Intel® Cluster Toolkit 3.2.1
Release Notes
(Revision 20090612)

Contents

Overview

Intel® Cluster Toolkit 3.2.1 for Linux* OS and Microsoft* Windows* Compute Cluster Server OS (Microsoft Windows CCS*) accelerate parallel software development on homogenous cluster systems based on IA-32, or IA-64 (Itanium®), or Intel® 64 architectures. In terms of the Intel® Cluster Toolkit software for Windows*, consider references within this document to Microsoft Windows CCS OS, and Microsoft* Windows* HPC Server 2008 OS **as interchangeable**. The Microsoft Windows CCS and Microsoft Windows HPC Server 2008 operating systems only support Intel® 64 architecture.

Intel Cluster Toolkit 3.2.1 supports application development using Intel® MPI Library with optimized parallel libraries, performance analysis, and benchmarks. Intel Cluster Toolkit 3.2.1 saves software developers time and improves execution performance on distributed computing systems.

Intel Cluster Toolkit 3.2.1 for Linux and Microsoft Windows CCS supports crucial parts of the message-passing interface (MPI) application development process including:

- Intel MPI Library 3.2 Update 1 which is approaching conformity with the Message Passing Interface 2 Standard (MPI-2), enables multiple interconnect solutions with a single implementation. Intel seeks to be a software leader in MPI and open standards.

- The Intel® Trace Analyzer and Collector 7.2 Update 1 (ITAC)
  - Intel® Trace Collector 7.2 Update 1 (ITC) provides event-based tracing in cluster applications through an instrumentation library that causes low-overhead in execution. The trace information provides performance data, statistics, multi-threaded events, and automatic instrumentation of user binaries on IA-32 architectures.
  - The Intel® Trace Analyzer 7.2 Update 1 (ITA) provides visual analysis of application activities gathered by the Intel Trace Collector. This software component has been completely rewritten.
  - A message checking component of the Intel Trace Collector provides a novel MPI correctness technology which detects errors with data types, buffers, communicators, point-to-point messages and collective operations, deadlocks, and data corruption.
- Application tuning with optimized mathematical library functions from Intel® Math Kernel Library 10.2 (Intel® MKL 10.2) that includes ScaLAPACK solvers and Cluster DFTs (Discrete Fourier Transforms).
- Intel® MPI Benchmarks 3.2 make it easy to gather performance information about a cluster system.

Note that when the system administrator or user completes the Intel Cluster Toolkit 3.2.1 installation process, there will be a file called `Doc_Index.htm` in the `doc` folder on the master node of the cluster. This file can be used as a documentation map to navigate to various information resources pertaining to the Intel Cluster Toolkit. Additional information about the exact location of `Doc_Index.htm`, and its content are further described in the **Installation Notes** section of this release notes document.

Back to Top

Product Contents

Table for product components, including documentation from these components.

Tool	Version	Manual	Revision
Intel® MPI Library	3.2 Update 1	Doc_Index.html for Linux* Doc_Index.htm for Microsoft* Windows* CCS*	3.2 Update 1
Intel® Trace Analyzer and Collector	7.2 Update 1	ITA_Reference_Guide.pdf ITC_Reference_Guide.pdf	7.2 Update 1
Intel® Math Kernel Library	10.2	mkl_documentation.htm	10.2
Intel® MPI Benchmarks	3.2	IMB_Users_Guide.pdf	3.2

Back to Top

New Features

Intel® Cluster Toolkit 3.2.1

- With one installation session, the Intel Cluster Tools installer on Linux OS and/or Windows* OS will install:
  - Intel® MPI Benchmarks 3.2
  - Intel® MKL 10.2
  - Intel® MPI Library 3.2 Update 1
  - Intel® Trace Analyzer and Collector 7.2 Update 1
- SUSE Linux Enterprise Server* 11 is now supported
- Please see the section of this document titled, “Installation on Linux* OS” for some possible installation issues and workarounds involving Linux OS on Itanium® architecture

Intel® MPI Library 3.2 Update 1 for Linux* OS

This release includes the following updates from the Intel® MPI Library 3.2 for Linux* OS (see product documentation for more details):

- Linux* Standard Base (LSB) compliant RPMs
- ILP64 support
- Process layout information exchange
- Process pinning improvement
- Collective optimization
- Scalable mpdboot startup
- Extended interoperability
  - Intel® Compiler 11.1 support

Intel® MPI Library 3.2 Update 1 for Windows* OS

The Intel® MPI Library 3.2 Update 1 for Windows* OS includes the following updates compared to the Intel® MPI Library 3.2 for Windows* (see product documentation for more details):

- ILP64 support
- Process layout information exchange
- Process pinning improvement
- Collective optimization
- Active Directory based user authentication
- Direct interprocess memory copy
- Scalable mpdboot startup
- Extended interoperability
  - Intel® Compiler 11.1 support

Intel® Math Kernel Library (MKL) 10.2

New in Intel® MKL 10.2 for Linux* OS

- Performance Improvements in the BLAS:
  - 32-bit improvements
    - 40-50% improvement for (Z,C)GEMM on Quad-Core Intel® Xeon® processor 5300 series
    - 10% improvement for all GEMM code on Quad-Core Intel® Xeon® processor 5400 series
  - 64-bit improvements
    - 2.5-3% improvement for DGEMM on 1 thread on Quad-Core Intel® Xeon® processor 5400 series
    - 50% improvement for SGEMM on the Intel® Core™ i7 processor family

    - 3% improvement for CGEMM on 1 thread on the f Intel® Core™ i7 processor family
    - 2-3% improvement for ZGEMM on 1 thread on the Intel® Core™ i7 processor family
    - 30% improvement for right-side cases of DTRSM on the Intel® Core™ i7 processor family
- Improvements to the direct sparse solver (DSS/PARDISO):
    - The performance of out-of-core PARDISO was improved by 35% on average.
    - Support of separate backward/forward substitution for DSS/PARDISO has been added.
    - A new parameter for turning off iterative refinement for DSS interface has been introduced.
    - A new parameter for checking sparse matrix structure has been introduced for PARDISO interface.
- The capability to track the progress of a lengthy computation and/or interrupt the computation has been added via a callback function mechanism. A function called `mkl_progress` can be defined in a user application, which will be called regularly from a subset of the MKL LAPACK routines. See the LAPACK Auxiliary and Utility Routines chapter in the reference manual for more information. Refer to the specific function descriptions to see which LAPACK functions support the feature.
- Transposition functions have been added to Intel MKL. See the reference manual for further detail.
- The C++ std::complex type can now be used instead of MKL-specific complex types.
- An implementation of the Boost uBLAS matrix-matrix multiplication routine is now provided which will make use of the highly optimized version of DGEMM in the Intel MKL BLAS. See the User guide for more information.
- Improvements to the sparse BLAS:
    - Support for all data types (single precision, complex and double complex) has been added.
    - Routines for computing the sum and product of two sparse matrices stored, both stored in the compressed sparse row format have been added.
- The Vector Math Library functions, CdfNorm, CdfNormInv, and ErfcInv, have been optimized to achieve much improved performance.
- Performance improvement on the Intel® Core™ i7 processor family:
    - 3-17% improvement for the following VML functions: Asin, Asinh, Acos, Acosh, Atan, Atan2, Atanh, Cbrt, CIS, Cos, Cosh, Conj, Div, ErfInv, Exp, Hypot, Inv, InvCbrt, InvSqrt, Ln, Log10, MulByConj, Sin, SinCos, Sinh, Sqrt, Tanh.
    - 7-67% improvement for uniform random number generation.
    - 3-10% improvement for VSL distribution generators based on Wichmann-Hill, Sobol, and Niederreiter BRNGs (64-bit only).
- The configuration file functionality has been removed. See the user guide for alternative means to configure the behavior of Intel MKL.
- When functions in Intel MKL are called from an MPI program they will be run on 1 thread by default (i.e., in the absence of explicit controls).
- The following VML functions: CdfNorm, CdfNormInv, and ErfcInv.
- The DftiCopyDescriptor function.
- The LP64 interface of DSS/PARDISO now uses 64-bit addressing for internal arrays on 64-bit operating systems. This allows the direct solver to solve larger systems.

- The default OpenMP runtime library for Intel MKL has been changed from libguide to libiomp. See the User Guide in the doc directory for more information.
- The optimized code paths for the Intel® Pentium® III processor have been removed from Intel MKL along with the associated processor specific dynamic link libraries. We continue to support the use of Intel MKL on this processor, but the default code path will be used and as a result performance may be reduced.
- The interval linear solver functions have been removed from MKL.
- Support for Intel MPI 1.x has ended.
- Documentation updates:
  - Eclipse IDE Infopop support for VML functions and VSL service functions. The infopop support means brief info on a function in a pop-up window appearing when the cursor is placed to the function/routine name in the Eclipse Editor panel. This Eclipse feature is implemented in the CDT 5.0 version.
  - The FFTW Wrappers for MKL Notes have been removed from the product package after their content was integrated into the Intel MKL Reference Manual (Appendix G).
  - New functions have been documented in the reference manual, and support for Boost uBLAS matrix-matrix multiplication has been described in the User Guide.
  - The parallel BLAS (PBLAS) which support ScaLAPACK are now documented in the Intel MKL reference manual.
  - Added FORTRAN 77 support info to the description of VML and VSL functions in the Intel MKL reference manual.

New in Intel® MKL 10.2 for Windows* OS

- Performance Improvements in the BLAS:
  - 32-bit improvements
    - 40-50% improvement for (Z,C)GEMM on Quad-Core Intel® Xeon® processor 5300 series
    - 10% improvement for all GEMM code on Quad-Core Intel® Xeon® processor 5400 series
  - 64-bit improvements
    - 2.5-3% improvement for DGEMM on 1 thread on Quad-Core Intel® Xeon® processor 5400 series
    - 50% improvement for SGEMM on the Intel® Core™ i7 processor family
    - 3% improvement for CGEMM on 1 thread on the Intel® Core™ i7 processor family
    - 2-3% improvement for ZGEMM on 1 thread on the Intel® Core™ i7 processor family
    - 30% improvement for right-side cases of DTRSM on the Intel® Core™ i7 processor family
- Improvements to the direct sparse solver (DSS/PARDISO):
  - The performance of out-of-core PARDISO was improved by 35% on average.
  - Support of separate backward/forward substitution for DSS/PARDISO has been added.
  - A new parameter for turning off iterative refinement for DSS interface has been introduced.
  - A new parameter for checking sparse matrix structure has been introduced for PARDISO interface.
- The capability to track the progress of a lengthy computation and/or interrupt the computation has been added via a callback function mechanism. A function called `mkl_progress` can be defined in a user application, which will be called regularly from a subset of the MKL LAPACK routines. See the LAPACK Auxiliary

and Utility Routines chapter in the reference manual for more information. Refer to the specific function descriptions to see which LAPACK functions support the feature.

- Transposition functions have been added to Intel MKL. See the reference manual for further detail.
- The C++ std::complex type can now be used instead of MKL-specific complex types.
- An implementation of the Boost uBLAS matrix-matrix multiplication routine is now provided which will make use of the highly optimized version of DGEMM in the Intel MKL BLAS. See the User guide for more information.
- Improvements to the sparse BLAS:
  - Support for all data types (single precision, complex and double complex) has been added.
  - Routines for computing the sum and product of two sparse matrices stored, both stored in the compressed sparse row format have been added.
- The Vector Math Library functions, CdfNorm, CdfNormInv, and ErfcInv, have been optimized to achieve much improved performance.
- Performance improvement on the Intel® Core™ i7 processor family:
  - 3-17% improvement for the following VML functions: Asin, Asinh, Acos, Acosh, Atan, Atan2, Atanh, Cbrt, CIS, Cos, Cosh, Conj, Div, ErfInv, Exp, Hypot, Inv, InvCbrt, InvSqrt, Ln, Log10, MulByConj, Sin, SinCos, Sinh, Sqrt, Tanh.
  - 7-67% improvement for uniform random number generation.
  - 3-10% improvement for VSL distribution generators based on Wichmann-Hill, Sobol, and Niederreiter BRNGs (64-bit only).
- The configuration file functionality has been removed. See the user guide for alternative means to configure the behavior of Intel MKL.
- All hurdles to creation of DLLs from the static libraries have been removed.
- When functions in Intel MKL are called from an MPI program they will be run on 1 thread by default (i.e., in the absence of explicit controls).
- The following VML functions have been added: CdfNorm, CdfNormInv, and ErfcInv.
- The DftiCopyDescriptor function has been added.
- The LP64 interface of DSS/PARDISO now uses 64-bit addressing for internal arrays on 64-bit operating systems. This allows the direct solver to solve larger systems.
- The default OpenMP runtime library for Intel MKL has been changed from libguide to libiomp. See the User Guide in the doc directory for more information.
- Documentation updates:
  - The parallel BLAS (PBLAS) which support ScaLAPACK are now documented in the Intel MKL reference manual.
  - Added instructions for using example programs in Microsoft* Visual Studio to the User Guide.
  - Intel MKL Documentation is now accessible from the Microsoft* Visual Studio 'Help' menu with F1 Help and Dynamic Help features provided in the code editor. For more information, see the Intel MKL User's Guide.
- It is no longer possible to set environment variables during the installation process. Three script files, mklvars32.bat, mklvarsem64t.bat, and mklvars64.bat are available in the tools\environment directory to set the PATH, LIB, and INCLUDE environment variables at the command prompt.
- The optimized code paths for the Intel® Pentium® III processor have been removed from Intel MKL along with the associated processor specific dynamic link

libraries. We continue to support the use of Intel MKL on this processor, but the default code path will be used and as a result performance may be reduced.
- The interval linear solver functions have been removed from Intel MKL.
- Documentation updates:
  - The FFTW Wrappers for Intel MKL Notes have been removed from the product package after their content was integrated into the Intel MKL Reference Manual (Appendix G).
- New functions have been documented in the reference manual, and support for Boost uBLAS matrix-matrix multiplication has been described in the User Guide.

Intel® Trace Analyzer and Collector 7.2 Update 1

New in Intel® Trace Analyzer and Collector 7.2 Update 1

- Linux* Standard Base (LSB) compliant RPMs
- Intel® Compiler 11.1 support
- Migration to the Trolltech* Qt* 4.x

Back to Top

Resolved Issues

Bug Number	Title of Bug Report
716	ScaLAPACK Test Suite Improvement Request for Microsoft Windows
877	[DOC] Links to www3.intel.com server is found in Release notes
881	Include updated EULA
883	[DOC] Invalid ITAC version in release notes for compiler edition
884	[INS] Invalid detail install directory for ICT CE
885	Ini file in CD package has wrong permission
886	Incomplete path for ICTCE windows default install
887	Windows packages contain old EULA
888	[INS] MKL install failure on second node with "--nonrpm" key
889	[DOC] www3 links in documentation

Back to Top

System Requirements

Processor System Requirements

Intel® Pentium® 4 processor, or
Intel® Xeon® processor, or
Intel® Itanium® 2 processor, or
Intel® Core™2 Duo processor (example of Intel® 64 architecture)

Note that it is assumed that the processors listed above are configured into homogeneous clusters.

Back to Top

Disk-Space Requirements

20 GBs of disk space (minimum)

Note that during the installation process the installer may need up to 4 gigabytes of temporary disk storage to manage the intermediate installation files.

Back to Top

Operating System Requirements

OS Distributions	IA-32 Architecture	Intel® 64 Architecture		IA-64 Architecture
		32-Bit Applications	64-Bit Applications	
SGI* Propack* 5 for Linux*		S	S	S
Red Hat Enterprise Linux* 4.0	S	S	S	S
Red Hat Enterprise Linux* 5.0	S	S	S	S
SUSE Linux Enterprise Server* 9	S	S	S	S
SUSE Linux Enterprise Server* 10	S	S	S	S
SUSE Linux Enterprise Server* 11	S	S	S	S
Microsoft* Windows* Compute Cluster Server (Microsoft Windows CCS*)	N/A	S	S	N/A
Microsoft* Windows* HPC Server 2008	N/A	S	S	N/A

S = Supported

Back to Top

Memory Requirements

2 GB of RAM (minimum)

Back to Top

Intel® Compilers

For all of the Intel processor architectures the version number on the Intel compilers should be 8.1 or greater.

Back to Top

Installation Notes

Setting Up Secure Shell on Linux OS

Installation of the Intel Cluster Toolkit 3.2.1 assumes that the homogenous cluster has `ssh` connectivity.

1. Make sure `ssh` is installed on your computing cluster. To do this type the shell command:

   ```
   which ssh
   ```

   If it does not exist, you can acquire ssh from the URL:

   http://www.openssh.org

2. Create an authentication key as follows:

   ```
   ssh-keygen -t rsa
   ```

   This will generate a private and public key pair. The private key should be saved in:

   ```
   ~/.ssh/identity
   ```

   and the public key should be saved in:

   ```
   ~/.ssh/identity.pub
   ```

3. Authorize access by placing the contents of the public key into the `~/.ssh/authorized keys` file. All keys listed in that file are allowed access. One way to do this is to issue the shell command:

   ```
   cat ~/.ssh/identity.pub >> ~/.ssh/authorized_keys
   ```

   If the computing system that you are connecting to does not share a common file system, then `~/.ssh/identity.pub` should be concatenated to the `~/.ssh/authorized keys` file of the computing system you will be connecting to. Secure Shell will insist that the file `authorized_keys` has its permissions set so that it is not group writable, so do the following:

```
chmod go-rwx ~/.ssh/authorized keys
```

This step avoids the need to enter your password each time you want to run a secure shell command.

4. In order to avoid typing in your pass phrase each time `ssh` is invoked, an `ssh-agent` needs to be created and your pass phrase added. This is done as follows:

```
ssh-agent $SHELL

ssh-add
```

Back to Top

Checklist in Case There Are Problems on Linux OS

1. Make sure that the host names listed in the `machines.LINUX` file are also listed in the `/etc/ssh_known_hosts` file on your network or your `~/.ssh/known_hosts` file in your home directory.
2. It is important that `/tmp` has permissions set to `377`, with root as owner and group 0.
3. `openssh` has a `-v` flag option which is very useful for tracking down handshaking problems.
4. If you encounter the following message:

```
"scp: FATAL: Executing ssh1 in compatibility mode failed
(Check that scp1 is in your PATH). Lost connection."
```

Then there is an incompatibility with the secure shell protocol between the master node and at least one of the other nodes listed in the `machines.LINUX` file. The solution is to install compatible versions of secure shell on all nodes of the cluster.

Back to Top

Alternative Solution to Establishing Secure Shell Connectivity on Linux OS

Within the “tar” package of the Intel Cluster Toolkit, there is an `expect` shell script file called `sshconnectivity.exp`. “`expect`” is a tool for automating interactive applications. To run “`sshconnectivity.exp`”, the `expect` runtime software needs to be installed on your Linux system. To make sure that the `expect` runtime software is properly installed, type:

```
which expect
```

If you encounter a "`Command not found.`" error message, you can download the expect software package from the following URL:

http://expect.nist.gov/

The `expect` shell script file called “`sshconnectivity.exp`” can be used to help you establish secure shell connectivity on a cluster system. The syntax for the command is:

```
./sshconnectivity.exp machines.LINUX
```

This `expect` shell script will create or update a `~/.ssh` directory on each node of the cluster beginning with the master node which must be the first name listed in the `machines.LINUX` file. This script will prompt you for your password twice.

```
Enter your user password:
Re-enter your user password:
```

To provide security each time you enter your user password, asterisks will appear in lieu of the password text. Upon successful completion of the script, the following message fragment will appear:

```
                                    ...
Node count = 4
Secure shell connectivity was established on all nodes.
                                    ...
```

A log of the transactions for this script will be recorded in:

/tmp/sshconnectivity.<*login-name*>.log

where <*login-name*> is a meta-symbol for your actual login.

Note that the `expect` shell script `sshconnectivity.exp` will remove the write access capability on the group and other "permission categories" for the user's home directory folder. If this is not done, a password prompt will continue to be issued for any secure shell activity.

This process is demonstrated by the following complete graph[1] (Figure 1) illustration where a vertex in the graph represents a cluster computing node, and an edge between two vertices connotes that the two cluster computing nodes have exchanged public keys for secure shell connectivity. Secure shell connectivity is intended to provide secure, encrypted communication channels between two or more cluster nodes over an insecure network.

The script `sshconnectivity.exp` will call the appropriate secure shell utilities to generate a private key and a public key for each node of the cluster.

For the complete graph example in Figure 1, suppose there are nodes (vertices) 1 to n in the cluster. For a given node i, nodes 1 to $i - 1$ and nodes $i + 1$ to n are provided with the public key from node i. The user's public keys for a given node will be stored in the `~/.ssh` folder associated with the user's home directory for that computing node. Since there are $n - 1$ edges to a given node, that node will have $n - 1$ public keys in the `~/.ssh` folder that were provided by the other $n - 1$ nodes in the cluster. The example in Figure 1 represents a computing cluster that has at total of 5 nodes. The edges connecting a node indicate that that node has received 4 public keys from the remaining computing nodes. Also looking out from a given node indicates that the given node has provided its own public key to the remaining nodes that are reachable via the 4 edge paths.

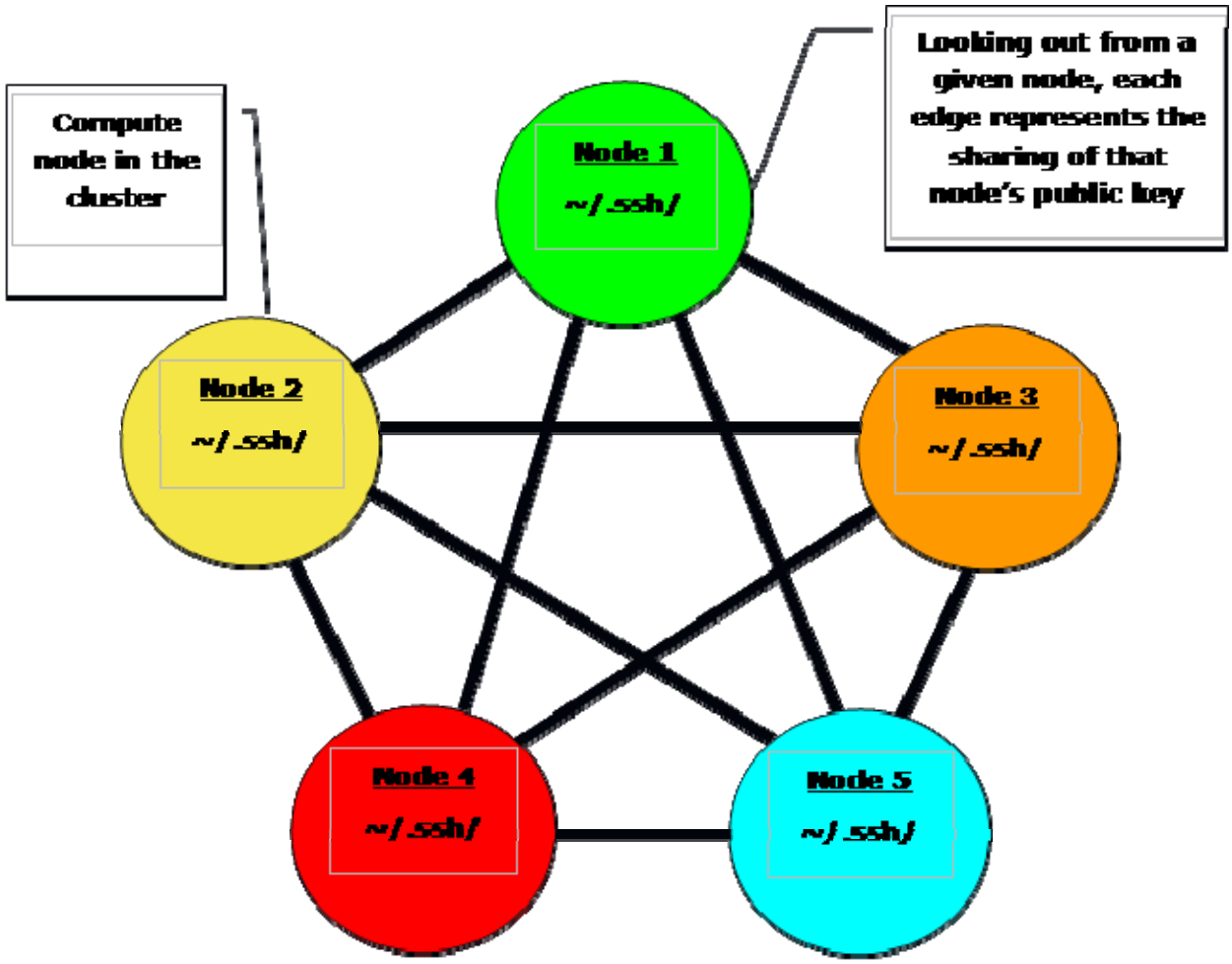

Figure 1 – Illustration of Secure Shell Connectivity for a Computing Cluster

Note that the expect shell script `sshconnectivity.exp` will remove the write access capability on the group and other "permission categories" for the user's home directory folder. If this is not done, a password prompt will continue to be issued for any secure shell activity.

Back to Top

The Intel Cluster Toolkit installation process on Linux is comprised of eight basic steps. The installation process on Microsoft Windows CCS alternatively uses the Microsoft installer model. The Intel Cluster Toolkit 3.2.1 package for Linux and Microsoft Windows CCS consist of the following components:

Software Component	Default Installation Directory on Linux	Default Installation Directory on Windows
Intel Math Kernel Library 10.2	/opt/intel/ict/3.2.1.0xx/mkl where "xxx" is a sequence of three digits denoting the build number.	c:\program files (x86)\intel\ict\3.2.1.0xx\mkl where "xxx" is a sequence of three digits denoting the build number.
Intel MPI Library 3.2 Update 1	/opt/intel/ict/3.2.1.0xx/impi where "xxx" is a sequence of three digits denoting the build number.	c:\program files (x86)\intel\ict\3.2.1.0xx\mpi where "xxx" is a sequence of three digits denoting the build number.
Intel MPI Benchmarks 3.2	/opt/intel/ict/3.2.1.0xx/imb where "xxx" is a sequence of three digits denoting the build number.	c:\program files (x86)\intel\ict\3.2.1.0xx\imb where "xxx" is a sequence of three digits denoting the build number.
Intel Trace Analyzer and Collector 7.2 Update 1	/opt/intel/ict/3.2.1.0xx/itac where "xxx" is a sequence of three digits denoting the build number.	c:\program files (x86)\intel\ict\3.2.1.0xx\itac where "xxx" is a sequence of three digits denoting the build number.

Note that the Intel Cluster Toolkit installer will automatically make the appropriate selection of binaries, scripts, and text files from its installation archive based on the Intel processor architecture of the host system where the installation process is initiated. You do not have to worry about selecting the correct software component names for the given Intel® architecture. Also note that the values 035 in 3.2.1.035 and 037 in 3.2.1.037 are *example build number values* and may vary with your actual installation.

Recall that you as a user of the Intel Cluster Toolkit on Linux, Microsoft Windows CCS, or both may need assistance from your system administrator in installing the associated software packages on your cluster system, if the installation directory requires system administrative write privileges (e.g. `/opt/intel` on Linux). This assumes that your login account does not have administrative capabilities.

Back to Top

Installation on Linux* OS

Here are some important installation notes that should be read before beginning an installation on Linux OS: The 4.2.2 version of RPM on Red Hat Enterprise Linux 3.0 OS for Itanium 2 architecture has a broken relocation feature. This will be a serious problem for users trying to install on clusters where there are shared devices. A recommended solution is for the user to upgrade to the latest release of RPM. A *possible* URL for retrieving a recent release of RPM that resolves this problem on the Itanium 2 architecture is:

http://www.redhat.com

Secondly, on Itanium architecture for some OSes (for example, Red Hat Enterprise Linux 4.0 Update 7 OS, and Red Hat Enterprise Linux 5.0 Update 2 OS), installation of the Intel® Math Kernel Library non-native components "…/lib/32" and

"…/lib/em64t" are missing when the install is done into a non-default directory (I.e., a directory path other than /opt/intel). This is a problem with the RPM utility. Two known workarounds for this issue on Itanium architecture are:

A. Install the Intel® Cluster Tools at the default location (/opt/intel)
B. Use the non-rpm mode for installation (i.e., use the installer command-line option --nonrpm)

The above problem does not seem to appear on SLES 10 OS and SLES 11 OS for Itanium architecture.

To begin installation on Linux:

1. A machines.LINUX file will either need to be created, or an existing machines.LINUX file can be used by the Intel Cluster Toolkit installer to deploy the appropriate software packages from the toolkit amongst the nodes of the cluster. This machines.LINUX file contains a list of the computing nodes (i.e. the hostnames) for the cluster. The format is one hostname per line:

   hostname

   The hostname should be the same as the result from the Linux* command "hostname". An example of the content for the file machines.LINUX, where a contrived cluster consists of eight nodes might be:

   ```
   clusternode1
   clusternode2
   clusternode3
   clusternode4
   clusternode5
   clusternode6
   clusternode7
   clusternode8
   ```

   A line of text above is consider a comment line if column 1 contains the "#" symbol. It is always assumed that the first node in the list is the master node. The remaining nodes are the compute nodes. The text clusternode1 and clusternode2, for example, represent the names of two of the nodes in a contrived computing cluster. The contents of the machines.LINUX file can also be used by users to construct an mpd.hosts file for the multi-purpose daemon (MPD) protocol. The MPD protocol is used for running MPI applications that utilize Intel MPI Library.

2. In preparation for doing the installation, the user may want to create a staging area. On the system where the Intel Cluster Toolkit software components are to be installed, it is recommended that a staging area be constructed in a directory such as /tmp. An example folder path staging area might be:

   /tmp/ict_staging_area

   where ict_staging_area is an acronym for Intel Cluster Toolkit staging area.

3. Upon registering for Intel Cluster Toolkit 3.2.1, you will receive a serial number (e.g., C111-12345678) for this product. Your serial number can be found within

the email receipt of your product purchase. Go to the Intel Registration Center site and provide the product serial number information. Once admission has been granted into the registration center, a user will be able to access the Intel® Premier web pages for software support.

4. The license for the Intel Cluster Toolkit license file that is provided to the user should be placed in a location pointed to by the `INTEL_LICENSE_FILE` environment variable. Do not change the file name as the "`.lic`" extension is critical. Common locations for the attached license file are:

   `<installation path>/licenses`

   where `licenses` is a sub-directory. For example, on the cluster system where the Intel Cluster Toolkit software is to be installed, all licenses for Intel-based software products might be placed in:

   `/opt/intel/licenses`

   If the license file is placed into a different file path from the above common location, it is also imperative that the user and/or the system administrator set the environment variable `INTEL_LICENSE_FILE` to the directory path where the Intel software licenses will reside *prior* to doing an installation of the Intel Cluster Toolkit. For Bourne* Shell or Korn* Shell the syntax for setting the `INTEL_LICENSE_FILE` environment variable might be:

   ```
   export INTEL_LICENSE_FILE=/opt/intel/licenses
   ```

   For C Shell, the syntax might be:

   ```
   setenv INTEL_LICENSE_FILE /opt/intel/licenses
   ```

5. Patrons can place the Intel Cluster Toolkit software package into the staging area folder.

6. The installer package for the Intel Cluster Toolkit has the following general nomenclature:

   `l_ict_<major>.<minor>.<update>.<package_num>.tar.gz`

   where `<major>.<minor>.<update>.<package_num>` is a string such as:

   `b_3.2.1.xxx`, where `b` is an acronym for beta

   or

   `p_3.2.1.xxx`, where `p` is an acronym for production

   The `package_num` meta-symbol is a string such as `035`. This string indicates the package number.

   The command:

   `tar -xvzf l_ict_<major>.<minor>.<update>.<package_num>.tar.gz`

will create a subdirectory called `l_ict_<major>.<minor>.<update>.<package_num>`. Change to that directory with the shell command:

```
cd l_ict_<major>.<minor>.<update>.<package_num>
```

For example, suppose the installation package is called `l_ict_p_3.2.1.035.tar.gz`. In the staging area that has been created, type the command:

```
tar -xvzf l_ict_p_3.2.1.035.tar.gz
```

This will create a subdirectory called `l_ict_p_3.2.1.035`. Change to that directory with the shell command:

```
cd l_ict_p_3.2.1.035
```

7. As mentioned in the release notes section titled, "**Alternative Solution to Establishing Secure Shell Connectivity**", the `expect` shell script file called "`sshconnectivity.exp`" can be used to help the user establish secure shell connectivity on a cluster system. "`sshconnectivity.exp`" is also located within the `l_ict_<version>.<release>` directory staging area. The syntax for the "`sshconnectivity.exp`" command is:

```
./sshconnectivity.exp machines.LINUX
```

This `expect` shell script will create or update a `~/.ssh` directory on each node of the cluster beginning with the master node which must be the first name listed in the `machines.LINUX` file. This script will prompt the user for a cluster password twice.

```
Enter your cluster password:
Re-enter your cluster password:
```

Each time the user enters the cluster password, asterisks will appear in lieu of the password text so as to provide security. Upon successful completion of the script, the following message fragment will appear:

```
                                    ...
Node count = 4
Secure shell connectivity was established on all nodes.
                                    ...
```

A log of the transactions for this script will be recorded in:

```
/tmp/sshconnectivity.<login-name>.log
```

where `<login-name>` is a meta-symbol for the user's actual login.

8. Once secure shell connectivity is established, type a variation of the `install` command as demonstrated by the table below and follow the prompts issued by this install script.

Installation command	Is root	Installer	Default installation

	password required initially?	prompts to be aware of	area
./install.sh	Yes		/opt/intel/ict/...
./install.sh --nonroot	No	We recommend that you install the software using RPM (option 1). This will require root password. If you do not have root password, you can do a local installation in your home folder by choosing option 2 below. Which of the following would you like to do ? 1. Install the software using RPM (root password required) - Recommended. 2. Install the software without using RPM database (root password not required). x. Exit Please make a selection :	./intel/ict/... in your home directory
./install.sh --nonrpm	Yes	If you do not have the root password, you can do a local installation in your home folder by choosing option 2 below. Which of the following	./intel/ict/... in your home directory, if option 2 is selected

		would you like to do? 1. Install the software using RPM (root password required) - Recommended. 2. Install the software without using RPM database (root password not required). x. Exit. Please make a selection: (1/2/x) [2]	
./install.sh --nonroot --nonrpm	No		./intel/ict/... in your home directory

Note that the Intel MPI Benchmarks are only installed on the master node.

By default, the global root directory for the installation of the Intel Cluster Toolkit is:

/opt/intel/ict/*<major>*.*<minor>*.*<update>*.*<package_num>*

where *<major>*, *<minor>*, *<update>*, and *<package_num>* are integers. An example would be 3.2.1.035.

Within the folder path /opt/intel/ict/*<major>*.*<minor>*.*<update>*.*<package_num>* you will find the text files:

ictvars.csh

ictvars.sh

and

ictsupport.txt

If you are using Bourne Shell or Korn Shell for the login session, you should type:

. ./ictvars.sh

and for a login session that uses C Shell, you should type:

source ./ictvars.csh

The file called:

ictsupport.txt

contains the Package ID and Package Contents information. Please use the information in `ictsupport.txt` when submitting customer support requests.

For the default installation path, an index file, an FAQ file, and the user's guide are located in the directory path:

/opt/intel/ict/*<major>*.*<minor>*.*<update>*.*<package_num>*/doc

where as mentioned above, *<major>*, *<minor>*, *<update>*, and *<package_num>* are integers. A complete default folder path to the documentation directory might be:

/opt/intel/ict/3.2.1.035/doc

The name of the index file is:

Doc_Index.htm

The index file can be used to navigate to the FAQ, the release notes, the user's guide, and an internet accessible Intel Cluster Toolkit Tutorial. The index file will also provide links to Intel Trace Analyzer and Collector documentation, Intel MPI Library documentation, Intel MKL documentation, and Intel MPI Benchmarks documentation.

The name of the FAQ file is:

HelpMe_FAQ.htm

The name of the user's guide file is:

Getting_Started.htm

By default, the local version of the release notes is located in the directory path:

/opt/intel/ict/*<major>*, *<minor>*, *<update>*, and *<package_num>*/release_notes

The name of the release notes file is:

Release Notes.htm

With the default installation settings, the Intel® Cluster Toolkit support file (`ictsupport.txt`) is installed at:

/opt/intel/ict/3.2.1.035/ictsupport.txt

Back to Top

Windows Installation

The installer package for the Intel Cluster Toolkit has the following general nomenclature:

w_ict_<*major*>, <*minor*>, <*update*>, and <*package_num*>.exe

where<*major*>, <*minor*>, <*update*>, and <*package_num*> is a string such as:

b_3.2.1.xxx, where b is an acronym for beta

or

p_3.2.1.xxx, where p is an acronym for production

The <*package_num*> meta-symbol is a string such as 037. This string indicates the package number.

To install Intel® Cluster Toolkit simply go to the staging area where "w_ict_p_3.2.1.xxx.exe" is located, where "xxx" is a sequence of three digits (i.e., the <*package_num*>), and do the following:

1. Double click on the ".exe" file
2. Follow the instructions that are provided by the installation wizard. The installation wizard will install the components of the Intel® Cluster Toolkit on all nodes of the cluster.

Back to Top

Documentation

Below is a list of documents related to the Intel® Cluster Toolkit.

Document Name	File Name	Type of Information in the Document	Comment
Intel® Cluster Toolkit Release Notes	Release_Notes.htm	Information about this release and how to use the library with various compilers. Includes: • Overview • System Requirements • Setting up Secure Shell on Linux • Installation	For details about the Intel® Cluster Toolkit, please see the *Intel® Cluster Toolkit Getting Started Guide 3.2.1*.

		• FAQ – Frequently Asked Questions • Technical Support and Feedback • Copyright and Legal Information	
Intel® Cluster Toolkit Getting Started Guide	`Getting_Started.htm`	Contains detailed information about: • Intel® MKL 10.2 • Intel® MPI Library 3.2 Update 1 • Intel® MPI Benchmarks 3.2 • Intel® Trace Analyzer and Collector 7.2 Update 1	For implementation-specific information about the toolkit, please see the *Intel® Cluster Toolkit Release Notes.*
Intel® Cluster Toolkit FAQ	`HelpMe_FAQ.htm`	Contains a list of frequently asked questions (FAQ) regarding the Intel® Cluster Toolkit 3.2.1.	This file may contain helpful hints when using the Intel® Cluster Toolkit 3.2.1.
Intel® Cluster Toolkit Tutorial	Note that this link is not relevant to an Intel® Cluster Tools Beta program that is in preparation for a major release.	Contains detailed information about: • Intel® MKL 10.2 • Intel® MPI Library 3.2 Update 1 • Intel® MPI Benchmarks 3.2 • Intel® Trace Analyzer and Collector	This is a web-based version of the Intel® Cluster Toolkit Getting Started Guide. It may contain information that is more recent than that of the local version of the Intel® Cluster Toolkit Getting Started Guide that has been installed

		7.2 Update 1	on the user's computing cluster. The tutorial and the user's guide will have a revision number located in the title page such as "(Revision 20090529)". The larger the number, the more current the document revision.

Back to Top

Known Limitations

1. On Itanium 2-based systems, the following symptom may occur for RPM version 4.2.2 when doing an install with the `--nonroot` option:

   ```
   install.sh –nonroot
   ```

   the following symptom appears:

   ```
   Installing Intel(R) MPI Library for Linux* version p 3.2.1.038...
   Extracting
   files...    ##################################################   [100%
   ]

   Would you like to:

       Install:

       i.  Intel(R) MPI Library, Development Kit for Linux* version
   3.2-038

       x.  Exit

   Your choice? (i/x) [i]: i

   Cannot install Intel(R) MPI Library, Development Kit for Linux*
   version 3.2-038 to "/home/tooluser/intel/impi/3.2.1":
   RPM version 4.2.2 prevent installation into a non-default directory.
   Installation failed.
   WARNING: Intel(R) MPI Library for Linux* installer did not create
   "/tmp/install.XX2svw1O/MPI.ini" file; the product cannot be
   installed to all cluster nodes.
   Press Enter to continue...
   ```

   The solution is to use an updated version of RPM.

Back to Top

2. There have been situations where during the installation process, `/tmp` has filled up. We recommend that you have at least one gigabyte of free space in `/tmp` when doing an install of the Intel® Cluster Toolkit. Also, the installer script `install.sh` has the command-line options:

```
--temp-path=PATH
```

or

```
--temppath=PATH
```

which can direct the use of intermediate storage to another disk partition.

Back to Top

3. If during installation you see a diagnostic which looks something like:

```
...

Cannot install Intel(R) Trace Analyzer And Collector version 7.2.1p-
029 into directory "/usr/home/user01/intel/itac/7.2.1":
The prefix directory path "/usr/home/user01/intel/itac/7.2.1" is
owned by cookie intel-ta ipf version 7.2.1p-029.
Installation failed.
...
```

try the following corrective action when the installer completes:

1) Run an `uninstall.sh` command that should look something like the following:

```
/usr/home/user01/intel/ict/3.2.1.035/uninstall.sh
```

2) Reissue the install.sh command as done previously
3) If you see a prompt of the following type:

```
Would you like to:

    Install:

    1.  Intel(R) Trace Collector version 7.2.1p-029
    2.  Intel(R) Trace Analyzer version 7.2.1p-029
    a.  All of the above.

    Uninstall:

    3.  intel-ta ipf version 7.2.1p-029
        (non-rpm in "/usr/home/user01/intel/itac/7.2.1")
    4.  intel-tac ipf version 7.2.1p-029
        (non-rpm in "/usr/home/user01/intel/itac/7.2.1")

    x.  Exit

Please type a selection (1/2/a/3/4/x) [a]:
```

Enter the digit 3 under the “Uninstall” menu selection. This will probably result in a message that looks something like:

```
"/usr/home/user01/intel/ict/itac/7.2.1" does not exist. It
    looks like installation was damaged. Force uninstallation.
Press Enter to continue...
```

4) For the above, repeat this process again for any item that appears in the "Uninstall" menu category prompt. You will eventually encounter messages that look something like:

```
Installing...
Installation successful.
To uninstall this package, run
    "/usr/home/user01/intel/itac/7.2.1/uninstall.sh".
```

5) Continue this process of doing the uninstall selection first, any time you encounter such subsequent prompts (e.g., Intel® MPI Library):

```
Would you like to:

    Install:

    i.  Intel(R) MPI Library, Development Kit for Linux*
version 3.2p-038

    Uninstall:

    1.  intel-mpi-ipf version 3.2p-038
        (non-rpm in
"/usr/home/user01/intel/ict/3.2.1.035/impi/")

    x.  Exit

Please type a selection (i/1/x) [i]:
```

...

6) Eventually you should see the following messages from the Intel® Cluster Toolkit installer:

...

```
Installation successful.
To uninstall this package, run
"/usr/home/user01/intel/ict/3.2.1.035/impi/uninstall".

Completed cluster installation successfully.
```

This "`Installation failed`" symptom occurs when a prior installation of the Intel® Cluster Toolkit into a directory such as `/usr/home/user01/intel/ict/3.2.1.035` was removed without having issued the `uninstall.sh` command that for this example, resides in the directory `/usr/home/user01/intel/ict/3.2.1.035`.

Back to Top

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 Premier Support account at the Registration Center.

NOTE: Registering for support varies for release product or pre-release products (alpha, beta, etc) - only released products have support web pages on http://support.intel.com/.

To register for an account, please visit the Intel® Registration Center web site at http://www.intel.com/software/products/registrationcenter/index.htm. If you have forgotten your password, please email a request to: quadsupport@mailbox.intel.com. Please do not email your technical issue to this email address.

The product support web site, http://software.intel.com/en-us/articles/intel-cluster-toolkit-support-resources/, provides top technical issues, FAQs & Known Issues, Documentation and Training, and product errata.

Back to Top

Submitting Issues

To submit an issue via the Intel Premier Support website, please perform the following steps:

1. Ensure that Java* and JavaScript* are enabled in your browser.
2. Go to https://premier.intel.com/.
3. Type in your Login and Password. Both are case-sensitive.
4. Click the "Submit Issues" button in the left margin.
5. Read the Confidentiality Statement and click the "I Accept" button.
6. Click on the "Go" button next to the "Product" drop-down list.
7. Click on the "Submit Issue" link in the left navigation bar.
8. Choose "Development Environment (tools, SDV, EAP)" from the "Product Type" drop-down list.
9. If this is a software or license-related issue choose "Intel(R) Cluster Toolkit" from the "Product Name" drop-down list.
10. Enter your question and complete the fields in the web-page windows that follow to successfully submit the issue.

Please follow these guidelines when forming your problem report or product suggestion:

1. Describe your difficulty or suggestion. For problem reports, please be as specific as possible (e.g., including compiler and link command line options), so that we may reproduce the problem. Please include a small test case if possible.
2. Describe your system configuration information. Be sure to include specific information that may be applicable to your setup: operating system, name and version number of installed applications, and anything else that may be relevant to helping us address your concern.

Back to Top

Disclaimer and Legal Information

Developers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Improper use of reserved or undefined features or instructions may cause unpredictable behavior or failure in developer's software code when running on an Intel processor. Intel reserves these features or instructions for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from their unauthorized use. 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, skoool, 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.

Back to Top

---

[1] A mathematical definition of a complete graph in graph theory is a simple graph where an edge connects every pair of vertices. The complete graph on n vertices has n vertices and $n(n - 1)/2$ edges, and is denoted by K_n. Each vertex in the graph has degree $n - 1$. All complete graphs are their own cliques (a maximal complete graph). A graph of this type is maximally connected because the only vertex cut which disconnects the graph is the complete set of vertices.