The NCBI C++ Toolkit

Release Notes (Version 12, May 2013)

Created: June 18, 2013.

Last Update: June 20, 2013.

Download

Download the source code archives at: ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/ARCHIVE/12_0_0/

- ncbi_cxx--12_0_0.tar.gz — for UNIX'es (see the list of UNIX flavors below) and MacOSX
- ncbi_cxx--12_0_0.exe — for MS-Windows (32- and 64-bit) / MSVC++ 10.0 — self-extracting
- ncbi_cxx--12_0_0.zip — for MS-Windows (32- and 64-bit) / MSVC++ 10.0

The sources correspond to the NCBI production tree sources, which are originally based on the development tree source snapshot from March 11, 2013 but also include many hundreds of important and safe code updates made since then and through May 17, 2013 (and then some).

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.

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, 4.6.21.1, 4.7.25, 4.6.21.NC
Boost Test	1.35.0 or newer	1.40.0.1, 1.42.0, 1.45.0
FastCGI	All versions	2.1, 2.4.0
libbzip2	All versions	1.0.2, 1.0.5
libjpeg	All versions	6b, 8.0
libpng	All versions	1.2.26, 1.2.7, 1.5.13
libtiff	All versions	3.6.1, 3.9.2, 4.0.0
libungif	All versions	4.1.3 (libungif), 4.1.6 (giflib)
libxml2	All versions	2 2.7.3, 2.7.6, 2.7.8,
libxslt	1.1.14	1.1.24, 1.1.26
LZO	2.x	2.05
PCRE	All versions	7.8, 7.9, 8.32,
SQLite3	3.6.6 or newer	3.6.12, 3.6.14.2, 3.6.22, 3.7.13
Sybase	All versions	12.5
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 3rd party packages themselves. The release's package list includes links to download sites. However, the user still needs a list of the 3^{rd} 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
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

[a] Applies to MSVC 9, MSVC 10

Build

For guidelines to configure, build and install the Toolkit see here.

New Developments

HIGHLIGHTS

Major advances, additions to the BAM, SRA, cSRA, WGS, VDB data loaders of the Bio-Sequence Object Manager

FreeTDS driver -- Support Kerberos authentication.

Redesigned Unicode support (stage 1) - added new CUtf8 class which will handle UTF8 conversions and replace CStringUTF8, prohibited implicit single byte character string conversions.
Significant additions and improvements in the XML and JSON serialization APIs.

Cleaned up the code (again) from non-MT-safe static objects.

CORELIB

New functionality:

- Added possibility of having several argument description (CArgDescription) objects in a program; proper description is chosen based on the value of the very first command line argument - "command", the rest of the arguments is then parsed according to the chosen description. Such command descriptions can be combined into command groups.

- Added platform-independent error reporting mechanism, similar to errno or SetLastError, - CNcbiError. When a Toolkit core API function fails, it reports aditional information there.
- Redesigned Unicode support - added new CUtf8 class which will handle UTF8 conversions and replace CStringUTF8, prohibited implicit single byte character string conversions.
- Added CException manipulators for severity and console output.
- NStr:: -- improved errno handling, dropped support for fIgnoreErrno flag.
- NStr:: -- addednew methods CommonPrefixSize(), CommonSuffixSize(), CommonOverlapSize().
- NStr::StringToNumeric() -- renamed to StringToNonNegativeInt().
- Nstr::ParseEscapes() -- added options to parse out-of-range escape sequences.
- NStr::CEncode() -- rewrite to produce use double-quoted strings by default, and added counterpart method CParse() to decode a "C" strings.
- CTime -- added GetCurrentTimeT() to get current GMT time with nanoseconds.
- CSignal -- added method ClearSignals().
- CDirEntry -- add permission/mode <-> string conversion methods.
- CDirEntry -- added methods: GetUmask(), SetUmask, ModeFromModeT().
- SetCpuTimeLimit() -- added new declaration and deprecated old one, re-enabled user print handler for SIGXCPU.
- SetMemoryLimit[Soft|Hard]() -- new methods to allow separately specify soft and hard memory limits for application.
- Added string literals as well as directory pathes to CExprParser
- CNCBIRegistry (and other registries) is able to work with configuration data not belonging to any section, when created with fSectionlessEntries
- CExprParser is able to accept logical literals starting with a number in fLogicalOnly mode

Improvements:

- In-heap CObject detection via TLS variable.
- Inline internal method CObject::InitCounter() for speed.
- CTempStringEx -- Optionally own data.
- NStr -- Split, Tokenize, etc. now accept flags, controlling not only delimiter merging but also whether to treat multi-character delimiters as patterns (generalizing TokenizePattern) and to treat any subset of \'" as special.

DATA SERIALIZATION

New functionality:

- Added support for mandatory elements with default in XML serialization.
- Added possibility of using NCBI_PARAM mechanism for data verification and
- skipping unknown members settings.

- Added possibility of skipping unknown data in JSON input; added JSONP output mode.

Improvements:

- Optimization of deserialization methods (mostly binary ASN.1).

Bug fixes:

- Avoid double closing tag when skipping enums in XML.
- Store serialization format flags for correct delayed parsing.

XMLWrapp:

- Safe dereferencing node iterators
- xml::nodes_view is not supported anymore
- A few memory leaks are fixed
- exslt auto registration if available
- XSLT extension functions support added
- XSLT extension elements support added
- run_xpath_expression(…) to handle boolean, number and string types as return values

DATATOOL

- Enhanced SOAP client code generation to support WSDL specification which contains several XML schemas - to handle elements with identical names in different namespaces.
- Added possibility of converting data to and from JSON format.

CGI

- CCgiUserAgent -- separate CCgiUserAgent into user_agent.[h|c]pp.
- CCgiUserAgent -- added methods: GetDeviceType(), IsPhoneDevice(), IsTabletDevice().
- Added flags to allow use external pattern lists on the parsing step to identify bots, phones, tablets and mobile devices. Return iPad back to the list of mobile devices. Interpret all Android based devices as mobile devices.
- CCgiUserAgent -- update list of browsers and mobile devices.

UTILITES

- Compression API -- allow concatenated files for eGZipFile mode by default.
- Compression API -- added support for "empty input data" compression via fAllowEmptyData flag, It will allow to compress zero-length input data and provide proper format header/footer in the output, if applicable. By default the Compression API not provide any output for zero-length input data.
- CRegexp -- changed GetSub()/GetMatch() methods to return CTempString instead of string.
- include/util/diff/dipp.hpp -- new DIFF API (CDiff, CDiffText).
- Added possibility to convert differently typed static array.
- Added limited_size_map<> for caching and garbage collection.

- Mask matching is rewritten with CTempString for efficiency.
- ILineReader -- Clarify API, introducing ReadLine and GetCurrentLine as synonyms of operator++ and operator* respectively.
- CTextJoiner -- New template for collecting and joining strings with a minimum of heap churn.

DBAPI

- Support Sybase ASE 15.5 servers.
- Python bindings -- Optionally release Python's global lock around blocking DBAPI operations; rework exception translation to follow PEP 249 (*).
- Added support for Kerberos authentication (copied from FreeTDS 0.91).

BIO-OBJECTS

New functionality:

- CDeflineGenerator -- Generally streamline; make expensive operations (currently just consulting related sequences' annotations) optional.
- CFastaOStream -- Add a gap-mode parameter, making it possible to represent gaps by runs of inline dashes or special >?N lines; optionally (but by default) check for duplicate sequence IDs; support processing an entire raw Seq-entry without even a temporary scope.
- CFastaReader -- Add two flags that can increase performance: fLeaveAsText skips reencoding in a (more compact) binary format, and fQuickIDCheck directs local ID validation to consider just the first character.
- CSeq_id -- Accept parse flags when parsing a single ID from a string; recognize WGS scaffolds, additional prefixes (F???, G???, HY, HZ, J??, JV-JZ, KA-KF, and WP_), 10-digit refseq_wgs_nuc accessions (notably for spruce), and more TPE protein accessions (still interspersed with EMBL's own accessions).
- Added CGeneFinder class for finding the genes of a feature using the flatfile generator's logic
- Seq_entry_CI can now optionally include the top seq-entry
- objects::CGC_Replicon now has accessors to return molecule type (‘Chromosome’, ‘Plasmid’, etc.) and location (‘Nuclear’, ‘Mitochondrion’, ‘Chloroplast’, etc.). You can also retrieve a label (GetMoleculeLabel()) which summarizes molecule type and location in one string.

BIO-TOOLS

New Development:

- Validator:
  - Added functions for validating and autocorrecting lat-lon, collection-date, and country BioSource SubSource modifiers. Synchronized validation with C Toolkit.
- Flat-file generator:
  - can now be set to show only certain blocks
  - optionally set callback for each item or bisoeq that's written which allows changing the text and specifying to skip that item or even halt flatfile generation.

  - support the /pseudogene qualifier
  - allow complex locations in transl_excepts. (a.k.a. code-breaks)
  - support "pcr" linkage-evidence
  - support for /altitude qualifier
  - Support "Assembly" in DBLINK
  - API for conversion between source-qualifier and feature-qualifier enums and strings
  - support assembly gap feature quals (e.g. /gap_type, /linkage_evidence, etc.)
- ASN.1 Cleanup:
  - Set pseudo to true if pseudogene is set
  - More places where it sorts and removes redundancies (example: sort and unique organism synonyms)
  - Remove duplicate pcr-primers
  - clean up altitude
  - fixing some genbank quals into real quals (example: gene-synonym)
- CFastaOstream:
  - Can optionally show [key=val] style mods in deflines
- CFeature_table_reader:
  - now supports more quals (example: centromere)
- CFastaReader:
  - optionally accumulate warnings in a vector instead of printing them to allow more flexible handling and more info to caller.
- AGP:
  - created CAgpToSeqEntry for converting an AGP file into a Seq-entry.

COBALT

Bug fixes:

- Incorrect alignments with sequence clustering

BIO-OBJECT MANAGER

New functionality:

- Added fast CScope methods for getting some sequence information without loading the whole entry - length, type, taxonomy id, GI, accession, label.
- Added processing of Seq-table column "disabled".
- Added FeatId manipulation methods.
- Added feature::ReassignFeatureIds().
- Added CSeq_table_CI with location mapping.
- Added CSeqVector_CI::GetGapSeq_literal().
- Added recursive mode and seq-entry type filtering to CSeq_entry_CI.

Improvements:

- Allow non-scope bioseq lookup in CSeq_Map (for segset entries).

- Allow post-load modification of sequences.
- Optimization of ContainsBioseq() for split entries.
- Added CTSE_Info::GetDescription() for better diagnostics.
- More detailed error message in annots.
- Allow iteration over non-set entries in CSeq_entry_CI - treat them as empty sets.

Bug fixes:

- Fixed generation of Seq-table features.
- Fixed loading of various Seq-id info from multiple data loaders.
- Made bulk and single requests to return the same results.
- Fixed unexpected CBlobStateException for non-existent sequences.
- Avoid deadlock when updating split annot index.
- Fixed recursive iteration in CSeq_entry_CI if sub-entry
- doesn't have matching entries.
- Fixed mixup of feature ids and xrefs.
- Fixed fetching by feat id/xref from split entries.
- Fixed in-TSE sequence lookup via matching Seq-id.
- Fixed matching Seq-id lookup with multiple candidates.
- CSeqMap_CI::GetRefData() should work for gaps too.
- Exclude removed features from un-indexed search.

OBJECT LIBRARIES

New functionality:

- Implemeted multi-id Seq-loc comparison.

GENBANK DATA LOADER

Bug fixes:

- Allow withdrawn/suppressed entries with non-default credentials.
- Preserve blob state if Seq-entry skeleton is attached to split info.
- Remember blob state from get-blob-ids reply too.
- Detect non-existent Seq-id when loading blob-ids.
- Release connection as soon as possible to avoid deadlock.
- Lock split TSE only after receiving split info.

BAM DATA LOADER

New functionality:

- Implemented pileup graphs for BAM loader.

Improvements:

- Generate simple ID2 split info to postpone record loading.

SRA DATA LOADER

New functionality:

- Added option to clip SRA sequences.

cSRA DATA LOADER

New functionality:

- Implemented CCSraShortReadIterator.
- Added short read info into Seq-align.ext.
- Added pileup graph param setter and getter.
- Added support for SECONDARY_ALIGNMENT.
- Use gnl|SRA|<acc>.<spot>.<read> for short read ids.
- Added lookup for short reads by SPOT_ID and READ_ID.
- Allow optional VDB columns.
- Added clippig by quality.
- Added option to exclude cSRA file path from short read ids.

Improvements:

- Allow cSRA reader to open old SRA tables.
- Reduced number of TSE chunks.
- Removed obsolete config parameters: INT_LOCAL_IDS, SEPARATE_LOCAL_IDS.
- Removed empty VDB table, cursor, and column constructors.
- Generate simple split info to postpone cSRA record loading.
- Exclude technical reads.
- Check VDB column data type to detect incompatible VDB files.
- Place short reads in a separate blob.
- Added lookup from short read to refseq.
- Added mapping align on short read.
- Added secondary alignment indicator.
- Added centralized MT-safe VDB cursor cache.
- Allow ERR accessions in cSRA loader.
- Switched to new SRA SDK accession resolution scheme.
- Use SRA SDK configuration mechanism.
- Added SRA file cache garbage collector.
- Accept multiple ids in reference sequences.
- Reduce number of reads per blob to 1 for speed.
- Allow cSRA data to have no REFERENCE table.
- Increased limit on allowed number of short reads per spot.
- Increased flexibility on existing VDB columns.
- Try to resolve remote VDB files too.
- Use GC for loaded entries.

- Indicate that cSRA loader can load data by blob id.
- Set max value of quality graph properly.

Bug fixes:

- Fixed MISMATCH generation for I segments.
- Added missing RegisterInObjectManager().

WGS DATA LOADER

New functionality:

- Implemented VDB WGS reader and data loader.

VDB DATA LOADER

New functionality:

- Implemented VDB graph reader and data loader.

BLAST

New functionality:

- Added new API to return blast preliminary stage result as a list of CStd_seg
- Added new tabular features for blast which includes taxonomy information, strand sign and query coverage
- Added new features for blastdbcmd batch sequence retrieval which allow user to specify strand sign and sequence range
- Added new functionality in makeprofiledb to produce database that supports composition based statistics
- For more details, see BLAST+ 2.2.27 and 2.2.28 release notes (http://www.ncbi.nlm.nih.gov/books/NBK131777/)

Bug fix

- Fix ASN 1 input for makeblastdb

APPLICATIONS

- convert_seq -- Allow for more efficient operation in some cases, mostly by bypassing object manager overhead; implement a new "IDs" input format; have non-zero inflags for ASN.1 or XML request sequence data repacking.
- multireader -- Added AGP.
- blastn's -- Changed default value - use_index to false
- vecscreen -- Added command line application
- rmblastn -- Added command line application
- asn2asn -- added ability to read and write Seq-submits

BUILD FRAMEWORK (UNIX)

- configure and frontends (compilers/unix/*.sh) -- Don't override explicitly specified optimization flags with default FAST settings (but do still apply custom FAST settings if also specified).

- compilers/unix/Clang.sh, .../LLVM-GCC.sh -- New frontends for configure to simplify compiler selection.
- new_project.sh -- Improve support for projects involving libraries.

CHANGES TO COMPILER SUPPORT

Linux ICC support extends up to version 13.

Mac OS X support extends to version 10.8.x, with Clang, FSF GCC, or LLVM GCC (also via Xcode).

Solaris support extends to version 11, with GCC or WorkShop (as with older OS versions).

Documentation

Location

The documentation is available online as a searchable book "The NCBI C++ Toolkit": http://www.ncbi.nlm.nih.gov/toolkit/doc/book/.

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.

Public assess to our SVN trunk:

- 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.x (LIBC 2.5)	x86-64	**GCC 4.4.2,** 4.0.1[a], 4.1.2[a], 4.3.3[a], 4.6.0[a], 4.6.3[a] ,GCC 4.7.2 [a]
CentOS 5.x (LIBC 2.5)	x86-32	**GCC** 4.4.5 [a]**, 4.6.0**
CentOS 6.x (LIBC 2.12)	x86-64	**GCC 4.4.2**, 4.6.3 [a], 4.7.2 [a], 4.8.0 [a]
Ubuntu 9.04 ("jaunty") (LIBC 2.9)	x86-32 x86-64	**GCC 4.3.3**
Solaris 10, 11[a]	SPARC	GCC 4.1.1[b], 4.5.3[b] **Sun Studio 12 (C++ 5.9)**, Sun Studio 12 Update 1 (C++ 5.10)[a] Oracle Studio 12.2 (C++ 5.11)[a]
Solaris 10, 11[a]	x86-32	GCC 4.2.3 **Sun Studio 12 (C++ 5.9)**, Sun Studio 12 Update 1 (C++ 5.10)[a] Oracle Studio 12.2 (C++ 5.11)[a]
Solaris 10, 11[a]	x86-64	**Sun Studio 12 (C++ 5.9)**, Sun Studio 12 Update 1 (C++ 5.10)[a] Oracle Studio 12.2 (C++ 5.11)[a]
FreeBSD-8.3	x86-32	GCC 4.2.2

[a] some support

[b] 32-bit only

MS Windows

Table 4. MS Windows and Supported Compilers

Operating System	**Architecture**	**Compilers**
MS Windows	x86-32	MS Visual C++ 2010 (C++ 10.0) NOTE: We also ship an easily buildable archive of 3rd-party packages for this platform.
MS Windows	x86-64	MS Visual C++ 2010 (C++ 10.0) NOTE: We also ship an easily buildable archive of 3rd-party packages for this platform
Cygwin 1.7.9	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 10.6 Mac OS X 10.8	Native (PowerPC or x86-32 or x86-64)	Xcode 3.0 - 3.2.6
Darwin 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) LLVM Clang 3.0

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

Darwin 10.x = Mac OS 10.6.x

Darwin 12.x = Mac OS 10.8.x

Added Platforms

Table 6. Added Platforms

Operating System	Architecture	Compilers
CentOS 5.x (LIBC 2.5)	x86-32	GCC 4.4.5 [a], 4.6.0
CentOS 5.x	x86-64	GCC 4.7.2 [a]
CentOS 6.x (LIBC 2.12)	x86-64	GCC 4.4.2 , 4.6.3 [a], 4.7.2 [a], 4.8.0 [a]
Mac OS X 10.5, MacOS x 10.6,	Native (PowerPC or x86-32 or x86-64)	Xcode 3.2.3 - 3.2.6 LLVM Clang 3.0

[a] some support

Discontinued Platforms

Table 7. Discontinued Platforms

Operating System	Architecture	Compilers
MS Windows	x86-32, 64	MS Visual C++ 2008 (C++ 9.0)
Mac OS X 10.4.x(Darwin 8.x), Mac OS X 10.5.x(Darwin 9.x)	Native (PowerPC or x86-32 or x86-64), Universal (PowerPC and x86-32)	GCC 4.0.1, Clang 3.0
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 July 1, 2013.