

# The NCBI C++ Toolkit

## Release Notes (Version 9, May 2012)

Created: January 20, 2013.

Last Update: May 10, 2012.

- 
- [Download](#)
  - [Third Party Packages](#)
  - [Build](#)
  - [New Developments](#)
    - [HIGHLIGHTS](#)
    - [CORELIB](#)
    - [DATA SERIALIZATION](#)
    - [DATATOOL](#)
    - [CGI](#)
    - [NETWORK](#)
    - [UTILITIES](#)
    - [BIO-OBJECTS](#)
    - [BIO-TOOLS](#)
    - [ALIGNMENT MANAGER](#)
    - [BIO-OBJECT MANAGER](#)
    - [GENBANK DATA LOADER](#)
    - [BAM DATA LOADER](#)
    - [SRA DATA LOADER](#)
    - [cSRA DATA LOADER](#)
    - [BLAST](#)
    - [APPLICATIONS](#)
    - [BUILD FRAMEWORK \(UNIX\)](#)
  - [Documentation](#)
    - [Location](#)
    - [Content](#)
  - [Supported Platforms \(OS's and Compilers\)](#)
    - [Unix](#)
    - [MS Windows](#)
    - [Mac OS X](#)
    - [Added Platforms](#)
    - [Discontinued Platforms](#)

## Download

Download the source code archives at: [ftp://ftp.ncbi.nlm.nih.gov/toolbox/ncbi\\_tools++/ARCHIVE/9\\_0\\_0/](ftp://ftp.ncbi.nlm.nih.gov/toolbox/ncbi_tools++/ARCHIVE/9_0_0/)

- `ncbi_cxx--9_0_0.tar.gz` — for UNIX'es (see the list of UNIX flavors below) and MacOSX
- `ncbi_cxx--9_0_0.exe` — for MS-Windows (32- and 64-bit) / MSVC++ 9.0, 10.0 — self-extracting
- `ncbi_cxx--9_0_0.zip` — for MS-Windows (32- and 64-bit) / MSVC++ 9.0, 10.0

The sources correspond to the NCBI production tree sources, which are originally based on the development tree source snapshot from October 25, 2010 but also include many hundreds of important and safe code updates made since then and through March 28, 2011 (and then some).

There are also two sub-directories, containing easily buildable source distributives of the NCBI C Toolkit (for MS Windows and UNIX) and selected 3rd-party packages (for MS Windows only). These are the versions that the NCBI C++ Toolkit should build with. For build instructions, see README files there:

- NCBI\_C\_Toolkit
- ThirdParty

### Third Party Packages

Some parts of the C++ Toolkit just cannot be built without 3<sup>rd</sup> party libraries, and other parts of the Toolkit will work more efficiently or provide more functionality if some 3rd-party packages (such as BerkeleyDB which is used for local data cache and for local data storage) are available.

For more information, see the FTP README.

Table 1. Currently Supported/Tested Versions of Third Party Packages

Package	Versions expected to work (obtained by build-environment inspection in some cases)	Versions known to work (used in-house on any platform)
BerkeleyDB	4.3.0 or newer	4.5.20, 4.6.21.NC, 4.6.21.1, 4.7.25
Boost Test	1.35.0 or newer	1.35.0, 1.40.0, 1.41.0, 1.42.0, 1.45.0
FastCGI	All versions	2.1, 2.4.0
libbzip2	All versions	1.0.2, 1.0.3, 1.0.5, 1.0.6
libjpeg	All versions	6b
libpng	All versions	1.2.7, 1.2.10, 1.2.26, 1.2.42, 1.2.46, 1.4.11
libtiff	All versions	3.6.1, 3.8.2, 3.9.2, 3.9.4, 3.9.5
libungif	All versions	4.1.3 (libungif), 4.1.6 (giflib)
libxml2	All versions	2.6.26, 2.7.6, 2.7.8
libxslt	1.1.14	1.1.17, 1.1.24, 1.1.26
LZO	2.x	2.03, 2.05
PCRE	All versions	6.6, 7.8, 7.9
SQLite3	3.6.6 or newer	3.6.14.2, 3.6.22
Sybase	All versions	12.0-EBF9209, 12.5, 12.5.1

zlib	All versions	1.2.3, 1.2.3.3
------	--------------	----------------

For Mac OS X and UNIX OS's, the user is expected to download and build the 3<sup>rd</sup> party packages themselves. The release's package list includes links to download sites. However, the user still needs a list of the 3<sup>rd</sup> party packages and which versions of them are compatible with the release.

To facilitate the building of these 3rd-party libraries on Windows, there is an archive that bundles together source code of the 3rd-party packages, plus MSVC "solutions" to build all (or any combination) of them.

Table 2. Versions of Third Party Packages Included in the FTP Archive

Package	Depends On	Included Version <sup>a</sup>
BerkeleyDB		4.6.21.NC
Boost Test		1.42.0
libbzip2		1.0.2
libjpeg		6b
libpng	zlib 1.2.3	1.2.7
libtiff	libjpeg 6b, zlib 1.2.3	3.6.1
libungif		4.1.3
LZO		2.05
PCRE		7.9
SQLite3		3.6.14.2
zlib		1.2.3

<sup>a</sup> Applies to MSVC 9, MSVC 10

## Build

For guidelines to configure, build and install the Toolkit see [here](#).

## New Developments

### HIGHLIGHTS

- Added UNICODE support. Internally, the Toolkit works with UTF8-encoded strings; when needed, they are converted into UNICODE and back automatically.
- Sequence Align Manager API and code have been significantly revised and improved.
- Added BAM, SRA and cSRA (compressed SRA) readers and data loaders – based on the now incorporated NCBI SRA Toolkit.
- Object writers and readers: Added support for the VCF and GVF files formats.

**CORELIB***New functionality:*

- Added macros `BEGIN_*_NAMESPACE` that require semicolon and do not break indentation.
- Added `CObjectCounterLocker::MonitorObjectType()` for debugging.
- Added `CObjectCounterLocker::TransferLock()` and updated `CRef<>::Swap()` to support `MonitorObjectType()`.
- Added `ICanceled` interface.
- Allow locking input data in stream buffer.
- Added UNICODE support. Internally, the Toolkit works with UTF8-encoded strings; when needed, they are converted into UNICODE and back automatically.
- Added condition variables.
- Added fast double-to-string conversion, which is 5 to 10 times faster than standard `sprintf`.
- `CArg(Allow)_Int8` -- Accept explicitly marked (0x...) hexadecimal integer argument values.
- `MemoryAdvise()` -- Advise on memory usage for specified memory region (Unix).
- `CPerfLogger/CPerfLogGuard` -- Performance logger. Measure time spend somewhere and put result to the performance log.
- `FileAPI::SetHonorUmask()` -- add configurable parameter to honor umask settings (Unix).
- `NStr::StringToSizet/SizetToString` -- special case for `size_t` type, that size depends from platform/OS.
- `NStr::` -- added `[U]LongToString()` variants. Deprecate incompatible types for `[U]IntToString()`.
- `CProcess::GetCurrentHandle()` added
- `CDirEntry::SetOwner()` added
- An option for `CProcess::Daemonize()` to retain parent process

*Improvements:*

- Changed macro `NCBI_EAT_SEMICOLON` to extern declaration so that uniqueness is not required anymore.
- Use more efficient direct `type_info` comparison.
- Optimized `errno` access in `NStr::StringToXxx()`.
- Optimize MD5 step functions.
- Print stack trace with critical error messages.
- Optimized `StringToDoublePosix()`.
- Fixed processing `[+/-]infinity` and error detection in number parsers.
- Return starting ptr if no number parsed and there are leading spaces.
- Added missing method necessary for `CRef::Swap()`.
- Added more restrictive `bool` operator.
- Enable safer iterator-dealing code if `_GLIBCXX_DEBUG` is defined.

*Bug fixes:*

- Fixed parsing of "+" and "-" strings in NStr::StringToXxx().
- Fixed sorting of CVersionInfo and SDriverInfo.

**DATA SERIALIZATION***New functionality:*

- Added support of different forms of namespace qualification as described in XML Schema.
- Added mechanism of matching ASN binary data against set of known data types
- Added a suite of sample programs:

asn\_assign, asnwalk\_read, asnwalk\_type, asnwalk\_write, bss\_info, hooks\_commented  
 hooks\_copy\_member, hooks\_copy\_object, hooks\_copy\_variant, hooks\_read\_member,  
 hooks\_read\_object, hooks\_read\_variant, hooks\_skip\_member, hooks\_skip\_object,  
 hooks\_skip\_variant, hooks\_write\_member, hooks\_write\_object, hooks\_write\_variant,  
 seqannot\_splicer

- Allow cancellation via ICanceled callback.
- Implemented global pre-read hooks for class members and choice variants.

*Improvements:*

- Set internal names for inner classes, choices, and enums.
- Make error message in ThrowUnassigned() more informative for inner classes.
- Make configurable reading/writing of UTF8String tag.
- Allow ambiguous class names until lookup.
- Lock input data in buffer when guessing type.
- Make datatool -t argument optional, guess data type if necessary.
- Reimplemented pre-read member hooks as objects instead of function pointers.

*Bug fixes:*

- Stop AnyObject reading on closing brace.
- Fixed UTF8 processing in CopyString() and SkipString().
- Use UTF8String tag in ASN.1 binary.
- Correctly process UTF8 strings in datatool.
- Allow trailing spaces in delayed-parsing buffer.
- Reset typeinfo with null object pointer to make CObjectTypeInfo and CConstObjectInfo behave equally in bool context.

*XMLWrapp:*

- Respect the document encoding (if so) while printing or streaming nodes.

**DATATOOL**

- Added support of different forms of namespace qualification as described in XML Schema.
- Added possibility of matching ASN binary data against set of known data types. Modified code and schema generation to handle recursive element definitions.

- Added a code auto-generator for efficient traversal of records. Also eliminates the need for much hand-generated code. Note that the specification for autogenerated code is still in flux and is likely to change in the future.

## CGI

- CCgiResponse -- Let BeginPart and SetFilename take optional size arguments, with non-zero values yielding RFC 2183 size parameters (which are allowed to be approximate).
- CCgiUserAgent -- new methods IsBrowser(), GetEngineName() and GetPlatformName().
- CCgiUserAgent -- update list of browsers and mobile devices
- CCgiUserAgent -- implemented case insensitive parsing for user-agent strings; improved "Version/" parsing.
- CCgiUserAgent -- extended IsBot() and IsMobileDevice() methods to support exclusions lists.

## NETWORK

### *New features:*

- Identify HTTP connections with AppName in User-Agent Unicode (UTF-8) system error messages
- SOCK\_SetCork() added
- CONN to use new supplement (look ahead) I/O modes for faster I/O SERVICE connectors to maintain NCBI\_SID when detected CConn\_ServiceStream's ctor to accept customizable user-header CConn\_PipeStream to be able to get from the child's both STDOUT and STDERR output streams (merged together)
- CONN\_Status() to accept eIO\_Open
- Flex firewall mode (per checks with the help of CConnTest). Allow proxy leak configuration (to bypass the proxy if needed). CONN cancellation via callbacks. Allow bound sockets (listening and datagram) to use port "0" for OS auto-assign

### *Bugfixes:*

- SOCK\_Poll() to return error on closed / unopened sockets Detect quotes in the CONN environment / registry settings FTP stream to issue errors on incomplete uploads
- ConnNetInfo\_ParseURL() to fail for too long URL elements (no truncation)
- ConnNetInfo\_ExtendHeader() to check for duplicates (and avoid them)
- CConn\_MemoryStream::ToCStr() deprecated and dropped
- CRateMonitor::GetETA() fixed
- Workaround IAC-unclean implementations of FTPD
- CORE\_GetUsername() to avoid using getlogin() if at all possible

## UTILITES

- CFormatGuess -- Recognize BAM and (c)SRA files.
- misc/clog/ncbi\_c\_log.h -- new C logging API.
- Added CTar::Append() on a stream object

## BIO-OBJECTS

### *New functionality:*

- Added CSeq\_interval, CSeq\_loc\_mix, CPacked\_seqint getters for start and stop.
- Added method CompareOrdered() and predicate PLessOrdered to CSeq\_id\_Handle.

### *Improvements:*

- CSeq\_loc::Compare() now checks Seq-id and circular locations.
- Adjusted CSeq\_feat comparison to match C Toolkit and CFeat\_CI.
- Added CPDB\_seq\_id::Compare().
- Prefer gnl|ti|\* over gnl|TRACE|\*.
- Scoring functions made non-inlined.
- Added matching of integer and string versions of the same general id.
- Changed Seq-table's multi-utf8-string container type to vector.
- Added vector reserve in pre-read hooks for Seq-table and Dense-seg.
- Added reserving pre-read hooks for Seq-graph data.
- Reserve Byte-graph vector too for text formats.
- CSeq\_id::IdentifyAccession -- Implement or improve recognition for more prefixes (FN-FO, GB, HV-HX, JJ-JU, KA-KZ, and LA-LJ) and mixed-in EMBL TPA protein accessions.
- Rework publication labeling logic, designating the historic implementation as version 1 and adding a version 2 corresponding to GenBank/GenPept REFERENCE JOURNAL fields and the like. One difference (among many!) between version 1 and 2 labels is that the latter generally leave off item-specific author and title information, which would appear in neighboring flat-file fields.
- CFastaReader -- Rework split sequence packing to support optionally yielding gap literals as appropriate, potentially making for considerable savings (particularly for nucleotide data). NB: Because this feature changes the FASTA reader's traditional interpretation of data, enabling it requires supplying the new fLetterGaps flag. Add another avenue for space saving in the form of a new fNoUserObjs flag, which disables storage of raw defines.

### *Bug fixes:*

- Fixed CTextseq\_id::Compare() for STL requirements.
- Use MT-safe way to get CSeq\_id from CSeq\_id\_Handle.

## BIO-TOOLS

### *New Development:*

- Object writers and readers: Added support for the VCF and GVF files formats.

### *Improvements of existing object readers:*

- Wiggle: Added support for input files containing multiple tracks.
- Wiggle: Added support for input spanning multiple chromosomes.
- Bed: Chunked down huge (more than 100,000 records) tracks into a sequence of manageably sized (at most 100,000 features) feature tables.

- Bed: Recognized lines beginning with “#” as comment lines.
- Newick: Made node labels in Newick tree input optional.

*Improvement of existing object writers:*

- GFF3: Added support for writing alignment files.
- GFF3: Added support for a wide variety of ASN.1 attributes to be emitted as GFF3 attributes.
- GFF3: Added support for escaping characters that are considered special by the GFF3 spec.
- GFF3: Added support for source features.
- GFF3: Added support for assembly identifying, sequence region and species pragmas.
- GFF3: Added support for SOFA vocabulary.
- GFF3: Added support for circular sequences.
- GFF3: Added full headers identifying GFF3 revision spec and generating agent.
- Wiggle: Generated more compact byte graphs by pre-analyzing data before final processing.
- GTF: Made output for gene features and exon numbers optional (flag controlled).

*Other improvements:*

- Enhanced parsing and formatting of RepeatMasker repeat data. The improvements include greater fidelity in preserving information from the native tabular RepeatMasker output, and better standardization in representing repeat features.
- Flatfile generator:
  - Numerous changes to fix various incompatibilities with the C flatfile generator or bugs
- Feature Table Reader:
  - More lenient parsing
  - Better error reporting (e.g. line number)
  - 6th column for features to indicate minus strand
- ASN.1 Record Cleanup:
  - Changed to match C cleanup.
  - Uses autogenerated code, generated by changes to datatool
- CAgpReader:
  - Handle AGP 2.0
- AGP Writer
  - Output AGP 2.0 instead of AGP 1.1
- FASTA reader
  - Can keep track of bad mods and bad residues
- CFastaOstream
  - Now lets you output [key=value] modifiers if requested. Only a few basic ones currently appear, though.



**COBALT**

- COBALT now can align two multiple sequence alignments.
- COBALT now can use pre-computed RPS-BLAST alignments saved in BLAST archive format
- Fixed producing alignments in NEXUS format

**ALIGNMENT MANAGER***Improvements:*

- Added support for gaps/indels in pairwise, anchored and sparse alignments.
- Redesigned sparse alignment iterator to show more segment types.

**BIO-OBJECT MANAGER***New functionality:*

- Added API to work with annot zoom tracks.
- Added CSeq\_annot\_Handle::ReorderFtable().
- Implemented CFeat\_CI iterator of features by id/xref.
- Allow cancelling all tasks in prefetch manager.
- Added faster CScope::IsSameBioseq().

*Improvements:*

- Implemented gap annotation mapping on segset entries.
- Avoid using catch (...). Use guard class for cleanup instead of catch(...).
- Order Object-id.str with numeric value together with Object-id.id.
- Treat Gene-ref.desc as locus if necessary.
- Use gt/lt to indicate completely cut intervals in mapped annotations.
- CTableFieldHandle::TryGet() now returns false instead of throwing an exception.
- Check all mRNA features passing expensive check in GetBestMrnaForCds().
- Try to find mRNA parent even if gene link exists in CFeatTree.
- Keep blob cache size to avoid calling slow list<>::size().
- Make CAnnotCollector::x\_MatchLimitObject() non-inlined as it's used in other file.
- Use xref before overlap check in CFeatTree.
- Allow adding chunks to active split info in MT.
- Optimized CTSE\_Info::GetAnnotIds().
- Avoid multiple overlap test for multi-id locations in CFeatTree.
- Avoid unnecessary loading split Seq-data via GetInst().
- Heavily redesigned Compare() and TestForOverlap() functions to fix many problems and add more flexibility.

*Bug fixes:*

- Update annot index before looking for Seq-feat handle.
- Skip empty ranges when looking up for a feature handle.
- Fixed interval matching with circular locations in CHandleRangeMap.

- Fixed IgnoreStrand matching in CHandleRangeMap.
- Fixed loss of features with multi-id and circular locations.
- Fixed self-alignment duplication.
- Fixed alignment duplication on a boundary between chunks.
- Fixed objects::sequence::GetStop() off-by-1 error.
- Do not treat location with null as multi-id.
- Safe-guard against exceptions from TestForOverlap().
- Fixed CMappedFeat::GetLocationId() for mix locations.
- Repeat chunk loading if necessary as they may be added later.

## GENBANK DATA LOADER

### *New functionality:*

- Allow cache writer without reader.

### *Improvement:*

- Added workaroung to empty skeleton data when it's attached to split-info.
- Use more efficient retrieval of blobs together with versions.
- Added one-time check for ICache version compatibility.
- Added ID2 replies' retry-delay processing.

### *Bug fixes:*

- Avoid slash in cache subkey due to conflict with storing blobs in files.
- Skip zoom-level annotations instead of failing \_ASSERTion().
- Fixed loss of chunks' split and blob versions after loading from cache.
- Avoid copying uninitialized iterators.

## BAM DATA LOADER

### *New functionality:*

- Added support for SRZ cfg and graph.
- Added map quality and annot name.
- Added mate pairing getters.
- Added GetFlags() for real BAM files.
- Added configurable SPOT\_ID detector.
- Allow independent creation of CSeq\_align, CSeq\_annot, and short CBioseq.

### *Improvements:*

- Added detailed SRA SDK error reporting.
- Added more CIGAR types.
- Relax alignment size limit.
- Allow direct BAM file name in arguments.
- Convert numeric refseq ids < 1000 to local Seq-id.
- Added CDataLoader::GetPossibleAnnotNames().

- Implemented separate loading of alignments and short sequences.

*Bug fixes:*

- Fixed skipping entries with low quality.
- Reverse short sequences aligned with minus strand.
- Fixed soft clipping at the end of sequence.

## **SRA DATA LOADER**

*New functionality:*

- Implemented SRA and SRZ lookup in readers.

*Improvements:*

- Added explicit schema registration for XCODE and Windows.
- Separated SRA, SRZ, and BAM paths.

## **cSRA DATA LOADER**

*New functionality:*

- Implemented cSRA reader for alignments, short reads, coverage graphs, and pileup graphs.
- Added IidMapper to cSRA reader.
- Added mate read info.
- Added GetShortRead() to get reversed read sequence.
- Added GetMatchAnnot() to get annot with align only.
- Added generation of read quality graph.
- Added generation of reference sequence if necessary.

*Improvements:*

- Added check for SRA value read success.
- Added query arguments to test applications.
- Added mate ref-id if different.
- Changed local ids to a more efficient ints.
- Explicit mate pair info generation made optional.
- Added Seq-annot.desc for mate match indication.
- Allow SRR accession to be passed to RegisterInObjectManager.
- Added GetIds().

*Bug fixes:*

- Fixed soft clipping shift.
- Fixed seq-id mapping.

## BLAST

### *New functionality:*

- Added DELTA-BLAST - an new protein sequence alignment tool that constructs a PSSM using the results of a Conserved Domain Database search and searches a sequence database.
- Added a new API CDeltaBlast (see above)
- Added a new command line binary deltablast (see above)
- Added support for bl2seq in blast\_formatter
- Expose psiblast command line option to ignore the MSA master sequence
- Added blastdbcp, seedtop, deltablast, and makeprofiledb application
- Added new FSC statistics
- For more details, see BLAST+ 2.2.26 release notes.

## APPLICATIONS

### *Improvements to utilities:*

- Multireader: Add support for Newick trees, GVF, VCF and (bare bones) text alignment files.
- Annotwriter: Add support for GVF, and VCF formats.

### *agp\_validate*

- AGP version 2.0. Extra checks for chromosome-from-scaffold and scaffold-from-component files (-scaf, -chr).
- It is OK to not include the red-lined commits.
- Added agp/fasta comparison functionality

## BUILD FRAMEWORK (UNIX)

- The build system has been optimized to take advantage of GNU Make, which is now a requirement. (Support for Solaris Make is limited to chaining to gmake.)
- On Mac OS X, built bundles in addition to dynamic libraries only when explicitly given the new --with-bundles flag.
- Added a --with-relative-runpath=... flag to simplify specifying executable-relative runpaths, which could be cumbersome to quote and escape (and which previous releases mishandled slightly), and a --with-hard-runpath flag to favor old dynamic tags when the choice exists. (The default is now to use new tags, honoring LD\_LIBRARY\_PATH).
- Added --with(out)-symbols flags to allow decoupling -g from -D\_DEBUG.

## CHANGES TO COMPILER SUPPORT

- Supported GCC versions now range from 4.0.x (released in 2005) to 4.6.x (the current stable series). In addition, the Toolkit now supports building with LLVM Clang 3.0, at least on Mac OS X.

## Documentation

### Location

The documentation is available online as a searchable book "The NCBI C++ Toolkit": <http://www.ncbi.nlm.nih.gov/books/NBK7160/>.

The C++ Toolkit book also provides PDF version of the chapters. The PDF version can be accessed by a link that appears on each page.

### Content

Documentation has been grouped into chapters and sections that provide a more logical coherence and flow. New sections and paragraphs continue to be added to update and clarify the older documentation or provide new documentation. The chapter titled "Introduction to the C++ Toolkit" gives an overview of the C++ Toolkit. This chapter contains links to other chapters containing more details on a specific topic and is a good starting point for the newcomer.

A C/C++ Symbol Search query appears on each page of the online Toolkit documentation. You can use this to perform a symbol search on the up-to-date public or in-house versions using source browsers LXR, Doxygen and Library - or do an overall search.

**HEADS-UP:** We have switched our source control system from CVS to SVN (Subversion). From outside NCBI it's available:

- For browsing: <http://www.ncbi.nlm.nih.gov/viewvc/v1/trunk/c++>
- For retrieval: <http://anonsvn.ncbi.nlm.nih.gov/repos/v1/trunk/c++> (**NOTE:** Some WebDAV clients may require `dav://` instead of `http://`)

## Supported Platforms (OS's and Compilers)

- UNIX
- MS Windows
- Mac OS X
- Added
- Discontinued

This release was successfully tested on at least the following platforms (but may also work on other platforms). Since the previous release, some platforms were dropped from this list and some were added. Also, it can happen that some projects would not work (or even compile) in the absence of 3rd-party packages, or with older or newer versions of such packages. In these cases, just skipping such projects (e.g. using flag "-k" for make on UNIX), can get you through.

In cases where multiple versions of a compiler are supported, the mainstream version is shown in **bold**.

### UNIX

Table 3. UNIX OS's and Supported Compilers

Operating System	Architecture	Compilers
CentOS 5.4 (LIBC 2.5)	x86-64	<b>GCC 4.4.2</b> , 4.0.1 <sup>a</sup> , 4.1.2 <sup>a</sup> , 4.3.3 <sup>a</sup> , 4.6.0 <sup>a</sup> , 4.6.3 <sup>a</sup>
CentOS 5.4 (LIBC 2.5)	x86-32	<b>GCC 4.4.5<sup>a</sup></b> , <b>4.6.0</b>

Ubuntu 9.04 ("jaunty") (LIBC 2.9)	x86-32 x86-64	<b>GCC 4.3.3</b>
Solaris 10	SPARC	GCC 4.1.1 <sup>b</sup> , 4.5.3 <sup>b</sup> <b>Sun Studio 12 (C++ 5.9)</b> , Sun Studio 12 Update 1 (C++ 5.10) <sup>a</sup> Oracle Studio 12.2 (C++ 5.11) <sup>a</sup>
Solaris 10	x86-32	GCC 4.2.3 <b>Sun Studio 12 (C++ 5.9)</b> , Sun Studio 12 Update 1 (C++ 5.10) <sup>a</sup> Oracle Studio 12.2 (C++ 5.11) <sup>a</sup>
Solaris 10	x86-64	<b>Sun Studio 12 (C++ 5.9)</b> , Sun Studio 12 Update 1 (C++ 5.10) <sup>a</sup> Oracle Studio 12.2 (C++ 5.11) <sup>a</sup>
FreeBSD-8.0	x86-32	GCC 4.2.1

<sup>a</sup> some support

<sup>b</sup> 32-bit only

## MS Windows

Table 4. MS Windows and Supported Compilers

Operating System	Architecture	Compilers
MS Windows	x86-32	MS Visual C++ 2008 (C++ 9.0), 2010 (C++ 10.0) NOTE: We also ship an easily buildable archive of 3rd-party packages (including NCBI C Toolkit) for this platform.
MS Windows	x86-64	MS Visual C++ 2008 (C++ 9.0), 2010 (C++ 10.0) NOTE: We also ship an easily buildable archive of 3rd-party packages (including NCBI C Toolkit) for this platform, although not all libraries are available for x86-64 architecture.
Cygwin 1.5.25	x86-32	GCC 4.5.3- nominal support only.

## Mac OS X

Table 5. Mac OS and Supported Compilers

Operating System	Architecture	Compilers
Mac OS X 10.5, MacOS x 10.6	Native (PowerPC <u>or</u> x86-32 <u>or</u> x86-64 )	Xcode 3.0 - 3.2.6
Darwin 8.x, 9.x, 10.x	Native (PowerPC <u>or</u> x86-32 <u>or</u> x86-64), Universal (PowerPC <u>and</u> x86-32)	GCC 4.0.1 GCC 4.2.1 (only available under Darwin 10.x) Clang 3.0

**NOTE:** the correspondence between Darwin kernel versions and Mac OS versions:

Darwin 8.x = Mac OS 10.4.x

Darwin 9.x = Mac OS 10.5.x

Darwin 10.x = Mac OS 10.6.x

## Added Platforms

Table 6. Added Platforms

Operating System	Architecture	Compilers
MS Windows	x86-32, x86-64	MS Visual C++ 2010 (C++ 10.0)

CentOS 5.4 (LIBC 2.5)	x86-32	GCC 4.4.5 <sup>a</sup> , 4.6.0
Mac OS X 10.5, MacOS x 10.6	Native (PowerPC <u>or</u> x86-32 <u>or</u> x86-64 )	Xcode 3.2.3 - 3.2.6 Clang 3.0

<sup>a</sup> some support

### Discontinued Platforms

Table 7. Discontinued Platforms

Operating System	Architecture	Compilers
MS Windows	x86-32, 64	MS Visual C++ 2005 (C++ 8.0)
FreeBSD-6.1	x86-32	GCC 3.4.6
All	All	All GCC 4.0.1 and below

### Last Updated

This section last updated on May 10, 2012.