

The NCBI C++ Toolkit

Release Notes (May, 2009)

Created: June 23, 2009.

Last Update: June 23, 2009.

-
- [Download](#)
 - [Third Party Packages](#)
 - [Build](#)
 - [New Developments](#)
 - [CORELIB — Portability and Application Framework](#)
 - [CONNECT — Data streaming, Networking, and Dispatching](#)
 - [UTIL — Miscellaneous Low-Level APIs](#)
 - [SERIAL — Data Serialization \(ASN.1, XML, JSON\)](#)
 - [DATATOOL — Code Generator and Data Converter Tool](#)
 - [CGI — CGI and Fast-CGI Application Framework](#)
 - [DBAPI — Generic SQL Database Connectivity](#)
 - [BIO-OBJECTS — Bio-Object Specific Utility Functions \(Not Involving OM++\)](#)
 - [Local data storage \(LDS\)](#)
 - [ALGO/ALIGN — Generic Alignment Algorithms](#)
 - [COBALT — Multiple sequence alignment tool](#)
 - [OM++ — Object Manager — For Retrieving and Processing Bio-Objects](#)
 - [OM++ LOADERS/READERS — Data Retrieval Libraries for OM++](#)
 - [OBJTOOLS/SEQMASKS_IO — input/output of masked sequences](#)
 - [ID2 Communication Protocol](#)
 - [BIO-TOOLS](#)
 - [PTB — Project Tree Builder](#)
 - [APPLICATIONS](#)
 - [Grid — Distributed Computation Client APIs](#)
 - [BLAST](#)
 - [Miscellaneous](#)
 - [Documentation](#)
 - [Location](#)
 - [Content](#)
 - [Supported Platforms \(OS's and Compilers\)](#)
 - [Unix](#)
 - [MS Windows](#)
 - [Mac OS X](#)
 - [Added Platforms](#)

- Discontinued Platforms
- Caveats and Hints
 - GCC 3.0.4
 - GCC 3.4.x, 4.0.x
 - ICC 8.0

Download

Download the source code archives at:

ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/2009/May_15_2009/

- [ncbi_cxx—May_15_2009.tar.gz](#) — for UNIX'es (see the list of UNIX flavors below) and MacOSX
- [ncbi_cxx—May_15_2009.gtar.gz](#) — for UNIX'es (see the list of UNIX flavors below) and MacOSX
- [ncbi_cxx—May_15_2009.exe](#) — for MS-Windows (32- and 64-bit) / MSVC++ (8.0, 9.0) — self-extracting
- [ncbi_cxx—May_15_2009.zip](#) — for MS-Windows (32- and 64-bit) / MSVC++ (8.0, 9.0)

The sources correspond to the NCBI production tree sources, which in turn roughly corresponds to the development tree sources from February 6, 2009.

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 3rd 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](#).

The following table shows the versions of 3rd party packages that are believed to be compatible with the C++ Toolkit.

Table 1. Compatible Versions of Third Party Packages

Package	FreeBSD 32	Linux 32	Linux 64	Mac OS X	SunOS x86	SunOS SPARC	Windows ^a
Berkeley DB	4.4.20	4.6.21.1	4.6.21.1	4.5.20	4.3.21	4.5.20	4.5.20.NC ^b
Boost Test	1.35.0	1.35.0	1.35.0	1.35.0	1.35.0	1.35.0	1.35.0 ^b
FastCGI	-	2.4.0	2.4.0	-	2.1	2.4.0	-
libzip2	current	current	current	current	current	current	1.0.2 ^b

Package	FreeBSD 32	Linux 32	Linux 64	Mac OS X	SunOS x86	SunOS SPARC	Windows ^a
libjpeg	current	current	current	current	current	current	6b
libpng	current	current	current	current	current	current	1.2.7
libtiff	current	current	current	current	current	current	3.6.1
libungif	current	current	current	current	current	current	4.1.3
LZO	-	2.02	2.02	2.02	2.02	2.02	2.02 ^b
MySQL	-	-	-	-	-	3.23.40	3.23.55
PCRE							4.3
SQLite3	3.6.2	3.3.5	3.3.5	-	-	-	3.6.2
Sybase	-	12.5.0.6-ESD13	12.5.0.6-ESD13	-	12.5.1	12.0-EBF209	-
zlib	current	current	current	current	current	current	1.2.3 ^b

^a Applies to MSVC 2005 and 2008. Unless otherwise noted, 32-bit is supported and 64-bit is not supported.

^b MSVC 2005 64-bit is supported.

For Mac OS X and UNIX OS's, the user is expected to download and build the 3rd party packages themselves. The release's package list includes links to download sites. However, the user still needs a list of the 3rd 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 ^a
Berkeley DB		4.5.20.NC ^b
Boost Test		1.35.0 ^b
libbz2		1.0.2 ^b
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		2.02 ^b
MySQL		3.23.55
PCRE		4.3
SQLite3		3.6.2

Package	Depends On	Included Version ^a
zlib		1.2.3 ^b

^a Applies to MSVC 2005 and 2008. Unless otherwise noted, 32-bit is supported and 64-bit is not supported.

^b MSVC 2005 64-bit is supported.

Build

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

New Developments

(*) — potentially backward-incompatible changes.

CORELIB — Portability and Application Framework

- (*)CTime — removed deprecated operators to add/subtract/increment/decrement days.
- CTime — added CTime::SetTimeTM() to convert from an arbitrary "struct tm" value; CTime::GetTimeTM() to convert the current local time to a "struct tm"; and CTime::CTime(const struct tm& t, ETimeZonePrecision tzp).
- CStaticTls — new class that combines CSafeStaticRef and CTls but requires less overhead.
- NCBI-Boost unit testing framework — test_boost library now includes Boost.Test library as a whole, so there's no need to link against Boost.Test library separately when linking against test_boost.
- NCBI-Boost unit testing framework — now arranges to incorporate Boost.Test's (compiled) code into libtest_boost, thereby requiring only headers from Boost.
- NCBI-Boost unit testing framework – macro
NCBI_BOOST_NO_AUTO_TEST_MAIN is now obsolete (everything works whether or not it is used).
- NCBI-Boost unit testing framework — added support for timed out and skipped tests and for timed out units.
- NCBI-Boost unit testing framework — introduced new analogs of BOOST_CHECK* macros that also check the NO_THROW condition.
- CAAtomicCounter_WithAutoInit — new class that enables creating counters initialized with a specified value.
- CNcbiResourceInfo and CNcbiResourceInfoFile — new classes providing sensitive data encryption/decryption.
- CNcbiRegistry — reworked once more, with generic functionality factored out into a new CCompoundRWRegistry class; now allows automatic loading of subregistries listed in the ".Inherits" entry of the "NCBI" section, shadowing of set entries on lower layers by explicitly empty entries on higher layers, and the environment variable NCBI_CONFIG_OVERRIDES to name an extra high-priority configuration file.
- CAutoEnvironmentVariable — new class to allow setting environment variables for limited durations, with restoration of previous values when instances go out of scope.

- CExceptionReporter, CDiagMatcher — added diagnostic filtering by error code when reporting exceptions. Diagnostic filtering is set by SetDiagFilter function.
- CStringPairsParser — converted into CStringPairs template to allow using different containers, added methods for merging data into a single string.
- CProcess — new method Daemonize(), formerly a static function resided in ncbi_os_unix.h.
- GetPhysicalMemorySize() — implemented on a wider variety of platforms.
- CDirEntry — A new mode (eRecursiveIgnoreMissing) for the Remove() method was introduced. This mode deletes the files and directories recursively but ignores the errors when the file being deleted does not exist.
- The DeleteReadOnlyFiles CParam<> and the corresponding Set() method of CFileAPI were introduced to work around CDirEntry::Remove() not being able to remove read-only files on Windows.

CONNECT — Data streaming, Networking, and Dispatching

- Implemented HTTPS protocol support.
- Added new function UTIL_PrintableString[Size]().
- Added new function SERV_ServerPort();
- Retired wsock32 use on Windows; require (and link against) ws2_32 exclusively.
- Fully implemented new TRIGGER primitive.
- CSocket, SOCK_XXX — now can work via SSL (requires GNUTLS API).
- Introduced new (and centralized) ESOCK_Flags for various socket attributes, and used them throughout the library (deprecating older equivalents).
- The default connection managing mode was changed to eKeepConnection.

UTIL — Miscellaneous Low-Level APIs

- CZipCompression — added support for concatenated gzip files.
- CFormatGuess — the Format() method now throws an exception in the case of missing input. Format guessing was expanded to include streams other than file streams. FASTA files can now be properly recognized even in the presence of some non-printing characters or very long defines. Added "guessing" support for the BED, BED15, and WIGGLE file formats. For the AGP format, the presence of line comments starting with a '#' is now allowed.
- CTar — improved to work safely with program pipes (including buggy implementations in certain GLIBC releases).

SERIAL — Data Serialization (ASN.1, XML, JSON)

- CObjectOstreamJson — corrected the writing of UTF8 strings in JSON format.

DATATOOL — Code Generator and Data Converter Tool

- Added possibility to override code generation command line arguments in a DEF file.
- Added support for more data types, including dateTime, time, short, byte, negativeInteger, nonNegativeInteger, positiveInteger, nonPositiveInteger, unsignedInt, unsignedShort, and unsignedByte.
- Fixed several bugs in XML Schema parsing.

CGI — CGI and Fast-CGI Application Framework

- CGI library — applications can no longer send secure cookies over insecure connections.
- Fixed parsing of indexes (arguments without values) in query string.
- A new constructor `CCgiRequest::CCgiRequest(CNcbiIstream&)` was defined to allow construction of `CCgiRequest` by deserialization from a stream.

DBAPI — Generic SQL Database Connectivity

- `IResultSet` — `BindBlobToVariant()` now does nothing - how the user wants to read blobs is now automatically determined by `Read()` (which also means by `NcbiIstream`) or by `GetVariant()`.
- Fixed incorrect column naming in the SELECT results with Sybase when the column is just renaming a real column in the database.
- Introduced possibility to add hints for bulk insert operations (such as `CHECK_CONSTRAINTS`, `FIRE_TRIGGERS`, etc).
- `flds8` driver is not built automatically anymore and its sources will be removed in the next release.
- Added support for connection to mirrored databases for which master/mirror relations can be switched while application is working.

BIO-OBJECTS — Bio-Object Specific Utility Functions (Not Involving OM++)

- `CSeq_id::IdentifyAccession` — added or improved recognition for the prefixes AH, AL, BX, CR, CT, CU, DAAA-DZZZ, DM, FT, FU, GJ-GL, GO-GZ, HA-HD, and HAA-HZZ and some more (mixed-in) EMBL TPA protein accessions.

Local data storage (LDS)

- Dramatically reduced memory usage in LDS indexer by processing entries separately.

ALGO/ALIGN — Generic Alignment Algorithms

- `CCompartmentFinder` - the maximum intron length parameter has been exposed; the default max intron length increased to 1.2M bases.
- `CCompartmentAccessor` — added `AsSeqAlignSet()` method to support ASN.1 output.

COBALT — Multiple sequence alignment tool

- (*) Changed API and provided `CMultiAlignerOptions` class that provides multiple alignment parameters.
- (*) Rearranged the parameters for the COBALT demo application.
- Added query clustering functionality for multiple alignment.
- Added computation time improvements to multiple alignment.
- Added functionality for computing progressive alignment guide tree as a clustering dendrogram.

OM++ — Object Manager — For Retrieving and Processing Bio-Objects

- `CSeq_entry_EditHandle` now has method `TDescr& SetDescr()` which works for both bioseqs and bioseq-sets.
- Object manager now recognizes named annotation accessions with versions.

- Object manager now indexes non-feature Seq-tables, and allows extra columns for feature Seq-tables.
- Bioseq and Bioseq_set objects now forbid empty description field by default.
- The default timeout for opening an ID1/ID2 connection was reduced from 20 to 5 seconds. The default timeout after connection is established remains 20 seconds.
- Added sequence::FindLatestSequence() method for searching through a Bioseq's history.
- Fixed mapping of seq-graph data. The old version mapped only graph location, but left the related data array unchanged. The fix creates a new array and copies only the mapped portion of the data. As a result of this, a new valid seq-graph object is created (the old version created seq-graph with incorrect data). Added methods to CGraph_CI for checking mapped ranges without creating the whole mapped data array.

OM++ LOADERS/READERS — Data Retrieval Libraries for OM++

- Added support for the UCSC WIGGLE file format. Enhanced support for UCSC BED and microarray file formats.
- Preliminary version of an idmapper component that translates UCSC sequenceIDs to their corresponding GI IDs.

OBJTOOLS/SEQMASKS_IO — input/output of masked sequences

- Fixed a problem with sequence writer omitting accessions when ID parsing is requested.

ID2 Communication Protocol

- Blob-state information in ID2 reply was changed from ENUMERATED to INTEGER with bit flags. This field is not yet used by ID2 server or ID2 GenBank reader.

BIO-TOOLS

- Implemented a faster non-OM version of sequence::GetTitle(CBioseq), although it may fail if called when OM is necessary.
- format — The output of the flat file generator in the C++ Toolkit is intended to be essentially the same as the output in the C Toolkit. That was not achieved in the last release, so changes were made in this release to make the output more closely match the output obtained using the C toolkit.

PTB — Project Tree Builder

- Improved analysis of library-to-library dependencies; added possibility to enforce build order of static libraries.
- Enhanced the structure of the flat makefile on UNIX to speed up the build by reducing the number of calls to 'make' and avoiding attempts to build the same project more than once.

APPLICATIONS

Grid — Distributed Computation Client APIs

- CNetServiceAPI_Base — removed along with its unused/duplicate methods; the remaining methods were moved to SNetServiceImpl.
- All *Sink classes and interfaces — removed.

- CNetScheduleKeys_Base and CNetScheduleKeys — replaced by CNetScheduleKeys.
- CNetScheduleClient_LB and CNetScheduleClient — removed; CNetScheduleAPI should be used instead.
- CNetCacheClient and CNetCacheClient_LB — declared as deprecated.
- CNetScheduleSubmitter::GetJobDetails — implemented batch retrieval of job results. This enables retrieving results for large groups of jobs in smaller batches, rather than for all jobs at once.
- Implemented method CountActiveJobs() in NetScheduleAdmin and made it available via the netschedule_control command line utility in the form of the new -count_active command.
- Implemented method GetBlobSize() in CNetCacheAPI and made it available via the -size parameter of the netcache_control utility.
- LBSM affinity pass-through was implemented, by reading the affinity from the LBSMD configuration file and using it when querying the load balancer.
- All public Grid API classes were converted to "components" featuring reference counting. Component implementations were moved to the respective *Impl structures in the implementation part of the library (src/). These structures are accessible via the "->" operator.
- Unused/duplicate methods of CNetService were removed. Public methods of other classes that were only used internally, were moved to the implementation part.
- netschedule_control, ns_remote_job_control — New -cancel command.
- netcache_check — Updated to use CNetCacheAPI (instead of older CNetCacheClient).
- netschedule_node_sample — Updated to use new Grid APIs.
- remote_app — Fixed the error which resulted in inability to properly clean up temporary directories.
- netschedule_admin — Removed because it was older and less capable than netschedule_control.
- grid_mgr — Removed due to lack of use.
- Client-side Grid tools were moved from under app/netschedule/ to a new directory app/grid/. They also were assigned their own version numbers.

BLAST

- blast —
 - 1 Added partial sequence fetching to the BLAST+ command line applications. Modified the 2-hit algorithm so that no overlap between two hits is allowed.
 - 2 Split the BLAST database data loader into local and remote components.
 - 3 Bug fixes and performance improvements to subject masking.
- convert2blastmask — New application to convert masking information in lower-case masked FASTA input to file formats suitable for makeblastdb.
- makeblastdb, segmasker — Bug fixes.
- blastdbcmd — Added support for displaying masking information.

Miscellaneous

- multireader — A universal reader for all NCBI supported UCSC file formats.

- `formatguess` — Front end for the toolkit `format_guess` component to automatically determine NCBI supported file formats.
- `splign` — The maximum intron length parameter has been exposed (`-max_intron`).
The default max intron length was increased to 1.2M bases.
- `id1_fetch` — Now has `-maxplex`, `-extfeat`, and `-timeout` options.
- `id1_fetch_simple`, `id2_fetch_simple` — Now have options for arbitrary requests, and for saving replies in file.
- `compart` — A new parameter `min_query_len` introduced to specify the minimum length for transcripts for indexing. Transcripts shorter than `min_query_len` will be ignored. The default is 50 bases.
- `Alnmgr` — Added the ability to load sequence(s) into scope via:
 - 1 `Seq-entry` (`se_in` flag)
 - 2 `FASTA` (`fasta_in` flag)
 - 3 `BLAST db` (`blastdb` flag)

Documentation

Location

The documentation is available online as a searchable book "The NCBI C++ Toolkit": .

The C++ Toolkit book also provides PDF version of the chapters; although these are not up to date in this release. 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 Entrez, LXR, Doxygen and Library - or do an overall search.

HEADS-UP: We have switched our source control system from CVS to SVN (Subversion). Unfortunately, the SVN repository cannot (yet) be accessed from outside NCBI.

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 default version is shown in bold.

UNIX

Table 3. UNIX OS's and Supported Compilers

Operating System	Architecture	Compilers
Linux-2.6.x (LIBC 2.3.5)	x86-32	GCC 3.0.4 ^a , 3.4.2 , 4.1.2 ^a , 4.2.3 ^a , 4.3.3 ICC 8.0 , 10.1
Linux-2.6.x (LIBC 2.3.5)	x86-64	GCC 4.0.1 , 4.1.2, 4.2.3 ^b , 4.3.3 ICC 9.1 , 10.1
Solaris 10	SPARC	GCC 4.1.1 ^c Sun Studio 12 (C++ 5.9)
Solaris 10	x86-32	GCC 4.2.3 Sun Studio 12 (C++ 5.9)
Solaris 10	x86-64	Sun Studio 12 (C++ 5.9)
FreeBSD-6.1	x86-32	GCC 3.4.6
Darwin 8.x, 9.x	Native, Universal	GCC 4.0.1

^a some support

^b nominal support

^c 32-bit only

MS Windows

Table 4. MS Windows and Supported Compilers

Operating System	Architecture	Compilers
MS Windows	x86-32	MS Visual C++ 2005 (C++ 8.0), 2008 (C++ 9.0) 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++ 2005 (C++ 8.0), 2008 (C++ 9.0)
Cygwin 1.5.25	x86-32	GCC 3.4.4 (nominal support only)

Mac OS X

Table 5. Mac OS and Supported Compilers

Operating System	Architecture	Compilers
Mac OS X 10.5	Native (PowerPC <u>or</u> x86-32)	Xcode 2.5, 3.0, 3.1.2
Darwin 8.x, 9.x	Native (PowerPC <u>or</u> x86-32), Universal (PowerPC <u>and</u> x86-32)	GCC 4.0.1

Added Platforms

Table 6. Added Platforms

Operating System	Architecture	Compilers
Linux-2.6.x	x86-32, x86-64	GCC 4.3.3, ICC 10.1
FreeBSD-6.1	x86-32	GCC 3.4.6
MS Windows	x86-32, x86-64	MS Visual Studio 2008 (C++ 9.0)

Discontinued Platforms

Table 7. Discontinued Platforms

Operating System	Architecture	Compilers
Solaris 10	SPARC, x86-32	Sun Studio 8 (C++ 5.5)
Solaris 10	x86-64	Sun Studio 11 (C++ 5.8)
FreeBSD-6.1	x86-32	GCC 3.4.4
MS Windows	x86-32	MS Visual Studio .NET 2003 (C++ 7.1)
Mac OS X 10.x	Native	Xcode 1.0

Caveats and Hints

GCC 3.0.4

- Destructor of constructed class member is not called when exception is thrown from a method called from class constructor body (fixed in 3.3).
- STL stream uses locale in thread-unsafe way which may result in segmentation fault when run in multithread mode (fixed in 3.3).
- Long-file support for C++ streams is disabled/broken (first broken in 3.0; fixed in 3.4).

GCC 3.4.x, 4.0.x

- At least on Linux, `ifstream::readsome()` does not always work for large files, as it calls an `ioctl` that doesn't work properly for large files (we didn't test whether 4.0.x fixed this).
- GCC 3.4.4 has a bug in the C++ stream library that affects some parts of our code, notably CGI framework (fixed in 4.0.1).

ICC 8.0

- ICC 8.0 lacks large file support for C++ streams on 32-bit Linux (fixed in 10.1).

Last Updated

This section last updated on June 22, 2009.