The NCBI C++ Toolkit

Release Notes (June, 2010)

Created: June 15, 2010.
Last Update: June 29, 2010.

    — GCC 3.4.x

Download

Download the source code archives at:

ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/2010/Jun_15_2010/

- ncbi_cxx— Jun_15_2010.tar.gz — for UNIX'es (see the list of UNIX flavors below) and MacOSX
- ncbi_cxx— Jun_15_2010.exe — for MS-Windows (32- and 64-bit) / MSVC++ (8.0, 9.0) — self-extracting
- ncbi_cxx— Jun_15_2010.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 April 23, 2010.

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]
BerkeleyDB	4.4.20, 4.6.21.1	4.6.21.1	4.6.21.1	4.5.20	4.6.21.1	4.5.20	4.5.20.NC [b]
Boost Test	1.35.0	1.35.0	1.41.0	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	-
libbzip2	current	current	current	current	current	current	1.0.2 [b]
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 [c]	current	current	current	current	current	current	4.1.3
LZO	-	2.02	2.02	2.02	2.02	2.02	2.02 [b]

PCRE							7.9 [b]
SQLite3	3.6.2	3.3.5	3.3.5	-	-	-	3.6.14.2 [b]
Sybase	-	12.5.0.6-ESD13	12.5.0.6-ESD13	-	12.5.1	12.0-EBF209	12.5
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.

[c] Only 4.1.3 version is supported currently.

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]
BerkeleyDB		4.6.21.NC [b]
Boost Test		1.42.0 [b]
libbzip2		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]
PCRE		7.9
SQLite3		3.6.14.2
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

CORELIB

- CInterProcessLock -- new class for implementing simple inter-process lock
- CDirEntry::DereferencePath() -- new method to dereference all symbolic links in the path
- CTimeout -- new class to store timeout interval
- CObjPool -- Template for universal object pool (CObjPool<>) was added
- New NStr::GetField[_Unsafe] function -- searches for a field in the given string considering the given delimiters. May merge delimiters.
- New NStr::SQLEncode function implementation -- implemented: provides the encoded string with added outer single quotes.
- CNcbiRegistry -- reworked once more, with generic functionality factored out into a new CCompoundRWRegistry class; now allows automatic loading of subregistries listed in [NCBI].Inherits, 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.
- Added CTempStringEx for storing C-style zero ended strings.
- CFile -- added comparison of text files CFile::CompareTextContents
- CStringUTF8 -- when converting the UTF8 string to single byte representation, added option to append specific characters instead of throwing an exception
- CTime -- added time conversion from/to "struct tm"

UTILITIES

- New g_FloatingPoint_Compare function added: Floating point values comparisons in accordance with the required tolerance type and tolerance value.
- New g_GetPasswordFromConsole function added: gets a password without echoing the user input
- ILineReader -- new GetLineNumber method in addition to (byte-based) GetPosition.

XML

- XmlWrapp – a NCBI clone (fixed, extended and built into the C++ Toolkit tree) of a generic XML handling C++ library (built on top of libxml2/libxslt). See XmlWrapp (XML parsing and handling, XSLT, XPath).

SERIALIZATION

- Added possibility to read data in JSON format - CObjectIStreamJson class
- When writing data in JSON format, added several forms of serialization of binary data - OCTET STRING, hexBinary and Base64Binary

DATATOOL

- Added possibility to generate SOAP client code based on a WSDL specification. NCBI C++ Toolkit provides support of version 1.1 of SOAP specification over HTTP transport protocol, and uses document binding style with literal schema definition.

- When generating C++ code by a data specification, added option to generate sample DEF file to make it easier to create a custom one.

CGI

- CCgiUserAgent -- updated list of browsers and bots; fixed some corner cases; added new method IsMobileDevice() to detect browsers runs on mobile devices

DATABASE ACCESS

- Added SDBAPI -- new "simple" API to work with SQL servers
- Discontinued DBAPI drivers: ftds8, ftds_dblib, fdts_odbc, odbcw, msdblib and sqlite
- Added a wrapper to work directly with SQLite through SQLite API (library sqlitewrapp).
- Fixed classes CDB_Char and CDB_LongChar to reflect server-side logic, so now they pad values with spaces up to the specified size.

BIO-OBJECTS

- Implemented CFeatTree class to order features in a dependence tree.
- CSeqId::GetLabel -- adjust formatting of general, patent, and PDB IDs per the C Toolkit's historical behavior.
- CSeq_id::IdentifyAccession -- recognize more prefixes (AXXXX, FT-FV, GP-GZ,HA-HL) and (mixed-in) EMBL TPA proten accessions; recognize that "general" IDs within the SNP, ti, TR_ASSM_CH, TRACE_ASSM, and TRACE_CHGR databases correspond to nucleotides; recategorize GSS accessions as not necessarily genomic.
- CProt_ref -- support looking up EC numbers' status and replacements (if any).
- CSeqFeatData -- new feature subtype clone.
- Added optional feature comparator for ordering very similar features.
- Added ISeq_feat interface for common functionality of CSeq_feat, CSeq_feat_Handle, and CMappedFeat.
- Added overloaded methods CSeq_id_Handle::GetHandle(string or gi).
- Feature ids now can be not only integers but also strings.
- Seq-table can now contain other fields than those in Seq-feat.
- Added lookup functions Seq-feat -> CSeq_feat_Handle, and CSeq_feat_Handle -> CMappedFeat.
- Added CAnnot_CI(CSeq_entry_Handle).
- Added getters for new fields Seq-feat.ids and Seq-feat.exts to the CSeq_feat_Handle.
- Implemented FeatureFetchPolicy for gathering features on a segmented sequence.
- CMappedFeat implements faster getters for location/product Seq-id/range.
- Annot track ZoomLevel is appended to annot name with @@ delimiter.
- Implemented CScope::ResetDataAndHistory().
- Added CScope::RemoveBioseq() and CScope::RemoveAnnot().
- CSeq_id_Handle ordering puts GIs before other ids.
- CSeq_loc::GetTotalRange() now understands bond and equiv locations.
- CSeq_id_Handle can store some textseqid and general ids in packed form.

LOCAL DATA STORAGE

- Implemented indexing standalone Seq-annot, Seq-align, and Seq-align-set.
- Resolve LDS Seq-id conflict to a file indexed latest.

ALGORITHMS

- COBALT -- Added standard multiple sequence alignment formats: FASTA plus gaps, ClustalW, Phylip, Nexus, as well as in ASN.1.
- COBALT -- Changed default values for clustering related parameters.

BLAST

- Added blast_formatter and blastdbcheck applications to BLAST+ package
- Formatter -- split the Blast formatter library into three separate libraries, to get rid of unnecessary dependencies and have a Blast-independent alignment formatting library
- Options -- added a deep copy method to the Blast options class that copies all the nested structures
- Command-line Applications -- added an "Entrez query" command line argument to the BLAST applications, added -list, -list_outfmt, and -show_blastdb_search_path options to blastdbcmd
- Added library for real time computation of Gumbel parameters for sequence alignment using arbitrary scoring system.
- Added support for blastdbcmd range input to be in 1-based offsets (not backwards compatible)
- Added support for md5 checksum validation with update_blastdb.pl

GENBANK DATA RETRIEVAL

- Implemented configurable connect timeout and wait time.
- Detect connections closed by inactivity timeout.
- Increased stability of GenBank data loader on faulty connections.

OBJTOOLS

- Added optional parameter to feature table reader to collect errors from parsing. If no error container provided, errors would be thrown as exceptions.
- Added data loader for SRA files
- Added another method to reader that accepts ILineReader.

BIO-TOOLS

- CFastaOstream -- new flags fMapMasksUp, fMapMasksDown, and fNoExpensiveOps.
- CFastaReader -- improve error reporting and defline parsing; new flags fLaxGuess and fStrictGuess.
- feature::**GetLabel** -- new flag-based variant; optionally ignore comments.
- sequence::**GetTitle** -- new flag fGetTitle_NoExpensive.

BUILD FRAMEWORK

PROJECT TREE BUILDER

- Added Java-based GUI tool that allows on-the-fly customization of configuration parameters.
- Added possibility to assign 'tags' to projects, and filter generated solution by them.
- Enhanced support of Xcode build system
- Enhanced flat makefile generated on Unix to optimize compilation
- Added possibility to specify target platform enhanced library dependencies analysis

CONFIGURE/MAKE

- Various changes to support for optional third-party libraries; in particular, --with-package=* always has priority over any PACKAGE_PATH setting, and explicitly requested packages must be available.
- Configure and frontends (compilers/unix/*.sh) -- noteworthy new flags --with-configure-dialog, --with-saved-settings=..., --with-tcheck.
- Large-file support enabled by default when building with ICC 9 or newer.
- Support for building with GCC 4.2 on Mac OS X 10.6.x (aka "Snow Leopard").
- Support for building with GCC 4.4.x, particularly on Linux. (It should also work on other supported OSes, but that remains to be verified.)
- Limited support for building with Sun Studio 12 Update 1 (WorkShop 5.10) on Solaris. (Some applications wind up too large to link in debug builds.)
- Preliminary support for building with ICC 11.x on Linux.

APPLICATIONS

- asn2asn -- added options for merging external annotations.
- wig2table -- new simplified and fast application for parsing WIG data and writing it as ASN.1.
- agpconvert -- added two new options to set Haplotype (-ht <String>) and Sex/gender (-sex <String>) in Biosource.subtype
- id1_fetch -- added options: -maxplex, -timeout and -extfeat.
- agp_validate -- minor bug fixes (improve recognizing of nucleotide accessions, allow more input files)
- objmgr_demo -- added options -get_ids, -get_gi, -get_acc, -check_cds, -minus_strand, -check_seq_data, -scan_whole_sequence, -seq_vector_tse, -get_feat_handle, -no-feat-policy, -make_tree, -print_tree.
- objmgr_demo -- added optional data loader from SRA database.

GRID (Distributed Computing)

- NetCache server -- rewritten to use SQLite as back storage system (instead of Berkeley DB) and to add new capabilities
- Removed classes CNetCacheClient and CNetCacheClient_LB.
- Implemented blob storing via netcache_control.
- Worker node APIs -- new method CGridThreadContext::ReturnJob() to explicitly return the job to NetSchedule.

- New worker node behavior -- when a job is not explicitly marked as Done or Failed, it's not returned to the server (use [server]/allow_implicit_job_return to revert to old behavoir).
- User clean-up procedures for worker nodes via interfaces IWorkerNodeCleanupEventListener and IWorkerNodeCleanupEventSource.
- CNetCacheAPI/CNetScheduleAPI/CNetICacheClient constructors can use a user-supplied CConfig object, will search the application registry for the missing parameters.
- New feature of ns_remote_job_control -- bulk dump of output.
- New netschedule_control 'read_fail', 'cancel', and 'getconf' actions.
- New netcache_cotrol parameter 'auth'.
- New CNetCacheAPI method to retrieve blob size.
- New CNetCacheAdmin method to print server health information.
- New test application test_grid_worker, which mimics a worker node.
- New worker node configuration parameter [server]/force_exit -- exit immediately without waiting for the worker threads to finish.
- The remote_cgi worker node now passes NetSchedule job ID via the NCBI_NS_JID environment variable.
- The netbvstore server and its client API were removed.
- Commands -log and -dropstat were removed from the netcache_control utility.

MISCELLANEOUS

- Easy-to-build bundle of 3rd-party packages (in source codes) for MSVC++ has been prepared and put to a public FTP along with the NCBI C++ Toolkit sources.
- Integrated a new version of BitMagic C++ library version 3.6.1
- PCRE -- upgraded internal PCRE to version 7.9.

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 mainstream version is shown in bold.

UNIX

Table 3. UNIX OS's and Supported Compilers

Operating System	Architecture	Compilers
SuSE-9.x (LIBC 2.3.5)	x86-32	GCC **3.4.2**, 4.1.2 [a], 4.2.3 [a], 4.3.3, 4.4.2 ICC **10.1,** 11.1[a]
SuSE-9.x (LIBC 2.3.5)	x86-64	GCC **4.0.1**, 4.1.2, 4.2.3 [b], 4.3.3, 4.4.2 ICC **10.1,** 11.1[a]
CentOS 5.4 (LIBC 2.5)	x86-64	**GCC 4.4.2**
Ubuntu 9.04 ("jaunty") (LIBC 2.9)	x86-32	**GCC 4.3.3**
Solaris 10	SPARC	GCC 4.1.1 [c] Sun Studio 12 (C++ 5.9), Update 1[a]
Solaris 10	x86-32	GCC 4.2.3 Sun Studio 12 (C++ 5.9), Update 1[a]
Solaris 10	x86-64	Sun Studio 12 (C++ 5.9), Update 1[a]
FreeBSD-6.1	x86-32	GCC 3.4.6
FreeBSD-8.0	x86-32	GCC 4.2.1
Darwin 8.x, 9.x	Native, Universal	GCC 4.0.1
Darwin 10.5x (Mac OS X 10.6.x)	Native, Universal	GCC 4.0.1, GCC 4.2.1

[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) NOTE: We also ship an easily buildable archive of 3rd-party packages (including NCBI C Toolkit) for this platform, although not all x86-32 libraries are available for x86-64 architecture.
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, MacOS x 10.6	Native (PowerPC or x86-32 or x86-64)	Xcode 3.0, 3.1.2, 3.2.2
Darwin 8.x, 9.x, 10.x	Native (PowerPC or x86-32 or x86-64), Universal (PowerPC and x86-32)	GCC 4.0.1 GCC 4.2.1 (only available under Darwin 10.x)

Added Platforms

Table 6. Added Platforms

Operating System	Architecture	Compilers
SuSE-9.x (LIBC 2.3.5)	x86-32	GCC 4.4.2
SuSE-9.x (LIBC 2.3.5)	x86-64	GCC 4.4.2
CentOS 5.4 LIBC 2.5)	x86-64	**GCC 4.4.2**
Ubuntu 9.04 ("jaunty") (LIBC 2.9)	x86-32	**GCC 4.3.3**
Solaris 10	SPARC	Sun Studio 12 Update 1[a]
Solaris 10	x86-32	Sun Studio 12 Update 1[a]
Solaris 10	x86-64	Sun Studio 12 Update 1[a]
FreeBSD-8.0	x86-32	GCC 4.2.1
Darwin 10.5x (Mac OS X 10.6.x)	Native, Universal	GCC 4.0.1, GCC 4.2.1
MacOS x 10.6	Native (PowerPC or x86-32 or x86-64)	Xcode 3.2.2
Darwin 8.x, 9.x, 10.x	Native (PowerPC or x86-32 or x86-64), Universal (PowerPC and x86-32)	GCC 4.2.1 (only available under Darwin 10.x)

[a] some support

Discontinued Platforms

Table 7. Discontinued Platforms

Operating System	Architecture	Compilers
SuSE-9.x (LIBC 2.3.5)	x86-32	GCC 3.0.4, ICC 8.0
SuSE-9.x (LIBC 2.3.5)	x86-64	ICC 9.1

Mac OS X 10.5	Native (PowerPC or x86-32)	Xcode 2.5

Caveats and Hints

GCC 3.4.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).

Last Updated

This section last updated on June 29, 2010.