PGI Visual Fortran
For Multi-core x64+Accelerator Workstations and Clusters

PGI Visual Fortran® (PVF®) brings the PGI suite of high-performance 64-bit and 32-bit parallel Fortran compilers to Windows developers using Microsoft Visual Studio.

High-performance PGI Compilers
PVF features the latest version of PGI’s high performance optimizing Fortran 2003 compilers. PGI compilers are supported on and can generate fully optimized code for a broad range of popular high-performance computing systems based on 64-bit x64 (AMD64 and Intel 64) and 32-bit x86 processors, and HPC accelerators from Intel, NVIDIA and AMD.

PGI Fortran compilers offer world-class performance and features including both automatic and OpenMP 3.1 parallelization support for optimizing performance on multi-core processors. State-of-the-art compiler technologies found in PVF include vectorization, parallelization, interprocedural analysis, memory hierarchy optimization, cross file function inlining, CPU-specific optimization and more. PGI Unified Binary™ technology streamlines cross-platform support by combining into a single executable file code optimized for any number of different processors and accelerators.

Comprehensive GPU Support
PGI Visual Fortran includes PGI’s two models for programming accelerators. The PGI Accelerator™ with OpenACC directives is a high-level implicit programming model similar to OpenMP for multi-core x64 systems. PGI Accelerator Fortran enables the incremental offloading of compute-intensive loops and code regions from a host CPU to an accelerator using simple compiler directives.

Developed in cooperation with NVIDIA, CUDA Fortran extensions within PVF give expert programmers direct control of all aspects of GPU accelerator programming.

Visual Studio Integration
PVF includes features to enable rapid code development. PVF adds Fortran syntax coloring, Fortran intrinsics tips and keyword completion to the Visual Studio editor. PVF includes a complete Fortran build environment for Visual Studio with Fortran specific features such as automatic build dependency management. PVF is interoperable with Microsoft Visual C++ so Fortran sub-programs can call C functions and vice versa.

PVF augments the Visual Studio debugger with a custom debug engine that provides Fortran-specific debugging capabilities. The PVF debug engine supports debugging of single thread, multi-thread, OpenMP, MSMPI, and hybrid MSMPI+OpenMP Fortran applications. It enables debugging of applications using source code or with assembly code, and it provides full access to the registers and hardware state of the processor. The PVF debug engine is also interoperable with Visual C++.

PVF can manage the multi-process MPI capabilities built into Microsoft HPC Server. Use PVF to configure compile-time options, launch and debug programs either locally on a workstation or on a distributed-memory Windows cluster.


A Complete Development Solution

License options include single user node-locked and multi-user network floating.
### PGI Visual Fortran

#### Visual Studio Integration

- Fortran syntax coloring
- Fortran intrinsics tips
- Keyword completion
- Fortran debug engine
- Automatic dependency analysis
- Interoperable with Microsoft Visual C++
- Sample projects

#### Fortran Compilers

**Features**

- Full 64-bit support on AMD64 and Intel 64
- Intel co-processor, AMD APU and NVIDIA GPU support
- PGI Unified Binary™ contains code optimized for AMD64, Intel 64 and HPC accelerators
- Auto-parallelization
- OpenMP 3.1 parallelization
- Vectorization/Optimization directives

**Optimizations**

- AMD64 and Intel 64 cross-target optimization
- Interprocedural Analysis (IPA)
- SSE/SIMD vectorization
- Function inlining
- Loop fusion
- Profile feedback optimization
- Memory hierarchy and memory allocation optimizations
- Huge page support
- Instruction scheduling
- Invariant code and conditional hoisting
- Common sub-expression elimination (CSE)

**Vector/Parallel Transformations**

- State-of-the-art dependence analysis
- Nested loop autotemparalelization
- Nested loop vectorization
- Loop interchange
- Loop splitting, unroll & jam
- Iteration peeling
- Aligned access optimizations

- Scalar promotion
- Temporary vector creation
- Stripping, Cache Tiling
- Concurrent call support
- Inline concurrent code segments
- Single- and double-precision Streaming SIMD Extensions
- Software prefetching

#### Parallel Debug Engine

- 64-bit and 32-bit targets
- Debug Fortran and Visual C++
- Debug parallel OpenMP, MSMPI and hybrid MSMPI+OpenMP programs
- Variable rollover
- One touch breakpoint setting

- Step into, out, over of functions
- Attach to running processes
- One touch symbolic display
- Multiple format display of values or strings
- Track register states

---

* The registered trademarks and marks are the property of their respective owners.

PVF-1012-01