

-----  
Intel(R) MPI Library for Linux\*  
v2.0.1  
Release Notes  
-----

## CONTENTS

-----

- 0. WHAT'S NEW
- 1. OVERVIEW
- 2. PRODUCT CONTENTS
- 3. KEY FEATURES
- 4. SYSTEM REQUIREMENTS
  - 4.1 SUPPORTED HARDWARE
  - 4.2 SUPPORTED SOFTWARE
  - 4.3 SUPPORTED LANGUAGES
- 5. INSTALLATION NOTES
- 6. KNOWN LIMITATIONS
- 7. DOCUMENTATION
- 8. TECHNICAL SUPPORT
- 9. COPYRIGHT AND LEGAL INFORMATION

## 0. WHAT'S NEW

-----

Intel(R) MPI Library for Linux\* v2.0.1 is an update release of the Intel(R) MPI Library for Linux\* v2.0.

This release includes the following new features compared to Intel(R) MPI Library v2.0 (see product documentation for more details):

- Increased application performance
  - o Selection of the rdssm device as the default MPI device if the variable I\_MPI\_DEVICE is not defined.
  - o Control over the point-to-point protocol threshold through the I\_MPI\_EAGER\_THRESHOLD environment variable
  - o Optimized collective operations controlled by the I\_MPI\_FAST\_COLLECTIVES environment variable
  - o Extended process pinning for SGI Altix\* systems controlled by the I\_MPI\_PIN\_MODE and I\_MPI\_PIN\_PROCS environment variables
  - o Optimized memory registration controlled by the I\_MPI\_RDMA\_TRANSLATION\_CACHE environment variable
- Improved stability and correctness
  - o Control over the MPD\* daemon startup connection phase through the I\_MPI\_MPD\_CONNECTION\_TIMEOUT environment variable
  - o Co-existence of several MPD\* rings under one user id with the help

- o of the MPD\_CON\_EXT and I\_MPI\_MPD\_CONF environment variables
- o Synchronized fallback upon the sock device after detection of the DAPL\* provider issues at job startup
- o Control over the fallback upon the sock device through the I\_MPI\_FALLBACK\_DEVICE environment variable
- o Control over the DAPL\* connection establishment phase through the I\_MPI\_DAPL\_CONNECTION\_TIMEOUT environment variable
- o Several application relevant bug fixes
- Increased interoperability
  - o Optional non-modification of the /etc/ld.so.conf file during product installation through the --update-ldsoconf=no installer option
  - o Extraction installer capability for non-RPM\* based systems through the --extract installer option
  - o Selection of alternative TotalView\* executable files through the TOTALVIEW environment variable
- Enhanced operating system support
  - o SGI\* ProPack 4
  - o Red Flag\* DC Server, version 5.0
  - o SuSE\*, version 9.3 and 10.0
- Integration with the leading job schedulers
  - o LSF\*, version 6.1 and higher
  - o PBS Pro\*, version 7.1 and higher
  - o Torque\*, version 1.2.0 and higher
- Extended compiler support
  - o Intel(R) C++ Compiler for Linux\*, version 9.0 and higher
  - o Intel(R) Fortran Compiler for Linux\*, version 9.0 and higher
  - o GNU\* compilers, version 4.0 and higher
- Message queue browsing support in leading parallel debuggers
  - o DDT\*, version 1.9.2 and higher
  - o TotalView\*, version 7.1.0 and higher

---

## 1. OVERVIEW

---

Intel(R) MPI Library for Linux\* v2.0.1 is a multi-fabric message passing library based on ANL\* MPICH2\* and OSU\* MVAPICH2\*.

Intel(R) MPI Library for Linux\* v2.0.1 implements the Message Passing Interface, v2 (MPI-2) specification.

To receive technical support and updates, you need to register your Intel Software Product. See section 8, TECHNICAL SUPPORT.

---

## 2. PRODUCT CONTENTS

---

Intel(R) MPI Library Runtime Environment contains the tools you need to run programs including MPD daemons and supporting utilities, shared (.so) libraries, Release Notes, Getting Started guide, and Reference Manual.

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

---

### 3. KEY FEATURES

---

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

- MPI-1 and MPI-2 specification conformance (depending on the MPI device)
- Switchable, multi-fabric support for the following interconnection fabrics:
  - 1) Sockets, for example, TCP/IP over Ethernet\*, Gigabit Ethernet, and other interconnects
  - 2) Shared memory
  - 3) RDMA-capable network fabrics via DAPL\*, such as InfiniBand\* and Myrinet\*
  - 4) Concurrent use of 1) and 2) in clusters with SMP nodes
  - 5) Concurrent use of 1), 2), and 3) in clusters with SMP nodes connected with DAPL\*-based interconnects
- Support for IA-32 and Itanium(R) architecture clusters using Intel(R) C++ Compiler for Linux\* version 7.1, 8.0, 8.1, 9.0, and higher, Intel(R) Fortran Compiler for Linux\* version 7.1, 8.0, 8.1, 9.0, and higher, and GNU\* compilers
- Support for Intel(R) Extended Memory 64 Technology (Intel(R) EM64T) using Intel(R) C++ Compiler for Linux\* version 8.1, 9.0, and higher, Intel(R) Fortran Compiler for Linux\* version 8.1, 9.0, and higher, and GNU\* compilers
- C, C++, Fortran 77 and Fortran 90 language bindings
- Dynamic or static linking
- Clusters with homogeneous processor architectures only.

---

### 4. SYSTEM REQUIREMENTS

---

The following sections describe supported hardware and software.

---

#### 4.1 SUPPORTED HARDWARE

---

IA-32-based systems:

A system based on an Pentium(R) III processor 450-MHz or greater;  
Pentium(R) 4 or Intel(R) Xeon(R) processor recommended

1 GB of RAM recommended  
100 MB of free hard disk space

Itanium(R)-based systems:

Itanium(R) processor  
1 GB of RAM recommended  
100 MB of free hard disk space

Systems using Intel(R) Extended Memory 64 Technology (Intel(R) EM64T):

Intel(R) Xeon(R) processor  
1 GB of RAM recommended  
100 MB of free hard disk space

-----  
4.2 SUPPORTED SOFTWARE  
-----

Operating Systems:

IA-32-based systems:

Red Hat\* Linux\* version 8.0, or  
Red Hat Enterprise Linux\* 3.0, or  
Red Hat Enterprise Linux\* 4.0, or  
SuSE Linux\* version 9.0, or  
SuSE Linux\* version 9.1, or  
SuSE Linux\* Enterprise Server 9.

Itanium(R)-based systems:

Red Hat\* Advanced Server 2.1, or  
Red Hat Enterprise Linux\* 3.0, or  
Red Hat Enterprise Linux\* 4.0, or  
SuSE Linux\* version 9.0, or  
SuSE Linux\* version 9.1, or  
SuSE Linux\* Enterprise Server 9.

Systems using Intel(R) EM64T:

Red Hat Enterprise Linux\* 3.0 Update 3, or  
Red Hat Enterprise Linux\* 4.0, or  
SuSE Linux\* version 9.1, or  
SuSE Linux\* Enterprise Server 9.

Compilers:

GNU\*: C, C++, F77 compilers, version 2.96 or higher.  
or

IA-32-based systems:

Intel(R) C++ Compiler for Linux\* version 7.1, 8.0, 8.1, 9.0 and higher.  
Intel(R) Fortran Compiler for Linux\* version 7.1, 8.0, 8.1, 9.0 and higher.

Itanium(R)-based systems:

Intel(R) C++ Compiler for Linux\* version 7.1, 8.0, 8.1, 9.0 and higher.

Intel(R) Fortran Compiler for Linux\* version 7.1, 8.0, 8.1, 9.0 and higher.

Systems using Intel(R) EM64T:

Intel(R) C++ Compiler for Linux\* version 8.1, 9.0 and higher.

Intel(R) Fortran Compiler for Linux\* version 8.1, 9.0 and higher.

Additional Software:

- Python\* version 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\* version 2.2.3 etc. source distributions).
- An XML parser such as expat or pyxml.
- If using InfiniBand\*, Myrinet\*, or other RDMA-capable network fabrics, a DAPL\* v1.1 or DAPL\* v1.2 standard-compliant provider library/driver is required. DAPL\* providers are typically provided with your network fabric hardware and software.

---

### 4.3 SUPPORTED LANGUAGES

---

For GNU\* compilers: C, C++, F77

For Intel(R) compilers: C, C++, F77, F90

---

## 5. INSTALLATION NOTES

---

This section is intended for system administrators and describes the steps required to install Intel(R) MPI Library for Linux\*.

To install Intel(R) MPI Library do the following:

1. Obtain a license key. See Section 8, "SUPPORT INFORMATION", if you do not have a license key.
2. Install the l\_mpi\_p\_2.0.1.<package#>.tar.gz package by using the following commands:

```
# cp l_mpi_p_2.0.1.<package#>.tar.gz /tmp
# cd /tmp
# tar -xzf l_mpi_p_2.0.1.<package#>.tar.gz
# cd l_mpi_p_2.0.1.<package#>
# cp <license key>.lic .
    (alternatively, you can copy the license to /opt/intel/licenses)
# ./install
```

## NOTES:

- This installation does not overwrite any pre-existing Intel(R) MPI Library you may have installed.

After installing Intel(R) MPI Library v2.0.1 Development Kit you can continue to use the original Intel(R) MPI Library v1.0 Development Kit by referring to the original installation directory (typically, /opt/intel\_mpi\_10).

- The ./install script allows you to avoid modification of the /etc/ld.so.conf file during installation in root mode. Use the --update-ldsoconf option, for example:

```
# ./install --update-ldsoconf no
```

- You can install Intel(R) MPI Library v2.0.1 as an ordinary user. Run the ./install script and follow the instructions, or add the --nonroot --nonrpm options to the ./install invocation string above to select the non-root, non-RPM installation mode.
- You can install Intel(R) MPI Library v2.0.1 on a system that does not use the RPM\* package manager. Add the --nonrpm option to the ./install invocation string above to select the non-RPM installation mode.

Alternatively, you can extract Intel(R) MPI Library v2.0.1 RPM\* package and mpiEULA.txt file. Use the --extract option and indicate the desired extraction directory, for example:

```
# ./install --extract /tmp
```

- The ./install script enables you to select the install location on most operating systems. Select the exact same install location when installing Intel(R) MPI Library on each node of your cluster.
- Intel(R) MPI Library v2.0.1 Development Kit is installed by default into the directory /opt/intel/mpi/2.0.1 as compared to the directory /opt/intel\_mpi\_10 for Intel(R) MPI Library v1.0 Development Kit and the directory /opt/intel/mpi/2.0 for Intel(R) MPI Library v2.0 Development Kit.
- The ./install script has a silent installation mode that can help you to install Intel(R) MPI Library on clusters with many nodes. In order to use this mode, edit the file SilentInstallConfigFile.ini contained in the unpacked package directory. In particular, the line stating

EULA=reject

should be changed to read

EULA=accept

Other lines in the file enable you to define the INSTALLDIR, LICENSEPATH, INSTALLMODE, PROCEED\_WITHOUT\_PYTHON, and MODIFY\_LD\_SO\_CONF parameters.

After editing the file use the following command instead of the usual # ./install command:

```
# ./install --silent SilentInstallConfigFile.ini
```

3. Add the following PATH and/or LD\_LIBRARY\_PATH settings in your .cshrc or .bashrc files so that the settings are visible on all nodes in your cluster:
  - Ensure that Python\* is in your PATH
  - Source the appropriate mpivars.[c]sh script from the Intel(R) MPI Library bin (or, if applicable, bin64) directory
  - If using Intel(R) compilers, source any required \*vars.[c]sh scripts
  - Set up additional environment variables as needed. See the Intel(R) MPI Library Getting Started guide for more information.

---

## 6. KNOWN LIMITATIONS and TROUBLESHOOTING

---

The following are known issues and possible workarounds for this release:

- **IMPORTANT!** If the I\_MPI\_DEVICE variable is not defined, Intel(R) MPI Library v2.0.1 will try to load the MPI device library pointed to by the libmpi.def.so symbolic link before falling back upon the built-in MPI sock device. By default, the libmpi.def.so symbolic link points to the rdssm MPI device (libmpi.rdssm.so.2.1).
- **IMPORTANT!** Compiler drivers of Intel(R) MPI Library v2.0.1 embed the actual Development Kit library path (default /opt/intel/mpi/2.0.1) and default Runtime Environment library path (/opt/intel/mpi-rt/2.0) into the executables using the -rpath linker option.
- **IMPORTANT!** Intel(R) MPI Library v2.0.1 is not backward binary compatible with Intel(R) MPI Library v1.0. Recompile and relink your programs in order to use Intel(R) MPI Library v2.0.1.

- IMPORTANT! Compiler drivers of Intel(R) MPI Library v2.0.1 do not implicitly add the -fpic option to the generated compiler invocation string. Ensure that you use this option explicitly with the Intel(R) MPI Library v2.0.1 compiler drivers if you require position-independent code.
- IMPORTANT! Intel(R) MPI Library v2.0.1 enhances message-passing performance on DAPL\*-based interconnects by maintaining a cache of virtual-to-physical address translations in the MPI devices rdma and rdssm.

Set the environment variable LD\_DYNAMIC\_WEAK to "1" if your program dynamically loads the standard C library before dynamically loading Intel(R) MPI Library v2.0.1. Alternatively, use the environment variable LD\_PRELOAD to load Intel(R) MPI Library v2.0.1 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.

- Intel(R) MPI Library does not support heterogeneous clusters of mixed architectures and/or operating environments.
- Intel(R) MPI Library cannot load dynamic MPI devices if the application was linked statically. Use the --static\_mpi flag with the Intel(R) MPI Library compiler drivers to link the MPI library statically and allow the MPI device and system libraries to be linked dynamically at run time.
- Intel(R) MPI Library v2.0.1 requires Python\* version 2.2 or higher for process management. Certain older operating systems do not have this program installed by default, for example:
  - o On IA-32: Red Hat\* Linux\* 7.x or earlier
  - o On Itanium(R) architecture: Red Hat\* Advanced Server 2.1 or earlier
- Intel(R) MPI Library v2.0.1 requires the python-xml\* package or its equivalent on each node in the cluster for process management. For example, the following OS does not have this package installed by default:
  - o SuSE Linux\* Enterprise Server 9 or later
- Intel(R) MPI Library v2.0.1 requires the expat\* or pyxml\* package, or an equivalent XML parser on each node in the cluster for process management.
- The following MPI-2 features are not supported by Intel(R) MPI Library v2.0.1 since they are not supported in MPICH2 v1.0.1:

- o Process spawning and attachment for MPI devices other than sock device.
  - o Passive target one-sided communication when the target process does not call any MPI functions.
  - o Multithreading support of level MPI\_THREAD\_MULTIPLE.
  - o User-defined data representations and the external32 data representation.
- 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 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. Doing so 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.
  - If you decide to use the -host <hostname> option of the mpiexec command, make sure that the symbolic host names or IP addresses literally match those from the mpdboot mpd.hosts file.

- Presence of environment variables with non-printable characters in user environment settings may cause the process startup to fail. To work around this issue, 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.

- Intel(R) MPI Library v2.0.1 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 Intel(R) MPI Library v1.0.

- 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 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.

- In order to use GNU\* Fortran compiler version 4.0 and higher, set the environment variable MPICH\_F77 to "gfortran", or add the -fc=gfortran option to the invocation string of the Intel(R) MPI Library mpif77 compiler script.

- Intel(R) MPI Library v2.0.1 does not provide the MPI module for GNU\* Fortran 95 compiler version 4.0 and higher.

- Intel(R) MPI Library v2.0.1 does not provide support for exception handling for GNU\* C++ compiler version 3.4 and higher.

- Certain DAPL\* providers may not work with Intel(R) MPI Library for Linux\*, for example:

- o Mellanox\* IB Gold version 1.6.1 or later. Contact Mellanox\*, or

download alternative OpenIB\* DAPL\* provider at <http://sourceforge.net/projects/openib>, or use DAPL\* providers from Cisco\*, SilverStorm\*, or Voltaire\*.

o Myricom\* GM DAPL\* provider. Contact Myricom\* or download alternative GM DAPL\* provider at <http://sourceforge.net/projects/dapl-gm> .

- Certain DAPL\* providers may not function properly if your application uses 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 OpenIB\* DAPL\* provider with Linux\* kernel version earlier than 2.6.12

---

## 7. DOCUMENTATION

---

"Getting Started with Intel(R) MPI Library", found in Getting\_Started.pdf, contains information on the following subjects:

- Using Intel(R) MPI Library. Describes a basic usage model and walks you through the basic steps of setting up MPD daemons, compiling and linking, selecting a network fabric or device, and running an MPI program.
- Troubleshooting. Includes post-install testing steps and describes compiling and running a test program.

"Intel(R) MPI Library Reference Manual", found in Reference\_Manual.pdf, contains information on the following subjects:

- Command Reference. Describes compiler commands, options, and environment variables. Includes job startup commands, process placement, and MPD daemon commands.
- Tuning Reference. Describes environment variables that influence library behavior and performance. Covers process pinning, device control, collective operations, and other topics.

### Notation Conventions

---

Release Notes and other user documentation use the following notation conventions:

- <item> Indicates that the items enclosed in angle brackets are placeholders for elements such as installation paths.
- [items] Indicates that the items enclosed in brackets are optional
- { item | item } Indicates a set of choices from which you must select one
- ... (ellipses) Indicates that an argument can be repeated several times

---

## 8. TECHNICAL SUPPORT

---

This package is supported via Intel(R) Premier Support. Intel(R) Premier Support issues may be submitted at:  
<https://premier.intel.com>

General information on Intel(R) product-support offerings may be obtained at:  
<http://www.intel.com/software/products/support>

Requests for licenses can be directed to the Registration Center at:  
<http://www.intel.com/software/products/registrationcenter>

Before submitting a support issue, see "Getting Started with Intel(R) MPI Library" 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:

- Intel(R) MPI Library package name and version information
- Host architecture (for example, IA-32 or Itanium(R) architecture,)
- Compiler(s) and versions
- Operating system(s) and versions
- Specifics on how to reproduce problems. Include makefiles, command lines, small test cases, and build instructions.  
Use <installdir>/test sources as test cases, when possible.
- Try to reproduce your problem with the simplest device possible. For example, try the sock device first, then ssm, and then rdma or rdssm.

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

---

## 9. COPYRIGHT AND LEGAL INFORMATION

---

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

---

Copyright (C) Intel Corporation 2003-2006. All Rights Reserved.

This Intel(R) MPI Library software ("Software") is furnished under license and may only be used or copied in accordance with the terms of that license. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. The Software is subject to change without notice, and should not be construed as a commitment by Intel Corporation to market, license, sell or support any product or technology. Unless otherwise provided for in the license under which this Software is provided, the Software is provided AS IS, with no warranties of any kind, express or implied. Except as expressly permitted by the Software license, neither Intel Corporation nor

its suppliers assumes any responsibility or liability for any errors or inaccuracies that may appear herein.

Except as expressly permitted by the Software license, no part of the Software may be reproduced, stored in a retrieval system, transmitted in any form, or distributed by any means without the express written consent of Intel Corporation.