macOS

Once you have prepared for the installation, follow these instructions to install the software:

1. Access the installation package.

   Download the software from the PGI download page, pgicompilers.com/downloads or another electronic distribution site. If it does not mount automatically, open the download package to mount it.

2. Install the software.

   Double-click the `PGI 19.3.pkg` icon, which is part of the mounted disk. Follow the installation instructions.

   1. When you see the initial system check dialog, click continue to allow the installation script to check that your system has the required components for installing the software, such as Xcode 6.4 or greater (through 9.x) and `gcc`.
   2. Select the installation drive. The compilers will be automatically be installed into the default directory: `/opt/pgi`.

   After the software is installed, the installation script performs some system-specific customization and then initializes for licensing.

3. Make PGI products accessible and prepare for licensing.

   To initialize the PGI compilers and tools, import the `PGI 2019` terminal settings file located in the PGI installation directory using Terminal's Shell | Import command. You can access the installation directory in the Open file dialog by typing `Command-Shift-G` and entering the path, for example `/opt/pgi`.

   Once initialized, you can open a Terminal session preconfigured with the PGI environment directly from within Terminal. To do this, bring up a PGI window by using either the `Shell | New Window` or `Shell | New Tab` menu commands; and then select `PGI 2019`.

   For more configuration information, refer to End-user Environment Settings.

4. Verify the release number of the installed software. It should say 19.3.

   To verify the release number of the products you have installed, use the `-V` option on any of the compiler commands, as illustrated in the following examples. If you use `-v` instead, you can also see the sequence of steps the compiler uses to compile and link programs for execution on your system.

For Fortran 77, use:	**pgf77 -V x.f**
For Fortran 2003, use:	**pgfortran -V x.f**
For C++, use:	**pgc++ -V x.cpp**
For ANSI C, use:	**pgcc -V x.c**

These commands can be successfully executed even if the files `x.f` or `x.c` do not exist and you have not completed the licensing phase of the installation. Use

them to check that you have installed the proper version of the compilers and have initialized your environment to enable access to that version.

5. Generate and install license keys.

   Most PGI products are license-managed using FlexNet licensing. This system requires that you possess a valid license key file for the licensed product to operate. Most PGI products include temporary license keys with the installation package. Permanent license keys are available from the PGI website, https://www.pgroup.com/license/pin_manage.php?view=keys.

   Other components, such as Open MPI, are open-source products that are not license-managed.

   Any temporary license keys included with your package will be installed automatically. You can verify this by confirming the existance of the `license.dat` file in your PGI installation directory. If so, you may skip to the next section.

   To obtain your permanent license key, you need the following information:

   - An account on the PGI website. You probably created this account when you downloaded the PGI software.

     Tip The username (email address) and password required to connect to the pgicompilers.com website are the same ones you used to download the installation software from the web site.

     - If you purchased a license without creating an account, one was created for you when your order was processed. Please check for an activation email from sales-noreply@pgroup.com.
     - If you don't have an account, you can create one at: pgicompilers.com/register.
   - The FlexNet *hostid* and *hostname* of the computer on which the software is installed. The installer echoes this information to your screen.

   You can also obtain your FlexNet hostid by using the following command after you have installed the products and initialized the environment variables:

```
% lmutil lmhostid
```

   You should see a message similar to the following message with one or more hostids displayed.

```
The FlexNet host ID of this machine is "12345678abcd edcba9876543".
```

   In this example, you can use either `12345678abcd` or `edcba9876543`, but not both, as the hostid.

   Hostids come from configured network cards. If you use your computer in multiple network environments, you may need to run the following command in each environment to see which hostids are configured. Then, to reduce potential license problems, choose the hostid that occurs in all your environments.

The preceding command returns output in the form: `HOSTNAME=xxxxx`. However, when you create a license, use only the actual hostname, that is xxxxx in the hostname field. For example, if the information displayed is `HOSTNAME=abcd1234`, then use only `abcd1234`. The goal is to have the hostname map to the IP address of the license server. Every system on the network should be able to *ping hostname* and get the correct IP address.

Generate License Keys

Once you have logged on using your registered PGI email and password, you should see a screen similar to the following:

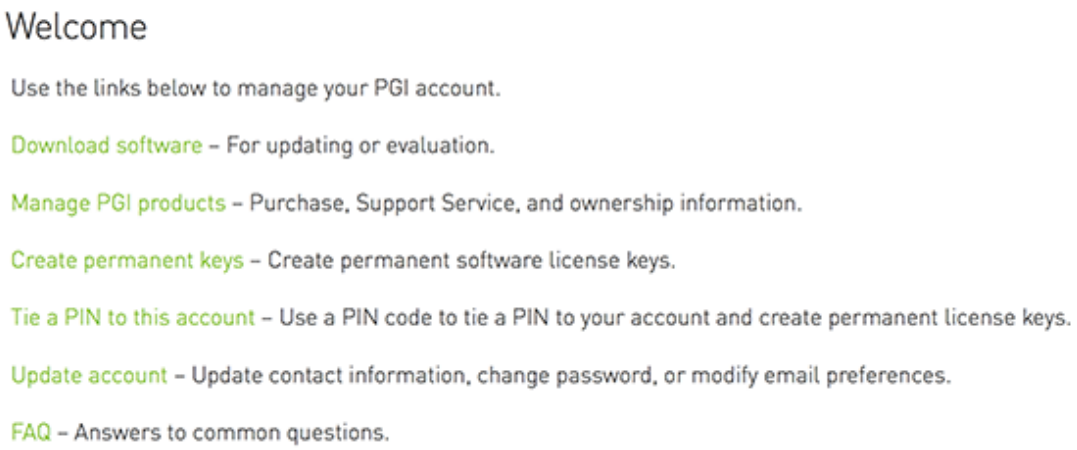

Figure 3 Welcome PGI Account Sample

Install License Keys

Once you have generated your permanent license key, copy and paste it into the file `/opt/pgi/license.dat`, or substitute the appropriate installation directory path if you have not installed in the default `/opt/pgi` directory.

For example, if you have purchased and generated a permanent licence key as previously described, the `license.dat` file should look similar to the following:

```
SERVER <hostname> <hostid> 27000
DAEMON pgroupd
PACKAGE PGI2016-<PGI_PIN> pgroupd <support end date> \
<...>
6167 7015 3F05 9C37 2315 ACDF 1B73 DAA9 FBAE"
```

In your license file:

- `<hostid>` should match the *hostid* you submitted above when you generated your license keys. You should see it on the SERVER line.
- If necessary, you can enter or edit the `<hostname>` entry manually, but you cannot edit the `<hostid>` entry or you will invalidate the license keys.
- <support end date> represents the date your support expires. For example, if your support end date for your PGI PIN (Product Identification Number) is August 1, 2017, then the date in your file is 2017.0801. For information on how to renew your support contract, refer to Product Support.

- <PGI_PIN> represents the six-digit product identification number for this license.

Please include your PIN when contacting PGI for technical support for the products you have purchased. This PIN is also in your order confirmation email.

6. Start the license manager daemon.

Important If you are using a PGI Community Edition license key or evaluating PGI software with a starter license key, you do not need to perform this step.

If this is your first installation of PGI software, you need to configure and start up the license server. This is not a requirement for subsequent installations.

Follow these steps to start the license server and `pgroupd` license daemon running on your system:

Please make sure that `com.pgroup.flexlm.plist` is owned and readable by root only

1. In Terminal, use the **sudo** command to get a root bash shell; enter your password:

```
% sudo bash
<password>
```

2. From the PGI bin directory, copy the PGI license configuration files:

```
% sudo cp com.pgroup.flexlm.plist /Library/LaunchDaemons
```

3. Change the file ownership:

```
% sudo chown root:wheel /Library/LaunchDaemons/com.pgroup.flexlm.plist
```

4. Change file access permissions:

```
% sudo chmod 0644 /Library/LaunchDaemons/com.pgroup.flexlm.plist
```

5. Start the license server:

```
% sudo launchctl load /Library/LaunchDaemons/com.pgroup.flexlm.plist
```

The license server should now be running. Enable remote login under the Sharing option in your System Preferences to have the license server restart automatically whenever you reboot.

Installation of your FlexNet licensing is now complete. For assistance with difficulties related to the installation, please contact the PGI technical reporting service, pgicompilers.com/support_request.

7. Review documentation.

PGI documentation is available online in both HTML, www.pgroup.com/resources/docs/19.3/x86/index.htm and PDF, www.pgroup.com/resources/docs/19.3/x86/index.htm formats.

5.4. End-user Environment Settings on macOS

Once software installation is complete, each user must initialize their environment to use the compilers successfully. You can either use the PGI Terminal settings or you can initialize your Terminal shell manually.

5.4.1. Use PGI Terminal Settings

There are two methods to use the PGI Terminal Settings:

- Use the PGI Terminal Settings by bringing up a PGI command window and double-clicking the `PGI2019.terminal` file. PGI compilers and tools are accessible and your environment is pre-initialized.
- Use the PGI Terminal Settings by launching the Terminal application located in the `Applications | Utilities` folder. Under the Shell menu, select either `New Window | PGI2019` or `New Tab | PGI2019`. Both commands open a bash shell that is preconfigured with the PGI environment.

5.4.2. Initialize your Terminal Shell Manually

To initialize your Terminal shell manually, you launch the Terminal application located in the `Applications | Utilities` folder and enter the commands described for the desired compilers.

Each of the following commands assume this:

The license file is in `/opt/pgi/license.dat`.

The `lmgrd` license manager is running.

Make the PGI compilers available

Launch the Terminal application located in the `Applications | Utilities` folder and enter the following commands to make the PGI compilers and tools available.

In bash, sh, or ksh, use these commands:

```
$ PGI=/opt/pgi; export PGI
$ LM_LICENSE_FILE=$PGI/license.dat; export LM_LICENSE_FILE
$ PATH=$PGI/osx86-64/19.3/bin:$PATH; export PATH
```

In csh, use these commands:

```
% setenv PGI /opt/pgi
% setenv LM_LICENSE_FILE $PGI/license.dat
% set path = ($PGI/osx86-64/19.3/bin $path)
```

5.4.3. Use MPICH

To use the bundled version of MPICH, use these commands:

In bash, sh, or ksh, use these commands:

```
$ PATH=$PGI/osx86-64/2019/mpi/mpich/bin:$PATH
$ export PATH
```

In csh, use this command:

```
% set path = ($PGI/osx86-64/2019/mpi/mpich/bin $path)
```

Chapter 6. INSTALLATIONS ON MICROSOFT WINDOWS

This section describes how to install PGI compilers and tools on a system running a Microsoft Windows operating system.

6.1. Preparing to Install on Windows

To use PGI compilers on Windows, there are a few prerequisites that must be fulfilled.

1. **Microsoft Update for Universal C Runtime**

   PGI compilers leverage the Microsoft tool chain. To use these tools on operating systems older than Windows 10, you must first install Microsoft's Update for Universal C Runtime, http://support.microsoft.com/en-us/kb/2999226 . Depending on the configuration details of your system, this update may have already been installed as part of the automatic Windows Updates applied to your system.
2. **Microsoft Windows SDK**

   The Microsoft Windows Software Development Kit (SDK) is required for all Windows installs. You can find links to download the Windows SDK on the PGI Microsoft SDK webpage, pgicompilers.com/microsoft-sdk.
3. **Microsoft Visual Studio 2017**

   Microsoft Visual Studio 2017 with Visual C++ is a prerequisite for PGI compilers on Windows. Either the community, professional, or enterprise editions will suffice. More information can be found on the Visual Studio webpage, visualstudio.microsoft.com/vs/.

6.2. Installation Steps for Windows

Download the software from the PGI download page, pgicompilers.com/downloads or another electronic distribution site. Select the install package appropriate for your operating system.

If you are using Internet Explorer 9, be sure to set the *compatibility mode* option when reading the download page, as it may look blank otherwise.

Once you have prepared for the installation, follow these instructions to install the software:

1. Move the installation executable to a separate, empty directory, e.g. **`%HOMEPATH%/pgi-installation`**.
2. Run the installation executables as Administrator.

   Administrator privileges are required to install PGI software.
3. If you have installed the PGI Community Edition, skip to step Step 9.
4. Run PGI Licensing.

   At the conclusion of the installation process, the installer will ask you if you would like to run the PGI Licensing Setup tool. This tool automates steps three (3) through six (6) in this process.

   To use the PGI Licensing Setup tool, your computer must meet these requirements:

   - Be connected to the Internet.
   - Have Microsoft .NET Framework 4.0 installed.
   - Have Internet Explorer Version 6.0 or higher installed and configured to allow storage of session cookies from websites, or specifically add **`https://www.pgroup.com`** to Control Panel > Internet Options > Security > Trusted Sites.

   If your computer does not meet these prerequisites, then you can either make any necessary changes to your system and run the PGI Licensing Setup tool later, or follow steps three through six in this process.

Important If you choose to run the PGI Licensing Setup tool later, you can find it in the PGI folder under the Start menu or as an icon on the Start screen.

When you run the program, the PGI Licensing Setup tool walks you through the license generation and installation process. You will be prompted for your PGI username and password. The tool also informs you of any problems it experiences with installation and provides instructions on how to proceed.

If the PGI Licensing Setup tool reports a successful license installation, installation is complete.

5. Make PGI products accessible and prepare for licensing.

   Once software installation is complete, PGI compilers and tools are accessible and your environment is pre-initialized whenever you bring up a PGI command window. To bring up a PGI command window, double-click on the PGI desktop icon.

6. Verify release number.

   Verify the release number of the products you have installed. Open the PGI command window from your desktop by double-clicking the PGI icon and entering one of the following commands. The release number is in the first line displayed in the BASH shell window.

For Fortran 77, use:	**pgf77 -V**
For Fortran 95, use:	**pgfortran -V**
For ANSI C, use:	**pgcc -V**

These commands can be successfully executed even if you have not completed the licensing phase of the installation. Use it to check that you have installed the proper version of the compilers and have initialized your environment to enable access to that version.

7. Generate and install license keys.

   Most PGI products are license-managed using FlexNet licensing. This system requires that you possess a valid license key file for the licensed product to operate. Most PGI products include temporary license keys with the installation package. Permanent license keys are available from the PGI website, https://www.pgroup.com/license/pin_manage.php?view=keys.

   Other components, such as Open MPI, are open-source products that are not license-managed.

   Any temporary license keys included with your package will be installed automatically. You can verify this by confirming the existance of the `license.dat` file in your PGI installation directory. If so, you may skip to the next section.

   To obtain your permanent license key, you need the following information:

   - An account on the PGI website. You probably created this account when you downloaded the PGI software.

     Tip The username (email address) and password required to connect to the pgicompilers.com website are the same ones you used to download the installation software from the web site.

     - If you purchased a license without creating an account, one was created for you when your order was processed. Please check for an activation email from sales-noreply@pgroup.com.
     - If you don't have an account, you can create one at: pgicompilers.com/register.
   - The FlexNet *hostid* and *hostname* of the computer on which the software is installed. The installer echoes this information to your screen.

After you have installed the PGI software, you can obtain your FlexNet hostid by double-left-clicking on the PGI desktop icon to open a PGI command window, and typing these commands:

```
PGI$  cd $PGI
PGI$  cat license.info
```

You should see information similar to the following:

```
   FlexNet Host ID:  0123456789ab
   Installation: C:\Program Files\PGI\
   PGI Release: 2019
```

With this information, connect to **pgicompilers.com/license/pin_manage.php** to generate license keys. Log in with your PGI registered email and password.

Install License Keys

Once you have generated your permanent license key, cut and paste it into your `license.dat` file. In a typical configuration, where `C:\` is the system drive and you installed the software using the default location, this file would be found in `C:\Program Files\PGI\license.dat`.

- If you have not previously received license keys from PGI, replace the contents of the `license.dat` file created during installation with the license keys you generated using the preceding steps.

You must have Administrator privileges to modify or replace the `license.dat` file.

- If your `license.dat` file already contains keys that you previously received from PGI, first make a copy of the existing `license.dat` file in case you encounter a problem with the new license. Then do one of the following:
  - If the license keys in the file are for a previous release of the same product, overwrite the keys.
  - If the license keys in the file are for different PGI products, append the new keys to the keys already in this file.

8. Start the PGI License Service.

If you are using a PGI Community Edition license key or evaluating a PGI product using a starter license key, you do not need to start the license server.

The PGI License Server is a Windows Service. To start it, follow these steps:

1. Open the Services dialog from the Control Panel:

   Control Panel | Administrative Tools | Services
2. Select *PGI License Server*.

3. Select *Start*, if the PGI service is stopped. If the PGI service is running, STOP and then START it. This process guarantees that the new PGI license is served and that the older, cached PGI license is no longer active.

The PGI License Server service starts automatically on system reboot, provided that the `license.dat` file contains valid keys.

9. Review documentation.

PGI documentation is available online in both HTML, www.pgroup.com/resources/docs/19.3/x86/index.htm and PDF, www.pgroup.com/resources/docs/19.3/x86/index.htm formats.

10. Customize PGI Settings

Optionally, you can customize the setup as described in Customizing the Command Window and PGI Default Installation Directories below.

6.3. Customizing the Command Window

By default, when you double-click on the PGI desktop icon, a standard black-background command window appears on your screen. This window is pre-initialized with environment and path settings for use of the PGI compilers and tools. If you prefer different background or text colors, font style, window size, or scrolling capability, you can customize the "shortcut" that creates the PGI command window.

To customize your window, right-click the PGI desktop icon, and select "Properties" from the pop-up menu. In the PGI Properties dialog box, select the tabs for the features you want to customize, and make the desired modifications.

6.4. PGI Default Installation Directories

This section contains information about the default installation directories. The default installation directory depends on your platform.

The following table lists the default installation directories, indicating the related platform or tool.

6.4.1. Default Installation Directories

Table 3 Default Windows Installation Directories

Product	Default Installation Directory
PGI	`%SYSTEMDRIVE%\Program Files\PGI`
PGI CE	`%SYSTEMDRIVE%\Program Files\PGICE`
Cygwin	`%SYSTEMDRIVE%\cygwin`

6.4.2. Tool Default Versions

PGI ships a subset of Cygwin with its products on Windows. Beginning with PGI release 14.4, the version of Cygwin bundled with PGI is Cygwin 1.7.27. This was the version of Cygwin current as of January 2014. In general, PGI users should not notice significant differences in Cygwin behavior from the previous version, which was Cygwin 1.7.16. For information about configuring and troubleshooting Cygwin, you can visit the following websites:

Cygwin FAQ: http://cygwin.com/faq.html
Cygwin Users Guide: http://cygwin.com/cygwin-ug-net.html
Cygwin mailing lists and archives: http://cygwin.com/lists.html

6.5. PGROUPD_LICENSE_FILE and FLEXLM_BATCH

This section describes two environment variables that you can use with FlexNet:

PGROUPD_LICENSE_FILE
FLEXLM_BATCH

6.5.1. PGROUPD_LICENSE_FILE

The recommended installation location is the default `C:\Program Files\PGI` folder; the recommended location for storing your license keys is as the file `C:\Program Files\PGI\license.dat`.

The system environment variable `PGROUPD_LICENSE_FILE` is not required by PGI products on Windows but you can use it to override the default location that is searched for the `license.dat` file.

To use the system environment variable `PGROUPD_LICENSE_FILE`, set it to the full path of the license key file. To do this, follow these steps:

1. Open the System Properties dialog from the Control Panel's **`System`** option.
2. Select the **`Advanced`** tab.
3. Click the **`Environment Variables`** button.
   - If `PGROUPD_LICENSE_FILE` is not already an environment variable, create a new system variable for it. Set its value to the full path of your `license.dat` file.
   - If `PGROUPD_LICENSE_FILE` already exists as an environment variable, append the path to the license file in the variable's current value using a semi-colon to separate entries.

6.5.2. FLEXLM_BATCH

By default, on Windows, the license server creates interactive pop-up messages to issue warnings and errors. This behavior is controlled by the environment variable `FLEXLM_BATCH`.

Although it is not recommended, you can prevent interactive pop-ups from appearing. To do this, set the environment variable `FLEXLM_BATCH` to 1.

6.6. Windows Firewall and PGI Debugging

Microsoft Windows Firewall runs by default on some Windows systems. Unless the PGI debugger is included in the Windows Firewall Exceptions list, Windows Firewall may try to block the PGI debugger engine when you debug a program.

6.6.1. Automatic Additions to Windows Firewall Exceptions List

When PGI software is installed

If Windows Firewall is enabled when PGI software is installed, certain components are automatically added to the Exceptions list. These components appear as follows:

PGI Debugger 2019

When PGI software is uninstalled

If Windows Firewall is enabled when PGI software is uninstalled, the components previously listed are automatically removed from the Exceptions list.

When Starting the Debugger

If the PGI debugger is not listed in the Exceptions list and Windows Firewall is enabled when you use the PGI debugger to debug, the Firewall displays a dialog that allows you to Unblock the application named **pgdebug**. If you choose the Unblock option, Windows Firewall adds the debugger (pgdebug) to its list of Exceptions and you do not see the pop-up dialog again unless you reinstall.

On rare occasions, this pop-up may prevent the debug session from starting correctly. If this occurs, use the Task Manager to end the debug session.

You can manually add these components to the Windows Firewall list of Exceptions.

6.7. Common Windows Installation Problems

The most common installation problems on Windows are related to licensing.

To troubleshoot your installation, first check that the `license.dat` file you are using contains valid license keys. Second, check that the PGI License Server, a Windows Service, has been started (Control Panel > Admin Tools > Services > PGI).

Typical FlexNet errors encountered may include the following:

- When using a PGI compiler or tool, a FlexNet License Manager dialog appears that states: `'LICENSE MANAGER PROBLEM: No such feature exists.'`

  This message may appear because the `license.dat` file accessed by the FlexNet License Manager does not contain valid license keys. Possible reasons for an invalid license key include:

  - Bad format license file—if you stored it from Word, make sure it was stored in plain text format.
  - License does not support this release or this operating system—Make sure your license covers the release you are installing, and is a Windows license.
- When using a PGI compiler or tool, you may see one of the following FlexNet License Manager dialog messages: `'LICENSE MANAGER PROBLEM: Cannot connect to license server system.'` or `'LICENSE MANAGER PROBLEM: Failed to checkout license'`

  This message may appear as a result of one of the following conditions:

  - The PGI License Server is not running.
  - Firewall problems exist.
  - The hostname in the license is not one the license server can locate.

    Refer to the information earlier in this section on restarting the license server.

    The hostname must map to the IP address of the license server. You should be able to `ping hostname` and find that it is mapped to the IP address of the license service. If not, double-check the hostname of the license server and/or your network configuration.

    Every machine using the license server, including the server itself, should be able to successfully ping the hostname and get back the IP address of the server. If your system is using a Firewall, you need to enable two ports for the licensing daemons `lmgrd` and `pgroupd` to communicate with the PGI compilers and tools. Edit or add these ports to the `license.dat` file.

    - On the SERVER line, change the default port 27000 to the `lmgrd` port.
    - On the DAEMON line, append PORT=### just after `pgroupd`, replacing `###` with the port number `pgroupd` is allowed to use.
- By default, on Windows, the license server generates interactive pop-up messages to issue warning and errors. You can use the environment variable FLEXLM_BATCH to prevent interactive pop-up windows.

  To do this, set the environment variable FLEXLM_BATCH to 1.

For assistance with difficulties related to the installation, please contact the PGI technical reporting service, pgicompilers.com/support_request.

Chapter 7. CONTACT INFORMATION

You can contact NVIDIA's PGI compilers and tools team at:

9030 NE Walker Road, Suite 100
Hillsboro, OR 97006

Or electronically using any of the following means:

Fax: +1-503-682-2637
Sales: sales@pgroup.com
WWW: https://www.pgroup.com or pgicompilers.com

The PGI User Forum, pgicompilers.com/userforum is monitored by members of the PGI engineering and support teams as well as other PGI customers. The forums contain answers to many commonly asked questions. Log in to the PGI website, pgicompilers.com/login to access the forums.

Many questions and problems can be resolved by following instructions and the information available in the PGI frequently asked questions (FAQ), pgicompilers.com/faq.

Submit support requests using the PGI Technical Support Request form, pgicompilers.com/support-request.

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA, the NVIDIA logo, Cluster Development Kit, PGC++, PGCC, PGDBG, PGF77, PGF90, PGF95, PGFORTRAN, PGHPF, PGI, PGI Accelerator, PGI CDK, PGI Server, PGI Unified Binary, PGI Visual Fortran, PGI Workstation, PGPROF, PGROUP, PVF, and The Portland Group are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2013–2019 NVIDIA Corporation. All rights reserved.