The NCBI C++ Toolkit Book
The NCBI C++ Toolkit Book

The NCBI C++ Toolkit

Release Notes (July 8, 2004)

Release Notes (July 8, 2004)

Download Location

ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/2004/Jul_08_2004/

Source Archive Contents

Source Code Archives

- ncbi_cxx_unix--Jul_08_2004.tar.gz-- for UNIX'es (see the list of UNIX flavors below) and MacOSX/GCC
- ncbi_cxx_unix--Jul_08_2004.gtar.gz-- for UNIX'es (see the list of UNIX flavors below) and MacOSX/GCC
- ncbi_cxx_win--Jul_08_2004.exe-- for MS-Windows: MSVC++ 6.0 and 7.1 (self-extracting)
- ncbi_cxx_win--Jul_08_2004.zip-- for MS-Windows: MSVC++ 6.0 and 7.1
- ncbi_cxx_win--Jul_08_2004.tar.gz-- for MS-Windows: MSVC++ 6.0 and 7.1
- ncbi_cxx_mac_cw--Jul_08_2004.tar.gz-- for MacOS 10.3.4 / CodeWarrior DevStudio for MacOS 9.2
- ncbi_cxx_mac_gcc--Jul_08_2004.tar.gz-- for MacOS 10.2, 10.3.4 / GCC 3.3

The sources correspond to the NCBI production tree sources from patch "GCC34_MSVC7," which in turn corresponds to the development tree sources from around end of may, 2004.

New Development

Some of the new development may introduce potentially backward-incompatible changes.

CORELIB -- Portability and Application Framework

1. CRef<>-- added methods for atomic release and reset.
2. EDiagPostFlag-- added optional conversion of multi-line diagnostic messages into single-line ones.
3. CDirEntry-- the methods that support determining entries' types now let the caller specify how to treat non-dangling symlinks.
4. CFile::ETmpFileCreationMode-- to control whether to actually create the temporary file (to get rid of a possible race condition).
5. CDir-- new methodGetTmpDir()to get path to the temporary directory; old methodGetEntries()now has an additional parameter to ignore self recursive directory entries (".", "..").
6. The genericFindFilealgorithm -- now can do recursive searches.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

7. CTime-- new formatting symbol 'z' (local time in format GMT{+|-}HHMM). Also added output timezone parameter toCTime::AsString().
8. CProcess-- addedWaitForAlive()andWaitForTerminate()methods.
9. CPIDGuard-- now does smart reference counting in the PID guard file to avoid confusion with multiple locks; also addedCPIDGuard::Remove()method to force the removal of PID guard file.
10. CPluginManager-- new methods to better control DLL resolution:FreezeResolution ()andDetachResolver().

CONNECT -- Data streaming, Networking, and Dispatching

1. CNamedPipe-- replaced static const with enum for the default Toolkit and system pipe buffer size.May introduce potentially backward-incompatible changes.
2. CONN_ReadLine()-- new function, to read from CONN line-by-line.

HTML -- HTML Generation Library

CHTMLPopupMenu-- added support for Sergey Kurdin's popup menu with configurations, extended the API to account for the new menu features and functionality.

UTIL -- Miscellaneous Low-Level APIs

1. "Bitset" API -- the NCBI adaptation of the BitMagic template library to manipulate with compressed bitsets (by BitMagic author).
2. Added UNICODE/UTF8 conversion functions.
3. CResourcePool- added access to the list of available objects.
4. CResourcePool- added access to the list of available objects.
5. C[RW]Stream[buf]-- added optional ownership of the underlying IReader/IWriter-derived objects to allow for their automagic destruction.

UTIL/COMPRESSION -- Data Compression (GZIP, BZ2)

1. CZipDecompressor-- now can recognize and skip GZIP file format header.
2. CCompressionStream-- added optional ownership of the underlying C++ iostreams and/or compression processors to allow for their automatic destruction.
3. CCompression[IO]Stream-- new methods GetProcessedSize() and GetOutputSize() to get, respectively, the amount of the (de)compressed and original data processed so far.

SERIAL -- Data Serialization (ASN.1, XML)

1. Context-sensitive serialization hooks -- now, can be set for the given read context, using a pattern string, with wildcards allowed: "TypeName.Member1.Member2.HookedMember".
2. CObjectIStream::SetStreamOffset()-- to rollback from the current position of input stream, if possible.
3. Improved XML serialization, added "Windows-1252" encoding.
4. Implemented optional skipping of unknown data members in XML and ASN streams.
5. MSerial_{AsnText,AsnBinary,Xml}-- new I/O manipulators for regular C++ iostreams to simplify the serialization of serial data objects.
6. Implemented non-recursive copying and comparing of serial objects.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

DATATOOL -- Code Generator and Data Converter Tool

1. Added the type conversion check (when it is defined in DEF file) to prohibit inappropriate conversions.
2. Added optional generation of doxygen-style comments and inserting of pre-compiled headers when generating C++ code.

CGI -- CGI and Fast-CGI Application Framework

1. Turned off background exception reporting, by default.
2. Convert multi-line diagnostic messages into one-line ones, by default.
3. CCgiRedirectApplication-- use "&{var}" rather than "$var" to refer to the value of the variable; also, all new string values in the registry file must be URL-encoded now.May introduce potentially backward-incompatible changes.

Berkeley DB API (bdb) -- Much Enriched C++ API Based On BerkeleyDB

1. Query parser and interpreter -- implemented NOT and comparison operators.
2. Added support for Berkeley DB transactions.
3. Implemented transaction protected BLOB Stream.
4. ICache-- improved stability, added transaction protection, greatly reduced probability of file corruption.
5. Implemented transaction protected cursors.
6. CBDB_Env::JoinEnvimproved to support several program instances working with the same database.
7. CBDB_FieldLString-- new field type to support length prefixed strings (for better compatibility withstd::stringclass).
8. CBDB_Env-- added error logging (OpenErrFile).
9. BDB_find_field-- new search method usingCBoyerMooreMatchersearch condition.
10. CBDB_Cache::Open()-- added RAM cache size parameter.

DBAPI -- Generic SQL Database Connectivity

1. FreeTDS -- the bundled FreeTDS code now supports usinggetaddrinfo() andgetnameinfo(), enabling multithreaded builds on platforms that lackgethostbyname_r()et al.
2. IConnection-- new public methodsGetStatement(),GetCallableStatement (),GetCursor()andGetBulkInsert(). All objects obtained by calling these methods work within a single connection, opened byIConnection::Open()and cannot be used simultaneously. Don't mix them with the oldCreate()methods, otherwise an exception will be thrown.
3. IStatement-- new methodPurgeResults(), to purge all results of the given query.
4. IStatement-- new methodExecuteQuery(), to return the first resultset of the given query and to discard any additional results.

ALGO -- Advanced Algorithms Library

1. Tree specific algorithms and manipulation methods -- addedTreeFindCommonParent,TreeTraceToRoot,TreeReRoot,MoveSubnodes,CTreeMinimalSet, operations on node lists, etc.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

2 CBioTree- new, advanced tree container for phylogenetic and taxonomy trees.

ALGO/ALIGN -- Generic Alignment Algorithms

1 XALGOSPLIGN -- new library to share the state-of-the-art transcript alignment functionality.

2 CSplign-- provides access to all parameters of the XALGOSPLIGN algorithm and controls the algorithm's execution. To useCSpligndirectly, BLAST hits must be computed externally and a sequence accessor class must be provided (derived from classCSplignSeqAccessor).

3 CSplignSimple-- a lightweight version ofCSplign, which needs only twoCSeq_locand aCScopeobject to access sequences, and does BLAST computation internally.

4 CSplignFormatter-- class to format output produced byCSplignorCSplignSimpleas a tab-separated list of exons and gaps (unaligned cDNA regions) or as aCSeq_align_setobject.

BLAST

1 Introduction of remote BLAST API and demo program.

2 Added support for RPS-BLAST.

3 Merged TwoSequences and Database engines.

4 Complete rewrite of low-level gapped alignment functions.

5 Cleaned up and modularized API for saving HSPs, HSPLists and HitLists

6 Refactoring of link hsps code to make it more self-contained.

7 Addition ofCSeqDband multiple sequences implementations of the BlastSeqSrc API.

8 Introduction of the BlastDiagnostics API in CORE BLAST.

9 Fixed various naming conflicts with the NCBI C Toolkit..

10 Introduction of the BlastHspStream abstraction.

11 Introduction of PSSM calculation engine.

12 Unify handling of gapped Karlin blocks for protein and nucleotide searches.

13 Consolidated validation of all options structures in one function in CORE BLAST.

14 Added check for word size value in options validation for MegaBLAST: word size must be >=12 for contiguous MegaBLAST, and can be 11 or 12 for discontiguous MegaBLAST.

15 Added setting of a most reasonable database scanning stride value dependent on word size W and lookup table width L. The best stride is calculated as W - L + 1 or as W - L + 4 if the "variable word size" is set to true. However the resulting stride is rounded to a number divisible by 4 for all values except 6 and 7. This is done because scanning database with stride divisible by 4 does not require splitting bytes of compressed sequences. For values 6 and 7 however the advantage of a larger stride outweighs the disadvantage of splitting the bytes.

16 Started separation of preliminary and traceback stages in CORE BLAST.

17 Rewrote BlastKarlinLHtoK..

18 Port of RedoAlignmentCore from the NCBI C toolkit to Kappa_RedoAlignmentCore.

19 Provide an interface to retrive the matrices' frequency ratios.

20 Modified ungapped protein alignment code to handle PSSMs, optimizations to the construction of the protein lookup table.

ALNMGR -- Bio-sequence Alignment Manager

1 CAlnMap-- addedGetResidueIndexMap().

2 CAlnVec-- addedGetColumnVector()andCalculatePercentIdentity(); madeCScopea required parameter.

3 CAlnMix-- added forcing translation of nucleotide sequences; emadeCScopea required parameter.

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

1 Changed type of Seq-align containers from list<> to vector<> for better performance.

2 MovedCSeq_feat::CompareLocations()toCSeq_loc::Compare

3 AddedFindGi(),FindTextseq_id()andGetSeq_idByType()to simpify ID lookups in containers.

4 CSeq_id-- support several new prefixes inIdentifyAccession()method.

5 CSeq_id-- improved support for ranking IDs by utility.

6 New flat-file generator (`xformat` in "objects/format"). The previous generator (in "objects/flat") is no longer in active development.

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

1 CDesc_CI-- renamed toCSeq_descr_CIfor consistency with the ASN.1 class naming.May introduce potentially backward-incompatible changes.

2 CAnnotTypes_CI-- is no longer inherited from SAnnotSelector, but accepts it as an argument.May introduce potentially backward-incompatible changes.

3 SAnnotSelector-- renamed methods of from *Choice to *Type.May introduce potentially backward-incompatible changes.

4 CAlign_CI-- implemented alignment mapping.

5 CFeat_CI-- now, the annotations can be filtered by a combination (with both inclusions and exclusions) of types and subtypes.

6 CSeqdesc_CI-- now, more than one type can be used for the filtering.

7 CScope-- allow to fine-tune the getting of BioseqHandle depending on resolution flags.

8 CScope-- new method GetAllTSEs to get all already resolved TSEs.

9 CSeqVector-- added ncbi2na ambiguity randomizer.

10 Significantly improved performance of annotation collecting and mapping functions.

11 CPrefetchToken-- new, to provides background data pre-fetching.

12 CSeq_loc_Mapper-- new, to map locations and alignments.

13 CSeq_{feat,align,graph}_Handle-- handles for annotation types, to be used instead of "real" bio-objects.

14 OM++ headers -- reduced inter-dependency. Some code may require to include headers which were included implicitly before.May introduce potentially backward-incompatible changes.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

15 CBioseq_set_Handleand set ofC*_EditHandle's-- added for working with different bioseq objects (CBioseq,CBioseq_set,CSeq_entry, andCSeq_annot).

16 A start-up edit API for bioseq objects through various edit handles.ConvertSeqToSet ()andConvertSetToSeq()forCSeq_entry. Methods to move Bioseq withing Seq-entry. ImplementedMoveToandCopyTomethods for various handles.

17 CScope::GetSynonyms()-- do not try to resolve any Seq-ids, just assume that any unresolved Seq-id is a synonym. Seq-id conflict messages made clearer.

18 CBioseq_Handle-- new methodGetExactComplexityLevel().

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

1 CReader-- revamped to useCPluginManager.May introduce potentially backward-incompatible changes.

2 GenBank loader and its readers -- moved sources to new location (in OBJTOOLS). Libraries with the OM++ data loaders renamed to "libncbi_xloader_*". Libraries with the OM++ GenBank data loader's readers renamed to "libncbi_xreader_*". OBJMGR_LIBS macro updated correspondingly.May introduce potentially backward-incompatible changes.

3 GenBank data readers -- added support for TRACEs: Seq-id ::= general { db "ti", tag id NNN }.

4 CCachedId1Reader-- now supports newICacheinterface.

ID2

1 ID2 ASN.1 specification file was split into two parts. Now, the ID2 communication protocol is specified in "id2.asn" module, while the ID2 split data structure specifications are moved to "seqsplit.asn" module.

2 Moved ID2 and ID1 specific code out of object manager. Protocol is processed by corresponding readers. ID2 split parsing is processed by ncbi_xreader library - used by all GenBank readers.

3 Various parts of the ID2 communication protocol and the ID2 split data structures have been modified; these are still changing, albeit at much slower rate, and are expected to by and large stabilize by around mid-fall.

BIO-TOOLS

1 GFF/GTF files -- added [semi-experimental] support for reading these.

2 CObjectsSniffer -- bug fixes and perfomance optimizations.

LDS

Implemented sequence search using query (using BDB query).

BUILD FRAMEWORK

The Toolkit can now take advantage of preexisting installations of zlib, bz2lib, and libpcre, though it still includes copies of all three libraries in order to maintain support for systems that lack them. You may need to adjust some of your makefiles accordingly.May introduce potentially backward-incompatible changes.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

APPLICATIONS

1. ASN2FLAT and CONVERT_SEQ -- new applications to support conversion between various common representations of annotated biosequence data.
2. ID1_FETCH_SIMPLE -- added possibility to send any ASN.1 request to ID1 (for testing purposes).
3. SPLIGN -- improved quality of transcript alignments. Added a powerful compartment identification algorithm (implemented within the XALGOSPLIGN library) to identify pseudogenes and paralogs. The updated output format includes the compartment identifier preceded by strand sign.

Documentation

Document Location

The documentation is available online atas a book titled "The NCBI C++ Toolkit". This is an online searchable book.

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.

The older HTML documentation has been deprecated and is no longer being updated, and "The NCBI C++ Toolkit" online book at the previously listed URLs is the official documentation.

Document 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 new comer.

The DOXYGEN source browser is used to complement the traditional docs with structured DOXYGEN-generated "Library Reference" ones based on the in-source comments. You can access the DOXYGEN source browser for the NCBI code from:

http://www.ncbi.nlm.nih.gov/IEB/ToolBox/CPP_DOC/doxyhtml/

The above link is also available under the "Source Code Browsers" that appears on each page.

A C/C++ Symbol Search query has been added to each page of the online Toolkit documentation. You can use this to perform a symbol search on the public or in-house versions of LXR, Doxygen and Library. The Library search runs a CGI script that lists the library name in which a symbol occurs.

The current release notes as well as past release notes are now in an appendix in the C++ Toolkit Book and a link to the current release notes appears on each page of the online Toolkit document.

Building on the MacOS

We now build all the libraries and most of the applications including the Genome Workbench (gbench). We do not build any of the many test or demo apps (excepttestvalidator). We also build the FLTK library's GUI editor,fluid.

All apps are built as application bundles exceptgbench_plugin_scananddatatoolwhich are built as command line apps. Any of the applications can be built as command line apps by tweaking the build scripts or the Codewarrior projects.

Build procedure is still the same: use an AppleScript editor to open and run the script files makeLibs.met and makeApps.met. You must use a script editor capable of opening a script file larger than 32K, such as Apple's Script Editor v2.0, or Smile. Script Editor v1.9 will not work since makeLibs.met just got too big. The command line tool osascript also works.

Projects include targets to compile with BSD/Apple headers and libraries and with MSL headers and libraries, but plugins (for gbench) built with MSL do not link properly, and so, because of lack of interest to keep up with changes, some source code does not currently compile with MSL. Hopefully this will become easier to work with and get fixed with Codewarrior v.9.

Targets to be compiled can be controlled by including an empty file or folder in the compilers:mac_prj folder with the name 'Build' followed by the keywords of the targets you want built. The keywords are: MSL, BSD, Debug and Final. For example, to build only the BSD Debug targets use: "Build BSD Debug", to build both BSD debug and release (final) versions: "Build BSD". The default is to build everything.

If you install the C++ toolkit under a different name than "ncbi_cxx" or in a different location than your home directory, you can edit the script's properties, pRootFolderName and pRootFolderPath, to override these defaults. Note: these paths, and those mentioned below, must be entered in mac format (e.g. disk:Users:username:) not Unix format (e.g./Users/username/). The disk name (and its following colon) may be omitted.

Certain third party libraries (seeTable 10) are required to build the C++ toolkit. The scripts will try and find them if they are in your home directory, or you can specify where they were installed using properties at the beginning of the script.

Table 10

Third Party Libraries

Library	Property	Example
FLTK 1.1.5rc2 (w/ NCBI patches)	pFLTKRootFolder	"usr:local:src:fltk-1.1.5rc2:"
BerkeleyDB 4.2.52	pBdbRootFolder	"Users:myhome:mylibs:db-4.2.52"
SQLite 2.8.15	gSqliteFolder	
dlcompat	pDLRootFolder	"usr:"

You do not have to build the FLTK or BDB libraries separately. This is done by the scripts and Codewarrior along with the toolkit libraries. Just unpack the source bundles in your home directory or where ever you have specified in the appropriate properties. The root folders for FLTK and BDB do not have to have any particular names. If there is more than one version the scripts will grab whichever is last alphabetically (e.g. fltk-1.1.4r2 will get used instead of fltk-1.1.3).

The scripts normally halt on any Codewarrior compilation errors. If you want them to continue and save errors, set the next script property, pSaveContinueOnErrors, to true. Compilation errors for a project will be saved in a file in the same folder as the project being built, with a name in the following format: projectName-targetNumber.errs (e.g. xncbi-2.errs).

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

The Genome Workbench's configuration file(s) is stored in the users Library:Application Support:gbench folder.

CFM builds are not supported. OS 8 or 9 are not supported. We know 10.2 works. We think 10.1 still works, and 10.3 might work.

Platforms (OS's, compilers used inside NCBI)

This release was successfully tested on the following platforms -- but may also work on other platforms. Since the previous release, some platforms were dropped from this list, just because we do not use them here anymore, and some were added (these new platforms are highlighted using bold font). 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" formakeon UNIX), can get you through.

Unix

Table 11

Unix OS's and Supported Compilers

Operating System	Platform	Compilers
Linux-2.4.23 (w/ LIBC 2.2.5)	INTEL	GCC 2.95.3, 3.0.4,**3.4.0**
Linux-2.4.23 (w/ LIBC 2.2.5)	INTEL	ICC 7.1
Solaris-8	SPARC	WorkShop 6 update 2 C++ 5.3 Patch 111685-13 (64-bit, 32-bit)
Solaris-8	SPARC	GCC 3.0.4
Solaris-7	INTEL	WorkShop 6 update 2 C++ 5.3 Patch 111685-13
Solaris-7	INTEL	GCC 3.0.4
IRIX64-6.5	SGI-Mips	MIPSpro 7.3.1.2m (64-bit, 32-bit)
FreeBSD-4.5	INTEL	GCC 3.0.4
Tru64 (OSF1) V5.1	ALPHA	GCC 3.3.2
Tru64 (OSF1) V5.1	ALPHA	Compaq C++ V6.5-014

MS Windows

Table 12

MS Windows and Supported Compilers

Operating System	Compilers
MS Windows	MSVC++ 6.0 Service Pack 5. To be DEPRECATED -- this is the last release where it is supported.
MS Windows	MSVC++**7.1**. See documentation onMS Visual C++.NET.

Mac OS X

Table 13

Mac OS, and Supported Compilers

Operating System	Compilers
MacOS 10.2,**10.3.4**	GCC 3.3
MacOS**10.3.4**	CodeWarrior**9.2**

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Caveats and Hints

MacOS 10.X / CodeWarrior 9.1

1. Not all of the test or demo applications are built.
2. The source code for the latest release of FLTK (1.1.x), BerkeleyDB (4.x) and SQLite (2.x) should be present. See the installation instructions for details.

MacOS 10.2/GCC 3.3

At least the GCC 3.3 update for Dec. 2002 Developers Tools required from Apple.

GCC 2.95

1. Poor MT-safety record.
2. Relatively incomplete/incorrect (comparing to modern compilers) STL implementation.
3. It is going to be deprecated in NCBI rather soon -- as soon as we have any significant trouble with its maintenance.

GCC 3.0.4

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

GCC 3.3

Other than the feature described below, GCC 3.3.2 has been very good for us; it has a lot of very ugly bugs finally fixed.

Painfully slow linking in debug mode on Linux with GCC-3.3 compiler:

1. Starting with binutils 2.12 linker tries to merge constant/debug strings marked for merging in object files. But it seems it does this job very inefficiently - I've seen messages about it in internet.

   GCC starting with version 3.2 marks section of string constants ready for merging, and also has an option to disable this flag in object files (-fno-merge-constants). Adding this flag to compilation stage allows to avoid slow linking.

   GCC 3.3 also sets merge flag for debug sections and unfortunately there is no option to disable this flag. As a result, linking of debug executables significatly slower than with gcc 3.0.4.

   The slowdown rate depends on size of debug strings section and it's non-linear, so bigger projects will suffer more of this bug.

   We had to patch GCC 3.3 in-house with the fix described athttp://lists.boost.org/MailArchives/boost/msg53004.php.
2. Long-file support still broken (first broken in 3.1).

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

GCC 3.4

1. The "Painfully slow linking..." (see GCC3.3 [1] above) is still an issue. -- Again, had to patch it in-house to speed it up, a la GCC 3.3.
2. 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.

Last Updated

This section last updated on August 25, 2004