

# PGI Visual Fortran<sup>®</sup> Release 8.0

## Release Notes

The Portland Group®  
STMicroelectronics, Inc  
Two Centerpointe Drive  
Lake Oswego, OR 97035  
[www.pgroup.com](http://www.pgroup.com)

Optimized for

---



While every precaution has been taken in the preparation of this document, The Portland Group® (PGI®), a wholly-owned subsidiary of STMicroelectronics, Inc., makes no warranty for the use of its products and assumes no responsibility for any errors that may appear, or for damages resulting from the use of the information contained herein. The Portland Group® retains the right to make changes to this information at any time, without notice. The software described in this document is distributed under license from STMicroelectronics, Inc. and/or The Portland Group® and may be used or copied only in accordance with the terms of the license agreement (“EULA”). No part of this document may be reproduced or transmitted in any form or by any means, for any purpose other than the purchaser’s or the end user’s personal use without the express written permission of STMicroelectronics, Inc and/or The Portland Group®

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this manual, STMicroelectronics was aware of a trademark claim. The designations have been printed in caps or initial caps.

PGF95, PGF90, and PGI Unified Binary are trademarks; and PGI, PGHPF, PGF77, PGCC, PGC++, PGI Visual Fortran, PVF, Cluster Development Kit, PGPROF, PGDBG, and The Portland Group are registered trademarks of The Portland Group Incorporated. PGI CDK is a registered trademark of STMicroelectronics. \*Other brands and names are the property of their respective owners.

PGI Visual Fortran® 8.0  
Release Notes  
Copyright © 2008-2009

The Portland Group®  
STMicroelectronics, Inc. - All rights reserved.  
Printed in the United States of America

First Printing	Release 8.0-1	November 2008
Second Printing	Release 8.0-2	December 2008
Third Printing	Release 8.0-3	January 2009
Fourth Printing	Release 8.0-4	February 2009
Fifth Printing	Release 8.0-5	March 2009
Sixth Printing	Release 8.0-6	April 2009

Technical support: [www.pgroup.com/support](http://www.pgroup.com/support)

# Table of Contents

---

<b>1</b>	<b>PVF<sup>®</sup> INTRODUCTION.....</b>	<b>1</b>
1.1	PRODUCT OVERVIEW .....	1
1.2	TERMS AND DEFINITIONS .....	2
<b>2</b>	<b>PVF 8.0 RELEASE OVERVIEW .....</b>	<b>3</b>
2.1	SUPPORTED PROCESSORS .....	3
2.2	SUPPORTED OPERATING SYSTEMS.....	4
<b>3</b>	<b>NEW OR MODIFIED FEATURES.....</b>	<b>7</b>
3.1	NEW OR MODIFIED COMPILER OPTIONS .....	8
3.2	COMMON COMPILER FEEDBACK FORMAT (CCFF) .....	10
3.3	NEW OR MODIFIED OPENMP FEATURES.....	10
3.3.1	New or Modified OpenMP Directives and Pragmas .....	10
3.3.2	New or Modified OpenMP Run-time Library Routines.....	11
3.3.3	New or Modified OpenMP Environment Variables.....	11
3.4	NEW OR MODIFIED MPI SUPPORT .....	12
3.5	STATIC AND DYNAMIC LINKING WITH PVF / PGI COMPILERS.....	12
3.5.1	Linking PVF and VC++ Projects .....	13
3.5.2	Common Link-time Errors .....	13
3.5.3	Feature Tips.....	14
3.6	ADDITIONAL LIBRARY INTERFACES .....	15
<b>4</b>	<b>ALTERNATE COMPILER RELEASE SELECTION.....</b>	<b>17</b>
4.1	FOR A SINGLE PROJECT .....	17
4.2	FOR ALL PROJECTS.....	18
<b>5</b>	<b>DISTRIBUTION AND DEPLOYMENT.....</b>	<b>19</b>
5.1	APPLICATION DEPLOYMENT AND REDISTRIBUTABLES .....	19
5.1.1	PGI Redistributables .....	20
5.1.2	Microsoft Redistributables .....	20

<b>6</b>	<b>KNOWN LIMITATIONS AND CORRECTIONS.....</b>	<b>21</b>
6.1	PVF IDE LIMITATIONS.....	21
6.2	PVF DEBUGGING LIMITATIONS.....	22
6.3	MICROSOFT VISUAL STUDIO ADD-IN LIMITATIONS FOR SUA ....	22
6.4	PGI COMPILER LIMITATIONS.....	23
6.5	DOCUMENTATION CORRECTIONS .....	24
6.5.1	Functions ctime() and time().....	24
6.5.2	Message Text Errors .....	24
6.5.3	FORTRANOPT Environment Variable.....	24
6.6	CORRECTIONS.....	25
<b>7</b>	<b>CONTACT INFORMATION AND DOCUMENTATION .....</b>	<b>27</b>

# 1 PVF<sup>®</sup> Introduction

---

Welcome to PGI Visual Fortran<sup>®</sup>, a set of Fortran compilers and development tools for 32-bit and 64-bit Windows integrated with Microsoft\* Visual Studio\*.

This document describes the new features of the PVF IDE interface, differences in the PVF 8.0-6 compilers and tools from previous releases, and late-breaking information not included in the standard product documentation.

PGI Visual Fortran (PVF<sup>®</sup>) is licensed using FLEXnet, the flexible license management system from Acrecco Software\*. Instructions for obtaining a permanent license are included in your order confirmation. More information on licensing is in the PVF Installation Guide for this release.

## 1.1 Product Overview

There are two products in the PVF product family. Each product is integrated with a particular version of Microsoft Visual Studio:

- PGI Visual Fortran 2005  
This product is integrated with Microsoft Visual Studio 2005 (VS 2005).
- PGI Visual Fortran 2008  
This product is integrated with Microsoft Visual Studio 2008 (VS 2008).

Throughout this document, “PGI Visual Fortran” refers to both PVF products collectively. Similarly, “Microsoft Visual Studio” refers to both VS 2005 and VS 2008. When it is necessary to distinguish between the products, the document uses the full product name.

Single-user node-locked and multi-user network floating license options are available for both products. When a node-locked license is used, one user at a time can use PVF on the single system where it is installed. When a network floating license is used, a system is selected as the server and it controls the licensing, and users from any of the client machines connected to the server can use PVF. Thus multiple users can simultaneously use PVF, up to the maximum number of users allowed by the license.

PVF provides a complete Fortran development environment fully integrated with Microsoft Visual Studio 2008 or 2005. It includes a custom Fortran Build Engine that automatically derives build dependencies, a Fortran-aware editor, a custom PGI Debug Engine integrated with the Visual Studio debugger, PGI Fortran compilers, and PVF-specific property pages to control the configuration of all of these.

Release 8.0 of PGI Visual Fortran includes the following components:

- PGF95 OpenMP and auto-parallelizing Fortran 90/95 compiler
- PGF77 OpenMP and auto-parallelizing FORTRAN 77 compiler
- PVF Visual Studio integration components
- AMD Core Math Library 4.1-0 (ACML)
- PVF documentation fully integrated with Visual Studio help

PGI Visual Fortran 2008 also includes:

- Microsoft Visual Studio 2008 Shell SP1 (Integrated Development Environment – IDE)

A version of PVF 2008 is also available without the Visual Studio Shell. Use this package if you already have Visual Studio 2008 installed on your system.

## 1.2 Terms and Definitions

These release notes contain a number of terms and definitions with which you may or may not be familiar. If you encounter a term in these notes with which you are not familiar, please refer to the online glossary at:

[www.pgroup.com/support/definitions.htm](http://www.pgroup.com/support/definitions.htm)

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

**AMD64** – a 64-bit processor from AMD™ designed to be binary compatible with 32-bit x86 processors, and incorporating new features such as additional registers and 64-bit addressing support for improved performance and greatly increased memory range. This term includes the AMD Athlon64™, AMD Opteron™, AMD Turion™, AMD Barcelona, and AMD Shanghai processors.

**Intel 64** – a 64-bit IA32 processor with *Extended Memory 64-bit Technology* extensions designed to be binary compatible with AMD64 processors. This includes Intel Pentium 4, Intel Xeon, and Intel Core 2 processors.

# 2

# PVF 8.0

## Release Overview

---

This chapter describes the processors and operating systems on which PGI Visual Fortran is supported.

For details on how to use PVF, refer to the *PGI Visual Fortran User's Guide*.

### 2.1 Supported Processors

PGI Visual Fortran is supported on 32-bit and 64-bit Windows operating systems running on x86 and x64 compatible processors. The following table lists the processors on which Release 8.0-6 of PGI Visual Fortran is supported. The table also includes the CPUs available and supported in dual-core versions.

PVF uses the Fortran | Optimization | Processor-Specific Optimization property to generate executables that utilize features and optimizations specific to a given CPU and operating system environment. From the command line, the `-tp <target>` command-line option accomplishes the same results.

#### Processors Supported by PGI 8.0

Brand	CPU	Cores	<target>	Memory Address	Floating Point HW					
					SSE1	SSE2	SSE3	SSSE3	SSE4	ABM and SSE4a
AMD	Opteron/Quadcore	4	shanghai-64	64-bit	Yes	Yes	Yes	No	No	Yes
AMD	Opteron/Quadcore	4	shanghai	64-bit	Yes	Yes	Yes	No	No	Yes
AMD	Opteron/Quadcore	4	barcelona-64	64-bit	Yes	Yes	Yes	No	No	Yes
AMD	Opteron/Quadcore	4	barcelona	32-bit	Yes	Yes	Yes	No	No	Yes
AMD	Opteron/Athlon64	2	k8-64	32-bit	Yes	Yes	Yes	No	No	No

AMD	Opteron/Athlon64	2	k8-32	32-bit	Yes	Yes	Yes	No	No	No
AMD	Opteron Rev E/F Turion /Athlon64	2	k8-64e	64-bit	Yes	Yes	Yes	No	No	No
AMD	Opteron Rev E/F	2	k8-32	32-bit	Yes	Yes	No	No	No	No
AMD	Turion64 Turion /Athlon64	1	k8-64e	64-bit	Yes	Yes	Yes	No	No	No
AMD	Turion64	1	k8-32	32-bit	Yes	Yes	No	No	No	No
Intel	Penryn	4	penryn-64	64-bit	Yes	Yes	Yes	Yes	Yes	No
Intel	Penryn	4	penryn	32-bit	Yes	Yes	Yes	Yes	Yes	No
Intel	Core 2	2	core2-64	64-bit	Yes	Yes	Yes	Yes	Yes	No
Intel	Core 2	2	core2	32-bit	Yes	Yes	Yes	Yes	Yes	No
Intel	P4/Xeon EM64T	2	p7-64	64-bit	Yes	Yes	Yes	Yes	No	No
Intel	P4/Xeon EM64T	2	p7	32-bit	Yes	Yes	Yes	Yes	No	No
Intel	Xeon/Pentium4	1	p7	32-bit	Yes	Yes	No	No	No	No
AMD	Athlon XP/MP	1	athlonxp	32-bit	Yes	No	No	No	No	No
Intel	Pentium III	1	piii	32-bit	Yes	No	No	No	No	No
AMD	Athlon	1	athlon	32-bit	No	No	No	No	No	No
AMD	K6	1	k6	32-bit	No	No	No	No	No	No
Intel	Pentium II	1	p6	32-bit	No	No	No	No	No	No
Generic	Generic x86	1	p5 or px	32-bit	No	No	No	No	No	No

## 2.2 Supported Operating Systems

The following table lists the operating systems, and their equivalents, on which PVF 8.0-6 is supported.

<i>Distribution</i>	<i>X86 (32-bit)</i>	<i>64-bit</i>
<b><i>Microsoft Windows XP Professional</i></b>	<i>Yes</i>	<i>No</i>
<b><i>Microsoft Windows XP Professional x64 Edition</i></b>	<i>Yes</i>	<i>Yes</i>
<b><i>Microsoft Windows Vista</i></b>	<i>Yes</i>	<i>No</i>
<b><i>Microsoft Windows Vista x64 Edition</i></b>	<i>Yes</i>	<i>Yes</i>
<b><i>Microsoft Windows Server 2008</i></b>	<i>Yes</i>	<i>No</i>
<b><i>Microsoft Windows Server 2008 x64 Edition</i></b>	<i>Yes</i>	<i>Yes</i>
<b><i>Microsoft Windows Server 2003</i></b>	<i>Yes</i>	<i>No</i>

<i>Microsoft Windows Server 2003 x64 Edition</i>	<i>Yes</i>	<i>Yes</i>
<i>Microsoft Windows Server 2003 R2</i>	<i>Yes</i>	<i>No</i>
<i>Microsoft Windows Server 2003 R2 x64 Edition</i>	<i>Yes</i>	<i>Yes</i>

**Note.** PVF is not supported on Microsoft Windows 2000.

[www.pgroup.com/support/install.htm](http://www.pgroup.com/support/install.htm) lists any new Windows distributions that may be explicitly supported by the PGI compilers. If your operating system is newer than any of those listed in the preceding table, the installation may still be successful.



# 3 New or Modified Features

---

The following list contains the new features of this release of PGI Visual Fortran as compared to prior releases.

- Support for **Visual Studio 2008 SP1**.
- **Visual Studio 2008 Shell SP1 bundled into PVF 2008.**  
For users with Visual Studio 2008 already installed on their systems, a version of PVF 2008 without the Visual Studio Shell is also available.
- Continued support for Visual Studio 2005, including support for concurrent use of Visual Studio 2005 and 2008 on the same system.
- **OpenMP 3.0** parallel programming for multi-core x64 CPUs and multi-socket servers, including full support for OpenMP 3.0 and TASKs in PGF95 and PGCC<sup>®</sup>, as described in New or Modified OpenMP Features on page 10.
- **Common Compiler Feedback Format (CCFF)** sections are now generated by all PGI 8.0 compilers; CCFF is a standardized format for storing and reporting of optimization information and hints to users, described in Common Compiler Feedback Format (CCFF) on page 10.
- **PGDBG Debugger and PVF Enhancements**
  - OpenMP 3.0 debugging, including support for tasks
  - Build MSMPI applications on Windows clusters from within PVF
- **Automatic licensing generation** from [www.pgroup.com](http://www.pgroup.com), and license server setup during the with point-and-click Windows installation.
- **Compiler Optimizations and Features**
  - Computation and reporting of compute intensity of loops in all languages
  - Packed SSE code generation for unrolled loops
  - SSE vectorization of generalized reduction loops
  - Improved scalar prefetching, spill tuning and live range splitting
  - Improved static estimation of block execution frequencies

- Auto-generation of DWARF for improved tools interoperability
- Enhanced Fortran 95 DWARF generation
- **Improved Fortran/C interoperability** with the new CVF-compatibility option `-Miface=cref`.
- Faster disassembly and symbol loading by the PVF debugger.
- **Improved support for debugging** of mixed PVF/VC++ applications through improved accuracy and speed of VC++ symbol processing.
- **Support for the CVF-compatible attribute:**  
!DEC\$ ATTRIBUTE nomixed\_str\_len\_arg.  
This attribute specifies that hidden lengths are placed in sequential order at the end of the list, the method used by `-Miface=unix`, (previously `-Munix`).
- **Note.** This attribute only applies to CREF-style routines or to routines that use the default Windows calling conventions.
- **Expanded Platform Support**
  - Support for the New Quad-Core AMD Opteron processor (AMD Shanghai)
  - PGI Unified Binary™ support for the latest AMD and Intel processors
  - Microsoft Vista 32-bit and 64-bit, Microsoft HPC Server 2008
- **Updated Documentation** including the PGI User's Guide, PGI Tools Guide, PVF User's Guide, and Fortran Reference.

## 3.1 New or Modified Compiler Options

Unknown options are now treated as errors instead of warnings. This change makes it a compiler error to pass switches that are not known to the compiler. The new `-noswitcherror` issues warnings instead of errors for unknown switches.

The following compiler options have been added or modified in PGI 8.0.

- `-Mnodwarf` is a new switch that specified not to add DWARF debug information
- `-Mnofpapprox` is a new switch that specifies not to use low-precision fp approximation operations

- *-Minfo* has a number of new suboptions:  
allImplies a number of suboptions:  
*-Minfo* =accel,inline,ipa,loop,lre,mp,opt,par,unified,vect  
accel        Enable Accelerator information  
ccff        Append information, such as optimization info,  
              to object file  
ftn         Enable Fortran-specific information  
hpf         Enable HPF-specific information  
inline      Enable inliner information  
lre         Enable LRE information  
par         Enable parallelizer information  
pfo         Enable profile feedback information  
vect        Enable vectorizer information
- *-Mneginfo* has a number of new suboptions:  
allImplies a number of suboptions:  
*-Mneginfo*=accel,inline,ipa,loop,lre,mp,opt,par,vect  
accel        Enable Accelerator information  
ftn         Enable Fortran-specific information  
hpf         Enable HPF-specific information  
inline      Enable inliner information  
ipa         Enable IPA information  
lre         Enable LRE information  
mp         Enable OpenMP information  
opt         Enable optimizer information  
par         Enable parallelizer information  
pfo         Enable profile feedback information  
vect        Enable vectorizer information
- *-Mconcur* has two new suboptions: *allcores* and *bind*.  
allcores     Use all available cores; specify at link time  
bind        Bind threads to cores; specify at link time
- *-Mprof* has a new suboption:  
[no]ccff    Enable [disable] CCFE information
- *-Msmartalloc* has a new suboption:  
hugebss    Put the BSS section in huge pages
- *-Mvect* has a new suboption:  
[no]short   Enable [disable] short vector operations.  
*-Mvect*=short enables generation of packed SSE instructions for  
short vector operations that arise from scalar code outside of loops  
or within the body of a loop iteration.

- `-mp` has 2 new suboptions:
  - `allcores` Use all available cores; specify at link time
  - `bind` Bind threads to cores; specify at link time
- `-tp` has 3 new target cpu types:
  - `shanghai` AMD Shanghai processor, 32-bit mode
  - `shanghai-32` AMD Shanghai processor, 32-bit mode
  - `shanghai-64` AMD Shanghai processor, 64-bit mode

## 3.2 Common Compiler Feedback Format (CCFF)

Using the Common Compiler Feedback Format (CCFF), PGI compilers save information about how your program was optimized, or why a particular optimization was not made, in the executable file. To append this information to the object file, use the compiler option `-Minfo=ccff`. Using the compiler option `-Mprof=ccff` also appends this information.

## 3.3 New or Modified OpenMP Features

OpenMP is a specification for a set of compiler directives, an application programming interface (API), and a set of environment variables that you can use to specify shared memory parallelism in FORTRAN and C/C++ programs. You may use OpenMP to obtain most of the parallel performance you can expect from your code, or to serve as a stepping stone to parallelizing an entire application with MPI.

PGI Visual Fortran 8.0 supports OpenMP 3.0. There are a number of enhancements that are associated with this support.

For more information and examples related to the directives, pragmas, clauses, run-time library routines, environment variables, and other information in this section, refer to Chapter 6 of the PVF User's Guide: *Using OpenMP*.

### 3.3.1 New or Modified OpenMP Directives and Pragmas

Every part of an OpenMP program is part of a task. In this release, PGI supports the following directives and clauses that are part of OpenMP 3.0.

- The Task directive, which defines an explicit task.
- The Taskwait directive, which specifies a wait on the completion of child tasks generated since the beginning of the current task.
- The Collapse (n) clause, associated with the DO...END DO, PARALLEL DO, and PARALLEL WORKSHARE directives, which specifies how many loops are associated with a loop construct.

### 3.3.2 New or Modified OpenMP Run-time Library Routines

PGI Visual Fortran 8.0 supports these new run-time library routines.

- **omp\_get\_ancestor\_thread\_num**  
This routine returns, for a given nested level of the current thread, the thread number of the ancestor.
- **omp\_get\_active\_level**  
This routine returns the number of enclosing active parallel regions enclosing the task that contains the call. PGI currently supports only one level of active parallel regions, so the return value currently is 1.
- **omp\_get\_level**  
This routine returns the number of parallel regions enclosing the task that contains the call.
- **omp\_get\_team\_size**  
This routine returns, for a given nested level of the current thread, the size of the thread team to which the ancestor belongs.
- **omp\_set\_schedule**  
This routine sets the value of the run\_sched\_var.
- **omp\_get\_schedule**  
This routine retrieves the value of the run\_sched\_var.

### 3.3.3 New or Modified OpenMP Environment Variables

PVF 8.0 has new or modified support for these OpenMP environment variables.

- **OMP\_MAX\_ACTIVE\_LEVELS**  
This variable currently has no effect. It typically enables (TRUE) or

disables (FALSE) nested parallelism.

- **OMP\_THREAD\_LIMIT**  
This variable, whose default is 64, specifies the absolute maximum number of threads that can be used in a program.
- **OMP\_STACKSIZE**  
Formally OMP\_STACK\_SIZE, this variable overrides the default stack size for a newly created thread.

## 3.4 New or Modified MPI Support

In this release, PVF 8.0-6, the local MPI capability continues to expand. PGI provides MPI support with PGI compilers and tools. You can build MPI applications on Windows using Microsoft's implementation of MPI, MSMPI.

For specific information about how to use MPI, refer to Chapter 7, *Using MPI*, of the PVF User's Guide.

The PGI Tools Guide describes the MPI-enabled tools in detail:

- PGPROF graphical MPI/OpenMP/multi-thread performance profiler.
- PGDBG graphical MPI/OpenMP/multi-thread symbolic debugger.

## 3.5 Static and Dynamic Linking with PVF / PGI Compilers

Prior to the 7.1 release, all executables were linked against the PGI runtime DLL called pg.dll and the multi-threaded DLL version of the Microsoft runtime libraries. All executables were therefore dependent upon pg.dll and the Microsoft runtime.

PVF 7.1 expanded these link-time options, and subsequent releases continue to support a choice between full static linking and dynamic linking. This change increases PVF's compatibility with other Fortran compilers on Windows.

The Fortran | Code Generation | Runtime Library property in a project's property pages determines which runtime library the project targets.

- For executable and static library projects, the default value of this property is **static linking** (*-Bstatic*). A statically-linked executable can be run on any system for which it is built; neither pg.dll nor the

Microsoft redistributable libraries need be installed on the target system.

- For dynamically linked library projects, the default value of this property is **dynamic linking** (*-Bdynamic*). A dynamically-linked executable can only be run on a system on which the PGI and Microsoft runtime redistributables have been installed.

For more information on deploying PGI-compiled applications to other systems, refer to the PVF User's Guide: *Distributing Files - Deployment*

### 3.5.1 Linking PVF and VC++ Projects

If you have multiple projects in a solution, be certain to use the same type of runtime library for all the projects. Further, if you have Microsoft VC++ projects in your solution, you need to be certain to match the runtime library types in the PVF projects to those of the VC++ projects.

PVF's property named Fortran | Code Generation | Runtime Library corresponds to the VC++ property named C/C++ | Code Generation | Runtime Library. The following table lists the appropriate combinations of Runtime Library property values when mixing PVF and VC++ projects.

If PVF Uses...	VC++ Should Use...
Multi-threaded (-Bstatic)	Multi-threaded (/MT)
Multi-threaded DLL (-Bdynamic)	Multi-threaded DLL (/MD)
Multi-threaded DLL (-Bdynamic)	Multi-threaded debug DLL (/MDd)

### 3.5.2 Common Link-time Errors

The runtime libraries specified for all projects in a solution should be the same. If both PVF and VC++ projects exist in the same solution, the runtime libraries targeted should be compatible.

The following examples provide a look at some of the link-time errors you might see when the runtime library targeted by a PVF project is not compatible with the runtime library targeted by a VC++ project. While the errors are easy to fix, they do require your action to resolve.

- Projects that produce DLLs should use the Multi-threaded DLL (*-Bdynamic*) runtime.
- Projects that produce executables or static libraries can use either type of linking.

To resolve these errors, refer to the table in Linking PVF and VC++ Projects and set the Runtime Library properties for the PVF and VC++ projects accordingly.

**Example 1:** Errors seen when linking a PVF project using *-Bstatic* and a VC++ library project using */MDd*:

```
MSVCRTD.lib(MSVCR80D.dll) : error LNK2005: _printf already
defined in libcmt.lib(printf.obj)
LINK : warning LNK4098: defaultlib 'MSVCRTD' conflicts with
use of other libs; use /NODEFAULTLIB:library
test.exe : fatal error LNK1169: one or more multiply defined
symbols found
```

**Example 2:** Errors seen when linking a PVF project using *-Bstatic* and a VC++ project using */MTd*:

```
LIBCMTD.lib(dbgheap.obj) : error LNK2005: _malloc already
defined in libcmt.lib(malloc.obj)
...
LINK : warning LNK4098: defaultlib 'LIBCMTD' conflicts with
use of other libs; use /NODEFAULTLIB:library
test.exe : fatal error LNK1169: one or more multiply defined
symbols found
```

### 3.5.3 Feature Tips

Often a feature involves changes that you might want to consider to enhance performance or make your work easier. This section contains some tips that might help you.

**Viewing Registers** – The registers window is available when debugging so you can see the value of the OS registers.

- To open the registers window during a debug session, select Debug | Windows | Registers. The first time you use the registers window, the CPU registers are shown by default.
- To show other register sets, follow these steps:
  1. Right-click in the registers window to bring up a context menu.
  2. From the context menu, select the group of registers to add to the registers window display.
- To remove a group from the display, follow these steps:
  1. Right-click in the registers window to bring up a context menu.
  2. From the context menu, deselect the group of registers to remove from the registers window display.

**Setting Fixed Format** – Some Fortran source is written in fixed-format style. If your fixed-format code does not compile, check that it is designated as fixed-format in PVF.

To do this, follow these steps:

1. Use the Solution Explorer to select a file:  
View | Solution Explorer.
2. Open the Properties Window:  
View | Other Windows | Properties Window.
3. From the dropdown list for the file property IsFixedFormat, select *True*.

## 3.6 Additional Library Interfaces

PGI provides access to a number of libraries that export C interfaces by using Fortran modules. Most of these libraries and functions are described in the *PVF User's Guide*.



# 4 Alternate Compiler Release Selection

---

Each release of PGI Visual Fortran contains two components - the newest release of PVF and the newest release of the PGI compilers and tools that PVF targets.

When PVF is installed onto a system that contains a previous version of PVF, the previous version of PVF is replaced. The previous version of the PGI compilers and tools, however, remains installed side-by-side with the new version of the PGI compilers and tools. By default, the new version of PVF will use the new version of the compilers and tools. Previous versions of the compilers and tools may be uninstalled using Control Panel | Add or Remove Programs.

There are two ways to use previous (6.2-3 or later) versions of the compilers:

- Use a different compiler release for a single project
- Use a different compiler release for all projects

The method to use depends on the situation.

## 4.1 For a Single Project

To use a different compiler release for a single project, you use the compiler flag `-V<ver>` to target the compiler with version `<ver>`.

For example, `-V7.2-4` causes the compiler driver to invoke the 7.2-4 version of the PGI compilers if these are installed.

To use this option within a PVF project, add it to the *Additional options* section of the Fortran | Command Line and Linker | Command Line property pages.

## 4.2 For All Projects

You can use a different compiler release for all projects.

The Tools | Options dialog within PVF contains entries that can be changed to use a previous version of the PGI compilers.

Under Projects and Solutions | PVF Directories, there are entries for Executable Directories, Include and Module Directories, and Library Directories.

- For the x64 platform, each of these entries includes a line containing \$(PGIToolsDir). To change the compilers used for the x64 platform, change each of the lines containing \$(PGIToolsDir) to contain the path to the desired bin, include, and lib directories.
- For the Win32 platform, these entries include a line containing \$(PGIToolsDir) on Win32 systems or \$(PGIToolsDir32) on Windows x64 systems. To change the compilers used for the Win32 platform, change each of the lines containing \$(PGIToolsDir) or \$(PGIToolsDir32) to contain the path to the desired bin, include, and lib directories.

**WARNING.** The debug engine in PVF 7.2 and 8.0 is not compatible with previous releases. If you use Tools | Options to target a release prior to 8.0, you cannot use PVF to debug. Instead, use the `-v` method described in section 4.1 to select an alternate compiler.

# 5 Distribution and Deployment

---

This chapter contains a number of topics that are related to using the compilers, including optimizing through the use of PGI Unified Binary technology, distributing applications, and customizing with site and user `rc` files.

- For more information on generating PGI Unified Binaries, including PGI Unified Binary command-line switches, directives, and pragmas, refer to Chapter 9, *Distributing Files – Deployment*, of the PVF User’s Guide.
- For more information and usage examples of the PGI compiler options that allow you to select static or dynamic linking, as well as information on using and creating static and dynamically-linked libraries, refer to Chapter 8, *Creating and Using Libraries*, of the *PVF User’s Guide*.
- For examples and information on customizing with `siterc` and user `rc` files to tailor a given installation for a particular purpose, refer to the PVF User’s Guide section: *Examples of Using `siterc` and User `rc` Files*.

## 5.1 Application Deployment and Redistributables

Programs built with PGI compilers may depend on run-time library files. These library files must be distributed with such programs to enable them to execute on systems where the PGI compilers are not installed. There are PGI redistributable files for all platforms. On Windows, PGI also supplies Microsoft redistributable files.

## 5.1.1 PGI Redistributables

PGI Visual Fortran 8.0-6 includes directories named REDIST. These directories contain all of the PGI dynamically linked libraries that can be re-distributed by PGI 8.0 licensees under the terms of the PGI End-User License Agreement (EULA). For reference, a copy of the PGI EULA in PDF form is included in the release.

The following paths for the REDIST directories assume 'C:' is the system drive.

- On a Win32 system, the REDIST directory is here:

```
C:\Program Files\PGI\win32\8.0-6\REDIST
```

- On a Win64 system, there are two REDIST directories:

```
C:\Program Files\PGI\win64\8.0-6\REDIST and
```

```
C:\Program Files (x86)\PGI\win32\8.0-6\REDIST
```

The REDIST directories contain the PGI runtime library DLLs for all supported targets. This enables users of the PGI compilers to create packages of executables and PGI runtime libraries that execute successfully on almost any PGI-supported target system, subject to the requirement that end-users of the executable have properly initialized their environment to use the relevant version of the PGI DLLs.

## 5.1.2 Microsoft Redistributables

PGI Visual Fortran includes Microsoft Open Tools, the essential tools and libraries required to compile, link, and execute programs on Windows. PVF 8.0 includes the latest version of the Microsoft Open Tools, version 9.

The Microsoft Open Tools directories contain a subdirectory named `redist`. PGI 8.0 licensees may redistribute the files contained in these directories in accordance with the terms of the PGI End-User License Agreement.

# 6 Known Limitations and Corrections

---

The following sections list known limitations, grouped by area. When possible, a workaround is provided.

## 6.1 PVF IDE Limitations

- Integration with source code revision control systems is not supported.
- PVF on Vista or Server 2008 may be unable to attach to an application running on a mapped network drive. If this occurs, run the application to which you want to attach on a local drive.
- When moving a project from one drive to another, all .d files for the project should be deleted and the whole project should be rebuilt. When moving a solution from one system to another, also delete the solution's Visual Studio Solution User Options file (.suo).
- The Resources property pages are limited. Use the Resources | Command Line property page to pass arguments to the resource compiler. Resource compiler output must be placed in the intermediate directory for build dependency checking to work properly on resource files.
- There are several properties that take paths or pathnames as values. In general, these may not work as expected if they are set to the project directory \$(ProjectDir) or if they are empty, unless empty is the default. Specifically:
  - *General / Output Directory* should not be empty or set to \$(ProjectDir).
  - *General / Intermediate Directory* should not be empty or set to \$(ProjectDir).
  - *Fortran / Output / Object File Name* should not be empty or set to \$(ProjectDir).
  - *Fortran / Output / Module Path* should not be empty or set to include \$(ProjectDir).

- Dragging and dropping files in the Solution Explorer that are currently open in the Editor may result in a file becoming “orphaned.” Close files before attempting to drag-and-drop them.
- Adding a resource file using Add New Item may cause PVF to open a dialog box stating that the file is already open. This dialog box can safely be ignored.

## 6.2 PVF Debugging Limitations

The following limitations apply to PVF debugging:

- Debugging of unified binaries is not fully supported. The names of some subprograms are modified in the creation of the unified binary, and the PVF debug engine does not translate these names back to the names used in the application source code. For more information on how to debug a unified binary, see [www.pggroup.com/support/tools.htm](http://www.pggroup.com/support/tools.htm).
- In some situations, using the Watch Window may be unreliable for local variables. Calling a function or subroutine from within the scope of the watched local variable may cause missed events and/or false positive events. Local variables may be watched reliably if program scope does not leave the scope of the watched variable.

## 6.3 Microsoft Visual Studio Add-in Limitations for SUA

A bug exists in the Visual Studio Debugger Add-in that is part of the "Utilities and SDK for UNIX-based Applications" installation. Microsoft confirmed that when this add-in is installed and enabled, it causes the executable of a Visual Studio solution to be inaccessible after a debug session terminates. When an executable is inaccessible, it cannot be deleted. Further, attempting to delete the inaccessible executable – by building, rebuilding, or cleaning a PVF or other Visual Studio project – fails, causing the error: "Access is Denied." The only way to free access to an inaccessible executable is to exit Visual Studio.

The VS Debugger Add-in is required to use Visual Studio to debug applications compiled for SUA using Microsoft's compilers. However, this add-in is *not* required to use PVF to debug applications compiled for SUA using the PGI Fortran compilers. If the add-in is not required, it can be disabled using the VS Add-in Manager, thus preventing the bug from occurring.

To disable the add-in, follow these steps:

- 1) From the Visual Studio Tools menu, open the Add-in Manager.
- 2) Select "VSAddin" from the list in "Available Add-ins."  
The description is: "VSAddin - Extension to the VS Debugger to support debugging UNIX-based Applications."
- 3) Clear the check-boxes for the VSAddin.
- 4) Click "OK" to close the Add-in Manager.
- 5) Exit Visual Studio.

When you restart Visual Studio, VSAddin is disabled.

## 6.4 PGI Compiler Limitations

The frequently asked questions (FAQ) section of the *pgroup.com* web page at [www.pgroup.com/support/index.htm](http://www.pgroup.com/support/index.htm) provides more up to date information about the state of the current release.

- If an executable is linked with any PVF-compiled DLL, the PVF runtime library DLLs must be used (in particular the static libraries cannot be used). To accomplish this, use the compiler option *-Bdynamic* when creating the executable.
- Do not use *-Mprof* with PVF runtime library DLLs. To build an executable for profiling, use the static libraries. The static libraries will be used by default in the absence of *-Bdynamic*.
- The *-i8* option can make programs incompatible with MPI; use of any INTEGER\*8 array size arguments can cause failures with these libraries.
- The *-i8* option can make programs incompatible with the bundled ACML library. Visit *developer.amd.com* to check for compatible libraries.
- Using *-Mprof=func* and *-mp* together with any of the PGI compilers can result in segmentation faults by the generated executable. These options should not be used together.
- Using *-Mpf* and *-mp* together is not supported. The *-Mpf* flag will disable *-mp* at compile time, which can cause run-time errors in programs that depend on interpretation of OpenMP directives or pragmas. Programs that do not depend on OpenMP processing for correctness can still use profile feedback. The *-Mpf* flag does not disable OpenMP processing.
- ACML 4.1-0 is built using the *-fastsse* compile/link option, which includes *-Mcache\_align*. When linking with ACML on Win32, all program units must be compiled with *-Mcache\_align*, or an aggregate

option such as `-fastsse` which incorporates `-Mcache_align`. This is not an issue on 64-bit targets where the stack is 16-byte aligned by default. The lower performance but fully portable *blas* and *lapack* libraries can be used on CPUs that do not support SSE instructions.

## 6.5 Documentation Corrections

### 6.5.1 Functions `ctime()` and `time()`

The 3F Functions section of the PGI Fortran Reference incorrectly indicates the return values for the `ctime()` and `time()` functions. The following information is correct:

`time`

**Return** system time.

**Synopsis**

```
integer*8 function time()
```

**Description**

Return the time since 00:00:00 GMT, January 1, 1970, measured in seconds.

`ctime`

**Return** system time

**Synopsis**

```
character*(*) function ctime(stime)
integer*8 stime
```

**Description**

`ctime` converts a system time in `stime` to its ASCII form and returns the converted form. Neither newline nor NULL is included.

### 6.5.2 Message Text Errors

The PVF User's Guide incorrectly lists the text information in messages W170 through W183 as related to F90 extension. The text in these messages should state: PGI Fortran extension.

### 6.5.3 FORTRANOPT Environment Variable

The PGI User's Guide incorrectly lists the name of the FORTRANOPT variable as `FORTRAN_OPT`. Further, the following statement is wrong:

*In a Windows environment, if `FORTTRAN_OPT` exists and contains the value `crif`, a sequential formatted or list-directed record is allowed to be terminated with the character sequence `\r\n` (carriage return, newline).*

The statement should read:

*In a **non-Windows** environment, if **FORTRANOPT** exists and contains the value `crif`, a sequential formatted or list-directed record is allowed to be terminated with the character sequence `\r\n` (carriage return, newline).*

## 6.6 Corrections

Some issues were reported in PGI 7.2 compilers or previous releases. Problems found in the PGI 7.2 compilers may not have occurred in previous releases.

Refer to [www.pgroup.com/support/release\\_tprs.htm](http://www.pgroup.com/support/release_tprs.htm) for a complete and up-to-date table of technical problem reports, TPRs, fixed in recent releases of the PGI compilers and tools. The table contains a summary description of each problem as well as the release in which it was fixed.



# 7 Contact Information and Documentation

---

You can contact The Portland Group at:

*The Portland Group  
STMicroelectronics, Inc.  
Two Centerpointe Drive  
Lake Oswego, OR 97035 USA*

Or contact us electronically using any of the following means:

*Fax: +1-503-682-2637  
Sales: sales@pgroup.com  
Support: trs@pgroup.com  
Web: [www.pgroup.com](http://www.pgroup.com)*

The PGI User Forum is monitored by members of the PVF development team and may contain answers to commonly asked questions. Log in to the PGI website to access the forum:

[www.pgroup.com/userforum/index.php](http://www.pgroup.com/userforum/index.php)

Technical support requests are submitted by email or by using an online form at:

[www.pgroup.com/support](http://www.pgroup.com/support)

Phone support is not currently available. Documentation is available from the Help Menu within PGI Visual Fortran and from the Start menu.

For PGI Visual Fortran 2008:

Start | All Programs | PGI Visual Fortran 2008 | PGI Documentation | PVF Help.

For PGI Visual Fortran 2005:

Start | All Programs | PGI Visual Fortran 2005 | PGI Documentation | PVF Help.

Additional PGI documentation is available online at

[www.pgroup.com/doc](http://www.pgroup.com/doc)