
Intel(R) MPI Library 4.0 Update 3 for Linux* OS
Release Notes

Contents

- Overview
- What's New
- Key Features
- System Requirements
- Installation Notes
- Documentation
- Special Features and Known Issues
- Technical Support
- Copyright and Licenses
- Disclaimer and Legal Information

Overview

The Intel(R) MPI Library for Linux* OS is a multi-fabric message passing library based on ANL* MPICH2* and OSU* MVAPICH2*.

The Intel(R) MPI Library for Linux* OS implements the Message Passing Interface, version 2.1 (MPI-2.1) specification.

To receive technical support and updates, you need to register your Intel(R) Software Development Product. See the Technical Support section.

Product Contents

The Intel(R) MPI Library Runtime Environment (RTO) contains the tools you need to run programs including MPD daemons and supporting utilities, shared (.so) libraries, and documentation.

The Intel(R) MPI Library Development Kit (SDK) includes all of the Runtime Environment components and compilation tools: compiler commands (mpicc, mpiicc, etc.), include files and modules, static (.a) libraries, debug libraries, trace libraries, and test codes.

Related Products and Services

Information on Intel(R) Software Development Products is available at <http://www.intel.com/software/products>.

Some of the related products include:

- The Intel(R) Software College provides training for developers on leading-edge software development technologies. The training consists of online and

instructor-led courses covering all Intel(R) architectures, platforms, tools, and technologies.

What's New

The Intel(R) MPI Library 4.0 Update 3 for Linux* OS is an update release of the Intel(R) MPI Library for Linux* OS.

This release includes the following updates compared to the Intel(R) MPI Library 4.0 Update 2 (see product documentation for more details):

- Performance and scalability improvements
 - o New scalable process manager mpiexec.hydra used by default in the mpirun utility
 - o Shared memory optimizations for platforms with Intel(R) Streaming SIMD Extensions 4.2 (Intel(R) SSE4.2) and Intel(R) AES New Instructions (Intel(R) AES-NI). This functionality is available for both Intel(R) and non-Intel microprocessors, but it may perform additional optimizations for Intel microprocessors than it performs for non-Intel microprocessors.
 - o Dynamic connection mode for shared memory
 - o Scalable hybrid UD/RDMA mode for the DAPL fabric
 - o Accelerated RDMA memory registration cache
 - o Dynamic queue pair (QP) creation and extensible reliable connection (XRC) mode support for the OFA fabric
 - o RDMA over converged ethernet (RoCE) support through the DAPL fabric
 - o TCP scalability improvements
 - o Substantially accelerated and enhanced MPI tuning utility
- Usability improvements
 - o Additional integrated performance monitoring (IPM) statistics summary format
 - o Extended debugging output control
 - o Enhanced processor information utility (cpuinfo)
 - o Bug fixes
- Extended interoperability
 - o Intel(R) Composer XE 2011 Update 6 support
 - o Tight integration with SLURM* job management systems through the mpiexec.hydra process manager

The Intel(R) MPI Library 4.0 Update 2 for Linux* OS is an update release of the Intel(R) MPI Library for Linux* OS.

This release includes the following updates compared to the Intel(R) MPI Library 4.0 Update 1 (see product documentation for more details):

- Usability improvements
 - o Support for SGI* Altix* UV* 1000 pinning with more than 64 cores
 - o Improved static DAPL connections establishment in the wait mode
 - o Improved stability of the shm:ofa fabric
 - o Improved mpiexec.hydra process manager support for SLURM and Cloud
 - o Static libraries compiled using the -fPIC option
 - o Improved error reporting for the Lustre* file system
 - o Bug fixes

- Extended interoperability
 - o Intel(R) Composer XE 2011 Update 4 support
 - o Ability to call MPI from the Co-Array Fortran programs

The Intel(R) MPI Library 4.0 Update 1 for Linux* OS is an update release of the Intel(R) MPI Library for Linux* OS.

This release includes the following updates compared to the Intel(R) MPI Library 4.0 (see product documentation for more details):

- Performance and scalability improvements
 - o Improved startup scalability through the mpiexec.hydra process manager
 - o Improved OFA fabric performance
 - o Further optimizations to several collective algorithms
- Usability improvements
 - o Use of ssh for remote connectivity by default (formerly rsh)
 - o Process pinning support for the mpiexec.hydra process manager
 - o Extended process pinning control for hybrid applications through the I_MPI_PIN_DOMAIN and I_MPI_PIN_CELL environment variables
 - o Improved mpitune for easier application tuning
- Extended interoperability
 - o Intel(R) Composer XE 12.0 Beta support

The Intel(R) MPI Library 4.0 for Linux* OS includes the following new features compared to the Intel(R) MPI Library 3.2 Update 2 (see product documentation for more details):

- New architecture for better performance and higher scalability
 - o Optimized shared memory path for industry leading latency on multicore platforms
 - o New flexible mechanism for selecting the communication fabrics (I_MPI_FABRICS) that complements the classic Intel MPI device selection method (I_MPI_DEVICE)
 - o Native InfiniBand* interface (OFED* verbs) support with multirail capability for ultimate InfiniBand* performance
 - Set I_MPI_FABRICS=ofa for OFED* verbs only
 - Set I_MPI_FABRICS=shm:ofa for shared memory and OFED* verbs
 - Set I_MPI_OFA_NUM_ADAPTERS, etc., for multirail transfers
 - o Tag Matching Interface (TMI) support for higher performance of Qlogic* PSM* and Myricom* MX* interconnect interfaces
 - Set I_MPI_FABRICS=tmi for TMI only
 - Set I_MPI_FABRICS=shm:tmi for shared memory and TMI
 - o Connectionless DAPL* UD support for limitless scalability of your TOP500 submissions
 - Set I_MPI_FABRICS=dapl for DAPL only
 - Set I_MPI_FABRICS=shm:dapl for shared memory and DAPL
 - Set I_MPI_DAPL_UD=enable for DAPL UD transfers over DAPL fabric
- Updated MPI performance tuner to extract the last ounce of performance out of your installation
 - o For a certain cluster, based on the Intel(R) MPI Benchmarks (IMB) or a user provided benchmark
 - o For a certain application run
- MPI 2.1 standard conformance
- Experimental dynamic process support

- Experimental fault tolerance support
- Experimental failover support
- Backward compatibility with Intel MPI Library 3.x based applications
- Man pages

Examples

Set the `I_MPI_FABRICS` environment variable to select a particular network fabric.

- To use shared memory for intra-node communication, and TMI for inter-node communication, do the following steps:

1. Copy the `<installdir>/etc64/tmi.conf` file to the `/etc` directory.
Alternatively you set the `TMI_CONFIG` environment variable to point to the location of the `tmi.conf` file. For instance,

```
$ export TMI_CONFIG=<installdir>/etc64/tmi.conf
```
2. Select `shm:tmi` for your fabric. For instance,

```
$ export I_MPI_FABRICS=shm:tmi
```
3. Execute an application. For instance,

```
$ mpiexec -n 16 ./IMB-MPI1
```

Set the `I_MPI_TMI_PROVIDER` environment variable if necessary to select a specific TMI provider. For instance,

```
$export I_MPI_TMI_PROVIDER=psm
```

Make sure that you have the `libtmi.so` library in the search path of the "ldd" command.

- To select shared memory for intra-node communication and OFED* verbs for inter-node communication, do the following steps:

```
$ export I_MPI_FABRICS=shm:ofa
$ mpiexec -n 4 ./IMB-MPI1
```

Set the `I_MPI_OFA_NUM_ADAPTERS` environment variable to utilize the multirail capabilities.

```
$ export I_MPI_FABRICS=shm:ofa
$ export I_MPI_OFA_NUM_ADAPTERS=2
$ mpiexec -n 4 ./IMB-MPI1
```

- To use shared memory for intra-node communication and the DAPL* layer for inter-node communication, do the following steps:

```
$ export I_MPI_FABRICS=shm:dapl
$ mpiexec -n 4 ./IMB-MPI1
```

Set the `I_MPI_DAPL_UD` environment variable to enable connectionless DAPL* UD.

```
$ export I_MPI_FABRICS=shm:dapl
$ export I_MPI_DAPL_UD=enable
$ mpiexec -n 4 ./IMB-MPI1
```

See more details in the Intel(R) MPI Library for Linux* OS Reference Manual.

----- Key Features -----

This release of the Intel(R) MPI Library supports the following major features:

- MPI-1 and MPI-2.1 specification conformance
- Support for any combination of the following interconnection fabrics:
 - o Shared memory
 - o Network fabrics with tag matching capabilities through Tag Matching Interface (TMI), such as Qlogic* Infiniband*, Myrinet* and other interconnects
 - o Native InfiniBand* interface through OFED* verbs provided by Open Fabrics Alliance* (OFA*)
 - o RDMA-capable network fabrics through DAPL*, such as InfiniBand* and Myrinet*
 - o Sockets, for example, TCP/IP over Ethernet*, Gigabit Ethernet*, and other interconnects
- (SDK only) Support for IA-32 and Intel(R) 64 architecture clusters using:
 - o Intel(R) C++ Compiler for Linux* OS version 11.1 through 12.1 and higher
 - o Intel(R) Fortran Compiler for Linux* OS version 11.1 through 12.1 and higher
 - o GNU* C, C++ and Fortran 95 compilers
- (SDK only) C, C++, Fortran 77 and Fortran 90 language bindings
- (SDK only) Dynamic or static linking

----- System Requirements -----

The following sections describe supported hardware and software.

Supported Hardware -----

Systems based on the IA-32 architecture:

- A system based on the Intel(R) Pentium(R) 4 processor or higher
- Intel(R) Core(TM) i7 processor recommended
- 1 GB of RAM per core
- 2 GB of RAM per core recommended
- 1 GB of free hard disk space

Systems based on the Intel(R) 64 architecture:

- Intel(R) Core(TM) processor family or higher
- Intel(R) Xeon(R) 5500 processor series recommended
- 1 GB of RAM per core
- 2 GB of RAM per core recommended
- 1 GB of free hard disk space

Supported Software -----

Operating Systems:

Systems based on the IA-32 architecture:

- Red Hat* Enterprise Linux* 5,
- Red Hat* Enterprise Linux* 6,

SuSE* Linux Enterprise Server* 10,
SuSE* Linux Enterprise Server* 11

Systems based on the Intel(R) 64 architecture:

Red Hat* Enterprise Linux* 5,
Red Hat* Enterprise Linux* 6,
Fedora* 15
CentOS* 5.3,
SuSE* Linux Enterprise Server* 10,
SuSE* Linux Enterprise Server* 11,
openSuSE* Linux* 11.1

(SDK only) Compilers:

GNU*: C, C++, Fortran 77 3.3 or higher, Fortran 95 4.0 or higher

Intel(R) C++ Compiler for Linux* OS 11.1 through 12.1 or higher
Intel(R) Fortran Compiler for Linux* OS 11.1 through 12.1 or higher

(SDK only) Supported Debuggers:

Intel(R) Debugger 9.1-23 or higher
Rogue* Wave* Software* TotalView* 6.8 or higher
Allinea* DDT* v1.9.2 or higher
GNU* Debuggers

Batch Systems:

Platform* LSF* 6.1 or higher
Altair* PBS Pro* 7.1 or higher
Torque* 1.2.0 or higher
Parallelnavi* NQS* for Linux* OS V2.0L10 or higher
Parallelnavi for Linux* OS Advanced Edition V1.0L10A or higher
NetBatch* v6.x or higher
SLURM* 1.2.21 or higher
Sun* Grid Engine* 6.1 or higher
IBM* LoadLeveler* 4.1.1.5 or higher
Platform* Lava* 1.0

Recommended InfiniBand Software:

- OpenFabrics* Enterprise Distribution (OFED*) 1.4 or higher.

Additional Software:

- Python* 2.2 or higher, including the python-xml module. Python* distributions are available for download from your OS vendor or at <http://www.python.org> (for Python* source distributions).
- An XML parser such as expat* or pyxml*.
- If using InfiniBand*, Myrinet*, or other RDMA-capable network fabrics, a DAPL* 1.2 standard-compliant provider library/driver is required. DAPL* providers are typically provided with your network fabric hardware and software.

(SDK only) Supported Languages

For GNU* compilers: C, C++, Fortran 77, Fortran 95

For Intel compilers: C, C++, Fortran 77, Fortran 90, Fortran 95

Installation Notes

See the Intel(R) MPI Library for Linux* OS Installation Guide for details.

Documentation

The Intel(R) MPI Library for Linux* OS Getting Started Guide, found in `Getting_Started.pdf`, contains information on the following subjects:

- First steps using the Intel(R) MPI Library for Linux* OS
- First-aid troubleshooting actions

The Intel(R) MPI Library for Linux* OS Reference Manual, found in `Reference_Manual.pdf`, contains information on the following subjects:

- Command Reference describes commands, options, and environment variables
- Tuning Reference describes environment variables that influence library behavior and performance

The Intel(R) MPI Library for Linux* OS Installation Guide, found in `INSTALL.html`, contains information on the following subjects:

- Obtaining, installing, and uninstalling the Intel(R) MPI Library
- Getting technical support

Special Features and Known Issues

- Intel(R) MPI Library 4.0 for Linux* OS is binary compatible with the majority of Intel MPI Library 3.x-based applications. Recompile your application only if you use:
 - o MPI one-sided routines in Fortran (`mpi_accumulate()`, `mpi_alloc_mem()`, `mpi_get()`, `mpi_put()`, `mpi_win_create()`)
 - o MPI C++ binding
- Intel(R) MPI Library 4.0 for Linux* OS implements the MPI-2.1 standard. The functions of the following MPI routines have changed:
 - o `MPI_Cart_create()`
 - o `MPI_Cart_map()`
 - o `MPI_Cart_sub()`
 - o `MPI_Graph_create()`

If your application depends on the strict pre-MPI-2.1 behavior, set the environment variable `I_MPI_COMPATIBILITY` to "3".

- The following features are currently available only on Intel(R) 64 architecture:
 - o Native InfiniBand* interface (OFED* verbs) support
 - o Multirail capability
 - o Tag Matching Interface (TMI) support
 - o Connectionless DAPL* UD support
- The Intel(R) MPI Library supports the MPI-2 process model for all fabric combinations with the following exceptions:
 - o `I_MPI_FABRICS` is set to `<fabric1>:<fabric2>`, where `<fabric1>` is not `shm`, and `<fabric2>` is not equal to `<fabric1>` (for example, `dapl:tcp`)
- If communication between two existing MPI applications is established using the process attachment mechanism, the library does not control whether the same fabric has been selected for each application. This situation may cause unexpected applications behavior. Set the `I_MPI_FABRICS` variable to the same values for each application to avoid this issue.
- The following restriction exists for the DAPL-capable network fabrics because it relates to support for the MPI-2 process model: if the size of the information about the host used to establish the communication exceeds a certain DAPL provider value, the application fails with an error message similar to:


```
[0:host1][./../dapl_module_util.c:397] error(0x80060028):....: could not\
connect DAPL endpoints: DAT_INVALID_PARAMETER(DAT_INVALID_ARG5)
```
- The Intel(R) MPI Library Development Kit package is layered on top of the Runtime Environment package. See the Intel(R) MPI Library for Linux* OS Installation Guide for more details.
- The SDK installer checks for the existence of the associated RTO package and installs it if the RTO is missing. If the RTO is already present, its location determines the default SDK location.
- The RTO uninstaller checks for SDK presence and proposes to uninstall the SDK and RTO packages.
- The SDK uninstaller asks the user if the RTO is to be uninstalled as well. The user is able to cancel the uninstallation at this point.
- The Intel(R) MPI Library automatically places consecutive MPI processes onto all processor cores. Use the `mpiexec -perhost 1` option or set the `I_MPI_PERHOST` environment variable to 1 in order to obtain the round robin process placement.
- The Intel(R) MPI Library pins processes automatically. Use `I_MPI_PIN` and related environment variables to control process pinning. See the Intel(R) MPI Library for Linux* OS Reference Manual for more details.

- Always set the I_MPI_PIN_DOMAIN environment variable to "auto" when using the new process manager mpiexec.hydra.
- The Intel(R) MPI Library provides thread-safe libraries up to level MPI_THREAD_MULTIPLE. The default level is MPI_THREAD_FUNNELED. Follow these rules:
 - o (SDK only) Use the Intel(R) MPI compiler driver option -mt_mpi to build a thread-safe MPI application.
 - o Do not load thread-safe Intel(R) MPI libraries through dlopen(3).
- To run a mixed Intel MPI/OpenMP* application, follow these steps:
 - o Use the thread-safe version of the Intel(R) MPI Library by using the -mt_mpi compiler driver option.
 - o Set I_MPI_PIN_DOMAIN to select the desired process pinning scheme. The recommended setting is I_MPI_PIN_DOMAIN=auto. See the Intel(R) MPI Library for Linux* OS Reference Manual for more details.
- Intel(R) MKL 10.0 may create multiple threads depending on various conditions. Follow these rules to correctly use Intel(R) MKL:
 - o (SDK only) Use the thread safe version of the Intel(R) MPI Library in conjunction with Intel(R) MKL by using the -mt_mpi compiler driver option
 - o Set the OMP_NUM_THREADS environment variable to 1 to run the application if linked with the non-thread-safe version of the Intel(R) MPI Library
- The Intel(R) MPI Library uses dynamic connection establishment by default for 64 and more processes. To always establish all connections upfront, set the I_MPI_DYNAMIC_CONNECTION environment variable to "disable".
- The Intel(R) MPI Library compiler drivers embed the actual Development Kit library path (default /opt/intel/impi/<version>.<package_num>) and default Runtime Environment library path /opt/intel/mpi-rt/<version>.<package_num> into the executables using the -rpath linker option.
- Use the LD_PRELOAD environment variable to preload the appropriate Intel(R) MPI binding library to start an MPICH2 Fortran application in the Intel(R) MPI Library environment.
- The Intel(R) MPI Library enhances message-passing performance on DAPL*-based interconnects by maintaining a cache of virtual-to-physical address translations in the MPI DAPL* data transfer path.

Set the environment variable LD_DYNAMIC_WEAK to "1" if your program dynamically loads the standard C library before dynamically loading the Intel(R) MPI Library. Alternatively, use the environment variable LD_PRELOAD to load the Intel(R) MPI Library first.

To disable the translation cache completely, set the environment variable I_MPI_RDMA_TRANSLATION_CACHE to "disable". Note that you do not need to set the aforementioned environment variables LD_DYNAMIC_WEAK or LD_PRELOAD when you disable the translation cache.

- (SDK only) Always link the standard libc libraries dynamically if you use the DAPL, OFA*, and TMI fabrics, individually or in combination with the shared memory fabric, to avoid possible segmentation faults.

Note: some compilers may use the -static option implicitly, for example, when using the -fast option for the Intel compilers. Therefore, use the ldd command to verify that the final executable is dynamically linked with the standard libc libraries.

It is safe to link the Intel(R) MPI Library statically through the -static_mpi option of the compiler drivers. This option does not affect the default linkage method for other libraries.

- Certain DAPL* providers may not work or provide worthwhile performance with the Intel(R) MPI Library for Linux* OS, for example:
 - o Qlogic*. Use the TMI libraries included with the Intel(R) MPI Library when running over the Qlogic* PSM* interconnect interface for best performance.
 - o Myricom*. Use the TMI libraries included with the Intel(R) MPI Library when running over the Myricom* MX* interconnect interface for best performance.
Alternatively, contact Myricom* or download the DAPL* provider at <http://sourceforge.net/projects/dapl-myri-net> which supports both the GM* and MX* interfaces.
- The GM DAPL* provider may not work with the Intel(R) MPI Library for Linux* OS using some versions of the GM* drivers. Set I_MPI_RDMA_RNDV_WRITE=1 to avoid this issue.
- Certain DAPL* providers may not function properly if your application uses system(3), fork(2), vfork(2), or clone(2) system calls. Do not use these system calls or functions based upon them. For example, system(3), with:
 - o OFED* DAPL* provider with Linux* kernel version earlier than official version 2.6.16. Set the RDMAV_FORK_SAFE environment variable to enable the OFED workaround with compatible kernel version.
- The Intel(R) MPI Library does not support heterogeneous clusters of mixed architectures and/or operating environments.
- The Intel(R) MPI Library requires Python* 2.2 or higher for process management.
- The Intel(R) MPI Library requires the python-xml* package or its equivalent on each node in the cluster for process management. For example, the following operating system does not have this package installed by default:
 - o SuSE* Linux Enterprise Server* 9
- The Intel(R) MPI Library requires the expat* or pyxml* package, or an equivalent XML parser on each node in the cluster for process management.
- The following MPI-2.1 features are not supported by the Intel(R) MPI Library:
 - o Passive target one-sided communication when the target process does not

call any MPI functions

- If installation of the Intel(R) MPI Library package fails and shows the error message: "Intel(R) MPI Library already installed" when a package is not actually installed, try the following:

1. Determine the package number that the system believes is installed by typing:

```
# rpm -qa | grep intel-mpi
```

This command returns an Intel(R) MPI Library <package name>.

2. Remove the package from the system by typing:

```
# rpm -e <package name>
```

3. Re-run the Intel(R) MPI Library installer to install the package.

TIP:

To avoid installation errors, always remove the Intel(R) MPI Library packages using the uninstall script provided with the package before trying to install a new package or reinstall an older one.

- Due to an installer limitation, avoid installing earlier releases of the Intel(R) MPI Library packages after having already installed the current release. It may corrupt the installation of the current release and require that you uninstall/reinstall it.
- Certain operating system versions have a bug in the rpm command that prevents installations other than in the default install location. In this case, the installer does not offer the option to install in an alternate location.
- If the mpdboot command fails to start up the MPD, verify that the Intel(R) MPI Library package is installed in the same path/location on all the nodes in the cluster. To solve this problem, uninstall and re-install the Intel(R) MPI Library package while using the same <installdir> path on all nodes in the cluster.
- If the mpdboot command fails to start up the MPD, verify that all cluster nodes have the same Python* version installed. To avoid this issue, always install the same Python* version on all cluster nodes.
- Presence of environment variables with non-printable characters in user environment settings may cause the process startup to fail. To work around this issue, the Intel(R) MPI Library does not propagate environment variables with non-printable characters across the MPD ring.
- A program cannot be executed when it resides in the current directory but "." is not in the PATH. To avoid this error, either add "." to the PATH on ALL nodes in the cluster or use the explicit path to the executable or ./<executable> in the mpiexec command line.

- The Intel(R) MPI Library 2.0 and higher supports PMI wire protocol version 1.1. Note that this information is specified as

```
pmi_version = 1  
pmi_subversion = 1
```

instead of

```
pmi_version = 1.1
```

as done by the Intel(R) MPI Library 1.0.

- The Intel(R) MPI Library requires the presence of the /dev/shm device in the system. To avoid failures related to the inability to create a shared memory segment, make sure the /dev/shm device is set up correctly.
- The Intel(R) MPI Library uses TCP sockets to pass stdin stream to the application. If you redirect a large file, for example, 5KB, the transfer could take a long time and cause things to hang on the remote side. To avoid this issue, pass large files to the application as command line options.
- (SDK only) Certain operating systems use GNU* compilers version 4.2 or higher that is incompatible with Intel(R) Professional Edition Compiler 9.1. Use Intel(R) Professional Edition Compilers 10.1 or later on the respective operating systems, for example:
 - o SuSE* Linux Enterprise Server* 11
- (SDK only) Certain GNU* C compilers may generate code that leads to inadvertent merging of some output lines at runtime. This happens when different processes write simultaneously to the standard output and standard error streams. In order to avoid this, use the -fno-builtin-printf option of the respective GNU* compiler while building your application.
- (SDK only) Certain versions of the GNU* LIBC library define free()/realloc() symbols as non-weak. Use the --allow-multiple-definition GNU* linker option to link your application.
- (SDK only) A known exception handling incompatibility exists between GNU C++ compilers version 3.x and version 4.x. Use the special -gcc-version=<nnn> option for the compiler drivers mpicxx and mpiicpc to link an application when running in a particular GNU* C++ environment. The valid <nnn> values are:
 - o 320 if GNU* C++ version is 3.2.x
 - o 330 if GNU* C++ version is 3.3.x
 - o 340 if GNU* C++ version is 3.4.x
 - o 400 if GNU* C++ version is 4.0.x
 - o 410 if GNU* C++ version is 4.1.x
 - o 420 if GNU* C++ version is 4.2.x
 - o 430 if GNU* C++ version is 4.3.x

A library compatible with the detected version of the GNU* C++ compiler is used by default. Do not use this option if the gcc version is older

than 3.2.

- (SDK only) The Fortran 77 and Fortran 90 tests in the <installdir>/test directory may produce warnings when compiled with the mpif77, etc. compiler commands. You can safely ignore these warnings, or add the -w option to the compiler command line to suppress them.
- (SDK only) In order to use GNU Fortran compiler version 4.0 and higher use the mpif90 compiler driver.
- (SDK only) A known module file format incompatibility exists between the GNU Fortran 95 compilers. Use Intel(R) MPI Library mpif90 compiler driver to automatically uses the appropriate MPI module.
- (SDK only) Perform the following steps to generate bindings for your compiler that is not directly supported by the Intel(R) MPI Library:
 1. Go to the binding directory

```
# cd <installdir>/binding
```
 2. Extract the binding kit

```
# tar -zxvf intel-mpi-binding-kit.tar.gz
```
 3. Follow instructions in the README-intel-mpi-binding-kit.txt
- (SDK only) To use the Intel(R) Debugger, set the IDB_HOME environment variable. It should point to the location of the Intel(R) Debugger.
- (SDK only) Use the following command to launch an Intel MPI application with Valgrind* 3.3.0:

```
# mpiexec -n <# of processes> <other_mpiexec_options> valgrind \
  --leak-check=full --undef-value-errors=yes \
  --log-file=<logfilename>.%p \
  --suppressions=<installdir>/etc/valgrind.supp <executable>
```

where:

<logfilename>.%p - log file name for each MPI process
<installdir> - the Intel MPI Library installation path
<executable> - name of the executable file

- Note that many routines in the libmpigi library (shipped with the Intel(R) MPI Library) are more highly optimized for Intel microprocessors than for non-Intel microprocessors.

Technical Support

Your feedback is very important to us. To receive technical support for the tools provided in this product and technical information including FAQ's and product updates, you need to register for an Intel(R) Premier Support account

at the Registration Center.

This package is supported by Intel(R) Premier Support. Direct customer support requests at:

<https://premier.intel.com>

General information on Intel(R) product-support offerings may be obtained at:

<http://www.intel.com/software/products/support>

The Intel(R) MPI Library home page can be found at:

<http://www.intel.com/go/mpi>

The Intel(R) MPI Library support web site,

<http://www.intel.com/software/products/support/mpi/> provides top technical issues, frequently asked questions, product documentation, and product errata.

Requests for licenses can be directed to the Registration Center at:

<http://www.intel.com/software/products/registrationcenter>

Before submitting a support issue, see the Intel(R) MPI Library for Linux* OS Getting Started Guide for details on post-install testing to ensure that basic facilities are working.

When submitting a support issue to Intel(R) Premier Support, please provide specific details of your problem, including:

- The Intel(R) MPI Library package name and version information
 - Host architecture (for example, IA-32 or Intel(R) 64 architecture)
 - Compiler(s) and versions
 - Operating system(s) and versions
 - Specifics on how to reproduce the problem. Include makefiles, command lines, small test cases, and build instructions.
- Use <installdir>/test sources as test cases, when possible.

You can obtain version information for the Intel(R) MPI Library package in the file mpisupport.txt.

Submitting Issues

- Go to <https://premier.intel.com>
- Log in to the site. Note that your username and password are case-sensitive.
- Click on the "Submit Issue" link in the left navigation bar.
- Choose "Development Environment (tools, SDV, EAP)" from the "Product Type" drop-down list. If this is a software or license-related issue, choose the "Intel(R) MPI Library, Linux*" from the "Product Name" drop-down list.
- Enter your question and complete the fields in the windows that follow to successfully submit the issue.

Note: Notify your support representative prior to submitting source code where access needs to be restricted to certain countries to determine if this request can be accommodated.

Copyright and Licenses

The Intel(R) MPI Library is based on MPICH2* from Argonne National Laboratory* (ANL) and MVAPICH2* from Ohio State University* (OSU).

See the information below for additional licenses of the following third party tools used within the Intel(R) MPI Library:
Eclipse*, Silicon Graphics Inc.* STL, libc, gdf, BOOST*, my_getopt, Python*, and AVL Trees*.

Eclipse*

<http://www.eclipse.org/legal/epl-v10.html>

Silicon Graphics, Inc.* Standard Template Library

* Copyright (c) 1996,1997
* Silicon Graphics Computer Systems, Inc.
*
* Permission to use, copy, modify, distribute and sell this software
* and its documentation for any purpose is hereby granted without fee,
* provided that the above copyright notice appear in all copies and
* that both that copyright notice and this permission notice appear
* in supporting documentation. Silicon Graphics makes no
* representations about the suitability of this software for any
* purpose. It is provided "as is" without express or implied warranty.
*
*/

libc

/*
* Copyright (c) 1988 Regents of the University of California.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the University nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT

* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

gdf

/**

* This is copy of the code which implements the GFD(32) hashing of datatypes
* described in this paper:

*

* Julien Langou, George Bosilca, Graham Fagg and Jack Dongarra (2005).

* Hash functions for MPI datatypes.

* In the Proceedings of the 12th European PVM/MPI Users' Group Meeting, Sorrento,
* Italy, September 2005.

* Springer's Lecture Notes in Computer Science, LCNS-3666:76-83, 2005.

* related software.

*

* <http://www.cs.utk.edu/~library/TechReports/2005/ut-cs-05-552.pdf>

* <http://www.cs.utk.edu/~langou/articles/LBFD:05/2005-LBFD.html>

*

* The code is used with permission of the author and was released under the

* "Modified BSD" license (no need to mention in advertising material). Here's

* a copy of the complete COPYING file that came with the source:

Copyright (c) 1992-2006 The University of Tennessee. All rights reserved.

\$COPYRIGHT\$

Additional copyrights may follow

\$HEADERS\$

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer listed
in this license in the documentation and/or other materials
provided with the distribution.
- Neither the name of the copyright holders nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BOOST*

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

my_getopt

my_getopt - a command-line argument parser

Copyright 1997-2001, Benjamin Sittler

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Python*

PSF LICENSE AGREEMENT FOR PYTHON 2.3

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 2.3 software in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly,

prepare derivative works, distribute, and otherwise use Python 2.3 alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, for example, "Copyright (c) 2001, 2002, 2003, 2004 Python Software Foundation; All Rights Reserved" are retained in Python 2.3 alone or in any derivative version prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 2.3 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 2.3.

4. PSF is making Python 2.3 available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 2.3 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 2.3 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 2.3, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python 2.3, Licensee agrees to be bound by the terms and conditions of this License Agreement.

AVL Trees*

Copyright (c) 1989-1997 by Brad Appleton, All rights reserved.

This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it freely, subject to the following restrictions:

1. Neither the authors of the software nor their employers (including any of the employers' subsidiaries and subdivisions) are responsible for maintaining & supporting this software or for any consequences resulting from the use of this software, no matter how awful, even if they arise from flaws in the software.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users

ever read sources, credits must appear in the documentation.

4. This notice may not be removed or altered.

The Intel MPI library includes altered AVL Trees* source codes.

Disclaimer and Legal Information

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature.htm>

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Go to:

http://www.intel.com/products/processor_number/

MPEG-1, MPEG-2, MPEG-4, H.261, H.263, H.264, MP3, DV, VC-1, MJPEG, AC3, AAC, G.711, G.722, G.722.1, G.722.2, AMRWB, Extended AMRWB (AMRWB+), G.167, G.168, G.169, G.723.1, G.726, G.728, G.729, G.729.1, GSM AMR, GSM FR are international standards promoted by ISO, IEC, ITU, ETSI, 3GPP and other organizations. Implementations of these standards, or the standard enabled platforms may require licenses from various entities, including Intel Corporation.

BlueMoon, BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Inside, Cilk, Core Inside, E-GOLD, i960, Intel, the Intel logo, Intel AppUp, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Intel Insider, the Intel Inside logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel Sponsors of Tomorrow., the Intel Sponsors of Tomorrow. logo,

Intel StrataFlash, Intel vPro, Intel XScale, InTru, the InTru logo, the InTru Inside logo, InTru soundmark, Itanium, Itanium Inside, MCS, MMX, Moblin, Pentium, Pentium Inside, Puma, skool, the skool logo, SMARTi, Sound Mark, The Creators Project, The Journey Inside, Thunderbolt, Ultrabook, vPro Inside, VTune, Xeon, Xeon Inside, X-GOLD, XMM, X-PMU and XPOSYS are trademarks of Intel Corporation in the U.S. and other countries.

* Other names and brands may be claimed as the property of others.

Microsoft, Windows, Visual Studio, Visual C++, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Java is a registered trademark of Oracle and/or its affiliates.

Copyright (C) [2003]-[2011], Intel Corporation. All rights reserved.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.