# Intel® Math Kernel Library 10.2 for Linux* Release Notes

Release Notes

Document number: 321363-001US

## Contents:



## 1 Introduction

Intel® Math Kernel Library (Intel® MKL) is a library of highly optimized, extensively threaded math routines for science, engineering, and financial applications that require maximum performance. Core math functions include BLAS, LAPACK, ScaLAPACK, Sparse Solvers, Fast Fourier Transforms, Vector Math, and more. Offering performance optimizations for the latest Intel microarchitecture, it includes integration with Eclipse*.

This document provides system requirements, installation instructions, issues and limitations, and legal information.

To learn more about this product's:

- New features, see the "New in Intel® MKL 10.2" section below.
- Documentation, help, and samples, see the Intel® Math Kernel Library Documentation item in the Start menu program folder.
- Technical support, including answers to questions not addressed in the installed product, visit the technical support forum at: http://www.intel.com/software/products/support/mkl.

Please remember to register your product at https://registrationcenter.intel.com/ by providing your email address. This helps Intel recognize you as a valued customer in the support forum.

## New in Intel® MKL 10.2

- New features
    - LAPACK 3.2
        - 238 new LAPACK functions
        - Extra Precise Iterative Refinement
        - Non-Negative Diagonals from Householder QR factorization
        - High Performance QR and Householder Reflections on Low-Profile Matrices
        - New fast and accurate Jacobi SVD
        - Routines for Rectangular Full Packed format
        - Pivoted Cholesky
        - Mixed precision iterative refinement (Cholesky)
        - More robust DQDS algorithm
    - Introduced implementation of the DZGEMM Extended BLAS function (as described at http://www.netlib.org/blas/blast-forum/). See the description of the ?gemm family of functions in the BLAS section of the reference manual.
    - PARDISO now supports real and complex, single precision data
- Usability/Interface improvements
    - Sparse matrix format conversion routines which convert between the following formats:
        - CSR (3-array variation) ↔ CSC (3-array variation)
        - CSR (3-array variation) ↔ diagonal format
        - CSR (3-array variation) ↔ skyline
    - Fortran95 BLAS and LAPACK mod files are now included

        - Modules are pre-built with the Intel compiler and located in the include directory (see Intel® MKL User's Guide for full path)
        - Source is still included for use with other compilers
        - Documentation for these interfaces can be found in the Intel® MKL User's Guide
    - The FFTW3 interface is now integrated directly into the main libraries
        - Source code is still included to create wrappers for use with compilers not compatible with the default Intel® Fortran compiler convention for name decoration
        - See Appendix G of the Reference Manual for information
    - DFTI_DESCRIPTOR_HANDLE now represents a true type name and can now be referenced as a type in user programs
    - Added parameter to Jacobi matrix calculation routine in the optimization solver domain to allow access to user data (see the description of the djacobix function in the reference manual for more information)
    - Added an interface mapping calls to single precision BLAS functions in Intel® MKL (functions with 's' or 'c' initial letter) to 64-bit floating point precision functions has been added on 64-bit architectures (See 'sp2dp' in the Intel® MKL User Guide for more information)
    - Compatibility libraries (also known as "dummy" libraries) have been removed from this version of the library
- Performance improvements
    - Further threading in BLAS level 1 and 2 functions for Intel® 64 architecture
        - Level 1 functions (vector-vector): (CS,ZD,S,D)ROT, (C,Z,S,D)COPY, and (C,Z,S,D)SWAP
            - Increase in performance by up to 1.7-4.7 times over version 10.1 Update 1 on 4-core Intel® Core™ i7 processor depending on data location in cache
            - Increase in performance by up to 14-130 times over version 10.1 Update 1 on 24-core Intel® Xeon® processor 7400 series system, depending on data location in cache

        - Level 2 functions (matrix-vector): (C,Z,S,D)TRMV, (S,D)SYMV, (S,D)SYR, and (S,D)SYR2
            - Increase in performance by up to 1.9-2.9 times over version 10.1 Update 1 on 4-core Intel® Core™ i7 processor, depending on data location in cache
            - Increase in performance by up to 16-40 times over version 10.1 Update 1 on 24-core Intel® Xeon® processor 7400 series system, depending on data location in cache
    - Introduced recursive algorithm in 32-bit sequential version of DSYRK for up to 20% performance improvement on Intel® Core™ i7 processors and Intel® Xeon® processors in 5300, 5400, and 7400 series.
    - Improved LU factorization (DGETRF) by 25% over Intel MKL 10.1 Update 1 for large sizes on the Intel® Xeon® 7460 Processor; small sizes are also dramatically improved
    - BLAS *TBMV/*TBSV functions now use level 1 BLAS functions to improve performance by up to 3% on Intel® Core™ i7 processors and up to 10% on Intel® Core™2 processor 5300 and 5400 series.
    - Improved threading algorithms to increase DGEMM performance
        - up to 7% improvement on 8 threads and up to 50% on 3,5,7 threads on the Intel® Core™ i7 processor
        - up to 50% improvement on 3 threads on Intel® Xeon® processor 7400 series.
    - Threaded 1D complex-to-complex FFTs for non-prime sizes
    - New algorithms for 3D complex-to-complex transforms deliver better performance for small sizes (up to 64x64x64) on 1 or 2 threads
    - Implemented high-level parallelization of out-of-core (OOC) PARDISO when operating on symmetric positive definite matrices.
    - Reduced memory use by PARDISO for both in-core and out-of-core on all matrix types
        - PARDISO OOC now uses less than half the memory previously used in Intel MKL 10.1 for real symmetric, complex Hermitian, or complex symmetric matrices
    - Parallelized Reordering and Symbolic factorization stage in PARDISO/DSS

- Up to 2 times better performance (30% improvement on average) on Intel® Core® i7 and Intel® Core™2 processors for the following VML functions: v(s,d)Round, v(s,d)Inv, v(s,d)Div, v(s,d)Sqrt, v(s,d)Exp, v(s,d)Ln, v(s,d)Atan, v(s,d)Atan2
- Optimized versions of the following functions available for Intel® Advanced Vector Extensions (Intel® AVX)
  - BLAS: DGEMM
  - FFTs
  - VML: exp, log, and pow
  - See important information in the Intel® MKL User's Guide regarding the mkl_enable_instructions() function for access to these functions

# 2 System Requirements

## Supported Architectures and Terminology

Intel® Math Kernel Library supports the following architectures:

- **IA-32 Architecture** refers to systems based on 32-bit processors generally compatible with the Intel Pentium® processors, (for example, Intel® Pentium® 4 processor or Intel® Xeon® processor), or processors from other manufacturers supporting the same instruction set, running a 32-bit operating system.
- **Intel® 64 Architecture** refers to systems based on IA-32 architecture processors which have 64-bit architectural extensions, for example, Intel® Core™2 processor family, running a 64-bit operating system. If the system is running a 32-bit operating system, then IA-32 architecture applies instead. Systems based on AMD processors running a 64-bit operating system are also supported.
- **IA-64 Architecture** refers to systems based on the Intel® Itanium® processor running a 64-bit operating system.

## System Requirements

**Hardware**

To install and use Intel® MKL you will need a system with a supported processor and 1.3 GB of free hard disk space plus an additional 450 MB during installation for download and temporary files (host system only).

Supported processors:

- Intel® Core™ processor family
- Intel® Xeon® processor family
- Intel® Itanium® processor family
- Intel® Pentium® 4 processor family
- Intel® Pentium® III processor
- Intel® Pentium® processor (300 MHz or faster)
- Intel® Celeron® processor
- AMD Athlon* and Opteron* processors

Software

To use Intel® MKL you will need a supported compiler and MPI implementation.

Following is the list of supported operating systems:

- Red Hat* Enterprise Linux* 3, 4, 5 (IA-32 / Intel® 64 / IA-64)
- SUSE LINUX Enterprise Server* 9, 10 (IA-32 / Intel® 64 / IA-64)
- SGI ProPack* for Linux 4, 5 (Intel® 64 / IA-64)
- Red Hat* Fedora* 9 (IA-32 / Intel® 64)
- Debian* GNU/Linux 4.0 (IA-32 / Intel® 64 / IA-64)
- Ubuntu* 8.04 (IA-32 / Intel® 64)
- Asianux* Server 3 (IA-32 / Intel® 64 / IA-64)
- Turbolinux* 11 (IA-32 / Intel® 64 / IA-64)

  Note: These Linux* distributions are supported, and Intel® MKL should work on many more. If you have trouble with your distribution, do let us know.

Following is the list of supported C/C++ and Fortran compilers:

- Intel® Fortran Compiler 11.1 for Linux*
- Intel® Fortran Compiler 11.0 for Linux*
- Intel® Fortran Compiler 10.1 for Linux*

- Intel® C++ Compiler 11.1 for Linux*
- Intel® C++ Compiler 11.0 for Linux*
- Intel® C++ Compiler 10.1 for Linux*
- GNU Compiler Collection (gcc, g77, GNU Fortran 4.2.0 and later)
- Absoft* Pro Fortran v10.1 for Linux*
- PGI* Workstation Complete version 7.1.6

Following is the list of MPI implementations that Intel® MKL has been validated against:

- Intel® MPI Library Version 2.0, 3.0, 3.1, and 3.2.x (http://www.intel.com/go/mpi)
- MPICH2 version 1.0.x (http://www-unix.mcs.anl.gov/mpi/mpich)
- MPICH version 1.2.x (http://www-unix.mcs.anl.gov/mpi/mpich)
- Open MPI 1.2.x (http://www.open-mpi.org)
- SGI* MPT on Intel® 64 and IA-64 (http://www.sgi.com/products/software/mpt/)

  Note: Usage of MPI linking instructions can be found in the User's Guide in the doc directory.

**Note:**

- Parts of Intel® MKL have Fortran interfaces, and data structures, while other parts have C interfaces and C data structures. The User Guide in the doc directory contains advice on how to link to Intel® MKL with different compilers

# 3 Installation Notes

Guidance on the installation of Intel® MKL is provided at install time. Links will be provided to a file with step-by-step instructions (filename: Install.txt). This file can also be found in the doc directory.

# 4 Issues and Limitations

A full list of the known limitations of this release can be found on the knowledgebase for the Intel® MKL.

# 5 Notices

The following change is planned for future versions of Intel MKL. Please contact customer support if you have concerns:

- Content in the libraries containing ‘solver’ in the filenames will be moved to the core library in a future version of Intel MKL. These ‘solver’ libraries will then be removed.

# 6 Attributions

As referenced in the End User License Agreement, attribution requires, at a minimum, prominently displaying the full Intel product name (e.g. "Intel® Math Kernel Library") and providing a link/URL to the Intel® MKL homepage (www.intel.com/software/products/mkl) in both the product documentation and website.

The original versions of the BLAS from which that part of Intel® MKL was derived can be obtained from http://www.netlib.org/blas/index.html.

The original versions of LAPACK from which that part of Intel® MKL was derived can be obtained from http://www.netlib.org/lapack/index.html. The authors of LAPACK are E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. Our FORTRAN 90/95 interfaces to LAPACK are similar to those in the LAPACK95 package at http://www.netlib.org/lapack95/index.html. All interfaces are provided for pure procedures.

The original versions of ScaLAPACK from which that part of Intel® MKL was derived can be obtained from http://www.netlib.org/scalapack/index.html. The authors of ScaLAPACK are L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley.

PARDISO in Intel® MKL is compliant with the 3.2 release of PARDISO that is freely distributed by the University of Basel. It can be obtained at http://www.pardiso-project.org.

Some FFT functions in this release of Intel® MKL have been generated by the SPIRAL software generation system (http://www.spiral.net/) under license from Carnegie Mellon University. Some FFT functions in this release of the Intel® MKL DFTI have been generated by the UHFFT software generation system under license from University of Houston. The Authors of SPIRAL are Markus Puschel, Jose Moura, Jeremy Johnson, David Padua, Manuela Veloso, Bryan Singer, Jianxin Xiong, Franz Franchetti, Aca Gacic, Yevgen Voronenko, Kang Chen, Robert W. Johnson, and Nick Rizzolo.

# 7 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 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.

This document contains information on products in the design phase of development.

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.