



Intel® MPI Library for Windows* OS

Getting Started Guide

The Intel® MPI Library is a multi-fabric message passing library that implements the Message Passing Interface, v2 (MPI-2) specification. Use it to switch interconnection fabrics without re-linking.

This *Getting Started Guide* explains how to use the Intel® MPI Library to compile and run a simple MPI program. This guide also includes basic usage examples and troubleshooting tips.

To quickly start using the Intel® MPI Library, print this short guide and walk through the example provided.

Copyright © 2003–2010 Intel Corporation

All Rights Reserved

Document Number: 316404-007

Revision: 4.0

World Wide Web: <http://www.intel.com>

Contents

1	About this Document	4
1.1	Intended Audience.....	4
1.2	Using Doc Type Field.....	4
1.3	Conventions and Symbols.....	4
1.4	Related Information.....	5
2	Using the Intel® MPI Library.....	6
2.1	Usage Model.....	6
2.2	Before you Begin.....	6
2.3	Quick Start.....	7
2.4	Compiling and Linking.....	8
2.5	Setting up SMPD Services.....	8
2.6	Selecting a Network Fabric.....	9
2.7	Running an MPI Program.....	10
3	Troubleshooting.....	12
3.1	Testing Installation.....	12
3.2	Compiling and Running a Test Program.....	12
4	Next Steps.....	14

Disclaimer and Legal Notices

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 by visiting [Intel's Web Site](#).

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See http://www.intel.com/products/processor_number for details.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Atom, Centrino Atom Inside, Centrino Inside, Centrino logo, Core Inside, FlashFile, i960, InstantIP, Intel, Intel logo, Intel386, Intel486, IntelDX2, IntelDX4, IntelSX2, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skool, Sound Mark, The Journey Inside, Viiv Inside, vPro Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2007-2010, Intel Corporation. All rights reserved.

1 About this Document

The *Intel® MPI Library for Windows* OS Getting Started Guide* contains information on the following subjects:

- First steps using the Intel® MPI Library
- Troubleshooting outlines first-aid troubleshooting actions

1.1 Intended Audience

This *Getting Started Guide* is intended for first time users.

1.2 Using Doc Type Field

This *Getting Started Guide* contains the following sections:

Document Organization

Section	Description
Section 1 About this Document	Section 1 introduces this document
Section 2 Using the Intel® MPI Library	Section 2 describes how to use the Intel® MPI Library
Section 3 Troubleshooting	Section 3 outlines first-aid troubleshooting actions
Section 4 Next Steps	Section 4 provides links further resources

1.3 Conventions and Symbols

The following conventions are used in this document.

Table 1.3-1 Conventions and Symbols used in this Document

<i>This type style</i>	Document or product names
This type style	Hyperlinks

<i>This type style</i>	Document or product names
This type style	Commands, arguments, options, file names
THIS_TYPE_STYLE	Environment variables
< <i>this type style</i> >	Placeholders for actual values
[items]	Optional items
{ item item }	Selectable items separated by vertical bar(s)
(SDK only)	For Software Development Kit (SDK) users only

1.4 Related Information

To get more information about the Intel® MPI Library, see the following resources:

[Product Web Site](#)

[Intel® MPI Library Support](#)

[Intel® Cluster Tools Products](#)

[Intel® Software Development Products](#)

2 Using the Intel® MPI Library

2.1 Usage Model

Using the Intel® MPI Library involves the following steps. These steps are described in the corresponding sections in detail.

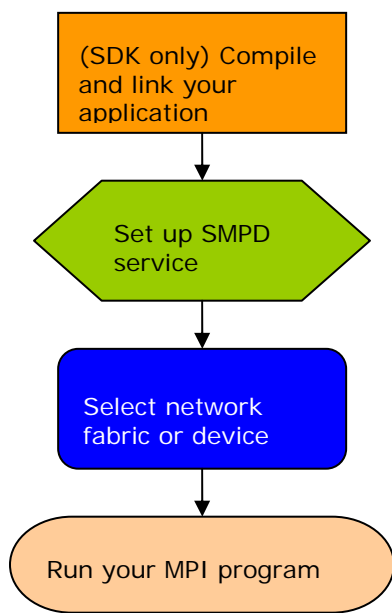


Figure 1: Flowchart representing the usage model for working with the Intel® MPI Library.

2.2 Before You Begin

1. Before using the Intel® MPI Library, ensure that the library, scripts, and utility applications are installed. See the *Intel® MPI Library for Windows* OS Installation Guide* for installation instructions.
2. For getting proper environment settings, use the following commands from the Start menu:
Start > Programs > Intel(R) Software Development Tools > Intel(R) MPI Library 4.0 > Build Environment for the IA-32 architecture

Start > Programs > Intel Software Development Tools > Intel(R) MPI Library 4.0 > Build Environment for the Intel® 64 architecture

Alternatively, you can open a new console (cmd) window and run one of the following BAT files from the command line.

```
<installdir>\ia32\bin\mpivars.bat
```

```
<installdir>\em64t\bin\mpivars.bat
```

3. You should have administrator privileges on all nodes of the cluster to start the `smpd` service on all nodes of the cluster.

2.3 Quick Start

1. Use the `call` batch command for getting proper environment settings from the `mpivars.bat` batch scripts included with the Intel® MPI Library. It is located in the `<installdir>\em64t\bin` directory for the Intel® 64 architecture, or in the `<installdir>\ia32\bin` directory for the 32-bit mode.
2. Make sure the `smpd` services are installed and started on compute nodes. Otherwise install them manually from the command line by using the `-install smpd` option. If the `smpd` service stops, start it through Computer Management -> Services and Applications -> Services or from the command line manually using the `-start smpd` option.
3. **(SDK only)** Make sure that you have a compiler in your `PATH`.
4. **(SDK only)** Compile the test program using the appropriate compiler driver. For example:

```
> mpicc.bat -o test <installdir>\test\test.c
```
5. Register your credentials using the `wmpiregister` GUI utility.
6. Execute the test using the GUI utility `wmpiexec`. Set the application name and a number of processes. In this case all processes start on the current host. To start a test on a remote host or on more than one host press the **Advanced Options** button and fill the appropriate fields. Use the **Show Command** button to check the command line. Press the **Execute** button to start the program.

You can use the command line interface instead of the GUI interface.

1. Use the `wmpiexec -register` option instead of the `wmpiregister` GUI utility to register your credentials.
2. Use the CLI `wmpiexec` command to execute the test.

```
> wmpiexec.exe -n <# of processes> test.exe
```

or

```
> wmpiexec.exe -hosts <# of hosts> <host1_name> \  
<host1 # of processes> <host2_name> \  
<host2 # of processes> ... test.exe
```

See the rest of this document and the *Intel® MPI Library Reference Manual for Windows*OS* for more details.

2.4 Compiling and Linking

(SDK only)

To compile and link an MPI program with the Intel® MPI Library do the following steps:

1. Create a Winxx Console project for Microsoft* Visual Studio* 2005.
2. Choose the x64 solution platform.
3. Add `<installdir>\em64t\include` to the include path.
4. Add `<installdir>\em64t\lib` to the library path.
5. Add `impi.lib` (Release) or `impid.lib` (Debug) to your target link command for C applications.
6. Add `impi.lib` and `impicxx.lib` (Release) or `impid.lib` and `impicxxd.lib` (Debug) to your target link command for C++ applications. Link application with `impimt.lib` (Release) `impidmt.lib` (Debug) for multithreading.
7. Build a program.
8. Place your application and all the dynamic libraries in a shared location or copy them to all the nodes.
9. Run the application using the `mpiexec.exe` command.

2.5 Setting up SMPD Services

The Intel® MPI Library uses a Simple Multi-Purpose Daemon (SMPD) job startup mechanism. In order to run programs compiled with Microsoft* Visual Studio* (or related), set up a SMPD service.

NOTE: You should have administrator privileges to start the `smpd` service and all users can launch processes with `mpiexec`.

To set up SMPD services:

1. During the Intel® MPI Library installation the `smpd` service is started. During installation you can cancel the `smpd` service startup.
2. You can start, restart, stop or remove the `smpd` service manually when the Intel® MPI Library is installed. Find `smpd.exe` in the `<installdir>\em64t\bin`
3. Use the following command on each node of the cluster: `> smpd.exe -remove` to remove the previous `smpd` service.
4. Use the following command on each node of the cluster: `> smpd.exe -install` to install the `smpd` service manually.

2.6 Selecting a Network Fabric

The Intel® MPI Library dynamically selects different fabrics for communication between MPI processes. To select a specific fabric combination, set the new `I_MPI_FABRICS` or the old `I_MPI_DEVICE` environment variable.

I_MPI_FABRICS

(I_MPI_DEVICE)

Select the particular network fabrics to be used.

Syntax

```
I_MPI_FABRICS=<fabric>/<intra-node fabric>:<inter-nodes fabric>
```

Where `<fabric> := {shm, dapl, tcp}`

`<intra-node fabric> := {shm, dapl, tcp}`

`<inter-nodes fabric> := {dapl, tcp}`

Deprecated Syntax

```
I_MPI_DEVICE=<device>[:<provider>]
```

Arguments

<code><fabric></code>	Define a network fabric
<code>shm</code>	Shared-memory
<code>dapl</code>	DAPL-capable network fabrics, such as InfiniBand*, iWarp*, Dolphin*, and XPMEM* (through DAPL*)
<code>tcp</code>	TCP/IP-capable network fabrics, such as Ethernet and InfiniBand* (through IPoIB*)

Correspondence with I_MPI_DEVICE

<code><device></code>	<code><fabric></code>
<code>sock</code>	<code>tcp</code>
<code>shm</code>	<code>shm</code>
<code>ssm</code>	<code>shm:tcp</code>
<code>rdma</code>	<code>dapl</code>

<code>rdssm</code>	<code>shm:dapl</code>
<code><provider></code>	Optional DAPL* provider name (only for the <code>rdma</code> and the <code>rdssm</code> devices) <code>I_MPI_DAPL_PROVIDER=<provider></code>

Use the `<provider>` specification only for the `{rdma, rdssm}` devices.

For example, to select the OFED* InfiniBand* device, use the following command:

```
> mpiexec -n <# of processes> \
    -env I_MPI_DEVICE rdssm:ibnic0v2-scm <executable>
```

For these devices, if `<provider>` is not specified, the first DAPL* provider in the `dat.conf` file is used.

NOTE: Ensure the selected fabric is available. For example, use `shm` only if all the processes can communicate with each other through shared memory. Use `rdma` only if all the processes can communicate with each other through a single DAPL provider. Ensure that the `dat.dll` library is in your `%PATH%`. Otherwise, use the `-genv` option for `mpiexec.exe` for setting the `I_MPI_DAT_LIBRARY` environment variable with the fully-qualified path to the `dat.dll` library.

2.7 Running an MPI Program

Use the `mpiexec` command to launch programs linked with the Intel® MPI Library:

```
> mpiexec.exe -n <# of processes> myprog.exe
```

NOTE: The `wmpiexec` utility is a GUI wrapper for `mpiexec.exe`. See the *Intel® MPI Library Reference Manual* for more details.

Use the only required `mpiexec -n` option to set the number of processes on the local node.

Use the `-hosts` option to set names of hosts and number of processes:

```
> mpiexec.exe -hosts 2 host1 2 host2 2 myprog.exe
```

If you are using a network fabric as opposed to the default fabric, use the `-genv` option to set the `I_MPI_DEVICE` variable.

For example, to run an MPI program using the `shm` fabric, type in the following command:

```
> mpiexec.exe -genv I_MPI_DEVICE shm -n <# of processes> \  
myprog.exe
```

You may use the `-configfile` option to run the program:

```
> mpiexec.exe -configfile config_file
```

The configuration file contains:

```
-host host1 -n 1 -genv I_MPI_DEVICE rdssm myprog.exe  
-host host2 -n 1 -genv I_MPI_DEVICE rdssm myprog.exe
```

For the `rdma` capable fabric, use the following command:

```
> mpiexec.exe -hosts 2 host1 1 host2 1 -genv I_MPI_DEVICE rdma myprog.exe
```

You can select any supported device. For more information, see Section [Selecting a Network Fabric](#).

If you successfully run your application using the Intel® MPI Library, you can move your application from one cluster to another and use different fabrics between the nodes without re-linking. If you encounter problems, see [Troubleshooting](#) for possible solutions.

3 Troubleshooting

Use the following sections to troubleshoot problems with installation, setup, and running applications using the Intel® MPI Library.

3.1 Testing Installation

To ensure that the Intel® MPI Library is installed and functioning, complete a general testing, compile and run a test program.

To test the installation:

1. Verify through the Computer Management that the `smpd` service is started. It calls the Intel MPI Process Manager.
2. Verify that `<installdir>\ia32\bin` (`<installdir>\em64t\bin` for the Intel® 64 architecture in the 64-bit mode) is in your path:

```
> echo %PATH%
```

You should see the correct path for each node you test.

3. **(SDK only)** If you use Intel compilers, verify that the appropriate directories are included in the `PATH` and `LIB` environment variables:

```
> mpiexec.exe -hosts 2 host1 1 host2 1 a.bat
```

where `a.bat` contains

```
echo %PATH%
```

You should see the correct directories for these path variables for each node you test. If not, call the appropriate `*vars.bat` scripts. For example, with Intel® C++ Compiler 11.0 for Windows*OS for the Intel® 64 architecture in the 64-bit mode, use the Windows program menu to select:

Intel(R) Software Development Tools > Intel(R) C++ Compiler 11.0 >Build Environment for the Intel® 64 architecture or from the command line

```
%ProgramFiles%\Intel\Compiler\C++\11.0\em64t\bin\iclvars.bat
```

3.2 Compiling and Running a Test Program

The install directory `<installdir>\test` contains test programs which you can use for testing. To compile one of them or your test program, do the following:

1. **(SDK only)** Compile a test program as described in Section 2.4 Compiling and Linking.
2. If you are using InfiniBand* or other RDMA-capable network hardware and software, verify that everything is functioning.
3. Run the test program with all available configurations on your cluster.

- Test the `sock` device using:

```
> mpiexec.exe -n 2 -env I_MPI_DEBUG 2 -env I_MPI_DEVICE sock a.out
```

You should see one line of output for each rank, as well as debug output indicating that the `sock` device is used.
- Test the `ssm` devices using:

```
> mpiexec.exe -n 2 -env I_MPI_DEBUG 2 -env I_MPI_DEVICE ssm a.out
```

You should see one line of output for each rank, as well as debug output indicating that the `ssm` device is used.
- Test any other fabric devices using:

```
> mpiexec.exe -n 2 -env I_MPI_DEBUG 2 -env I_MPI_DEVICE <device> a.out
```

where `<device>` can be `shm`, `rdma`, or `rdssm`

For each of the `mpiexec` commands used, you should see one line of output for each rank, as well as debug output indicating which device was used. The device(s) should agree with the `I_MPI_DEVICE` setting.

4 Next Steps

To get more information about the Intel® MPI Library, explore the following resources:

The *Intel® MPI Library Release Notes* include key product details. See the *Intel® MPI Library Release Notes* for updated information on requirements, technical support, and known limitations. Use the Windows program menu to select Intel(R) Software Development Tools > Intel(R) MPI Library > Intel(R) MPI Library for Windows* OS Release Notes.

For more information see Websites:

[Product Web Site](#)

[Intel® MPI Library Support](#)

[Intel® Cluster Tools Products](#)

[Intel® Software Development Products](#)