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

The NCBI C++ Toolkit

Release Notes (August, 2006)

Created: August 14, 2006.
Last Update: August 22, 2006.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

— GCC 3.4.x, 4.0.x

Download

Download the source code archives at:

ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/2006/Aug_14_2006/

- ncbi_cxx--Aug_14_2006.tar.gz — for UNIX'es (see the list of UNIX flavors below) and MacOSX/GCC
- ncbi_cxx--Aug_14_2006.gtar.gz — for UNIX'es (see the list of UNIX flavors below) and MacOSX/GCC
- ncbi_cxx--Aug_14_2006.exe — for MS-Windows (32- and 64-bit) / MSVC++ (7.1, 8.0) — self-extracting
- ncbi_cxx--Aug_14_2006.zip — for MS-Windows (32- and 64-bit) / MSVC++ (7.1, 8.0)
- ncbi_cxx_mac_xcode--Aug_14_2006.gtar.gz — for MacOSX 10.4 / Xcode 1.5-2.2.1

The sources correspond to the NCBI production tree sources from patch "RELEASE_AUG_2006", which in turn roughly corresponds to the development tree sources from the very end of July, 2006.

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

Build

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

New Developments

CORELIB — Portability and Application Framework

1. IBlobStorage::DeleteStorage() — new method, to delete the whole storage
2. CExec::Spawn*() — changed return type from 'int' to 'CResult'
3. CExec::QuoteArg() — new method to help quote cmd.line arguments
4. CExec::Wait() — now can work with a *list* of process handles
5. NStr::Split(), Tokenize(), TokenizePattern() — can optionally return the tokens' positions in the string
6. CTime — use special new type TSeconds to represent seconds
7. CStopWatch::AsSmartString() — new method to facilitate the printing of time span
8. FindFilesInDir<> — now can find files in sub-directories without applying masks to sub-directory names
9. Diagnostics — output now can be split and sent to several log files: error, trace and application access log
10. Added diag-stream manipulators Severity() and Message.

11 Command-line arguments — allow '-' in argument names. Allow '=' as separator between argument's name and value. Allow to omit separator for single-char names. Added argument aliases.

12 stream_utils.hpp — moved from UTIL to CORELIB

13 CRWStreambuf — now can report the stream's current position

14 CNcbiApplication::SetExitCode() — new (protected) method to force

15 AppMain() to return a specified value, either unconditionally or only on uncaught exceptions

CONNECT — Data streaming, Networking, and Dispatching

1 CPipe — allow to specify current working directory and environment for the child processes

2 Generic routines, which are not connection-related, such as GetUsername, GetVMPageSize, CRC32 moved from ncbi_connutil.[ch]pp to ncbi_util.[ch]pp

3 UTIL_MatchesMaskEx() — new function, to control over whether case-sensitive comparison must be done when matching the mask

4 ncbi_local.[ch] — implements LOCAL service mapper, the one that does not require NCBI or load-balancer to resolve services

5 SConnNetInfo::http_referer — new field, added and supported throughout HTTP/ service connections

6 ncbi_service.c — use HTTP referrer for services, if specified. Otherwise, set a default one, according to the mapper type being used

7 SSendMailInfo::mx_options — new bitmask field, to replace boolean field SSendMailInfo::mx_no_header. It's backward-compatible; the new options are to control how to deal with incomplete host names (non-FQDN), stricter mailer agents, and for future extensions. MX_PORT, MX_HOST and MX_TIMEOUT retired.

UTIL — Miscellaneous Low-Level APIs

1 CRegexp::WildcardToRegexp() — new method to convert wildcard search pattern to the regular expression one

2 CZipCompression::EstimateCompressionBufferSize() — added

3 CBlobStorage_File — new, file based implementation of IBlobStorage interface

4 CCache::CreateElement() — new callback to allow creation of absent elements on the fly

5 ILineReader — new lightweight interface for getting lines of text with minimal memory copying

6 CStreamLineReader, CMemoryLineReader — implementations of interface ILineReader for input streams and memory regions, respectively

SERIAL — Data Serialization (ASN.1, XML)

1 Data objects generated from DTD or XML schema specs now can be serialized in ASN.1 text or binary format so that writing such object in ASN.1 format and reading it back re-creates an exact copy of the original object.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

DATATOOL — Code Generator and Data Converter Tool

1. XML Schema parsing — now, in addition to ASN.1 and DTD specifications, it is possible to generate the C++ serialization code out of the XML Schema specs
2. XML Schema code generation — redesigned to preserve in ASN.1 the local elements defined in XML Schema
3. DTD parser — now can preserves comments found in DTD specification and print them out when converting the specification to other formats (ASN.1 or XML Schema)

CGI — CGI and Fast-CGI Application Framework

1. Set session tracking cookie by default
2. New CGI log formatting. Default log location set to /log/<port >/.
3. [FastCGI] WatchFile.RestartDelay — new configuration parameter, directing FastCGIs to stagger their restarts over the corresponding number of seconds

HTML — HTML Generation Library

1. CNCBINode::RemoveChild() — added
2. CHTMLPage — changed CreateTemplate(), LoadTemplateLibFile() and
3. SetTemplateFile() to minimize system calls if file template caching is enabled

BDB — Yet Another C++ API Based On BerkeleyDB

1. Implemented a split BLOB storage.

DBAPI — SQL Database Connectivity

1. CDB_ classes — interface and implementation have been separated
2. "sqlite3" — new driver, for SQLite v3 database. It implements LanfCmd and BCPCmd interfaces.
3. CDB_UserHandler — is inherited from CObject now.
4. CDBConnectionFactory — new methods CalculateConnectionTimeout() and CalculateLoginTimeout().
5. CDB_Exception — server and user names have been added as members. Also, added Set/GetServerName() and Set/GetUserName() access methods
6. CDB_Exception::Get/SetSybaseSeverity — new methods. Sybase severity will be available with Sybase ctlib/dblib driver only. It is not available with other drivers
7. I_SendDataCmd::Cancel(), CDB_SendDataCmd::Cancel() — new
8. C***Cmd::Release() — deprecated
9. C_ITDescriptorGuard — deprecated
10. I_BaseCmd::WasSent(), I_BaseCmd::WasCanceled() — deprecated
11. I_DriverContext — argument EOwnership has been added to methods PushCtxMsgHandler() and PushDefConnMsgHandler()

ALGO/ALIGN/SPLIGN — Spliced and Generic Alignment Algorithms

1. CSplign — upon completion of alignment, compartments with identities below the threshold specified by the minimal singleton identity parameter are screened off. Extra penalty is given to in-cds indels to void unneccesary frameshifts. Coding regions are identified on the fly as longest ORFs. Splign's pairwise mode benefited from using

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

the new version of the FASTA reader. The common 'sense' and 'antisense' terminology is now used to designate cDNA alignment direction.

2 CHitFilter — Coordinate marging argument introduced to balance between RAM and speed. Length and identity cut-offs are applied at the output. A new parameter (retain_overlap) has been introduced to exempt alignments overlapping over intervals longer than this parameter from uniquification.

BLAST

1 Reorganization of the BlastQueryInfo structure

2 Residues 'O' and 'J' are replaced by 'X' in queries

3 Standard scoring matrices are first searched for inside XUTIL library — before attempting to read them from the file system

4 Introduced functionality to compute blastn word size to optimize the probability of finding hits with the specified properties

5 Allow for 26- and 28-letter alphabets in RPS-BLAST databases

6 Enabled the use of composition-based statistics for blastp by default

7 Refactoring and optimization of DiagStruct to reduce size of working set

8 CRemoteServices — new class, to retrieve data from BLAST that is not associated with a particular RID

9 CSeqDBExpert — new class, for obscure, internal, and experimental features

10 Add an experimental option for computing a p-value based on the composition of two sequences and for combining this value with an alignment E-value to produce a "unified" E-value

11 Composition adjustment library — made compatible with either a 26- or 28-letter amino acid alphabet, to reflect the recent addition of 'J' and 'O' to the alphabet used by BLAST blastp — to compute approximate gapped alignments in the preliminary stage of the BLAST search (see reference in s_RestrictedGappedAlign)

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

1 CSeq_align::CreateDensegFromDisc() — new

2 CSeq_align::RemapToLoc() — deprecated

3 CSeq_id::IdentifyAccession() — support the new prefixes EC-EE

4 CSeq_id — loosen syntax requirements for bare PDB accessions

5 CFastaReader — new reader for FASTA files, implemented as a class designed to support a wide range of potential specialized subclasses

6 CFeature_table_reader::ReadSequinFeatureTables() — new method that takes a CSeq_entry& on which to place the feature tables read

7 PackAsUserObject, UnpackUserObject — add support for BitString and AnyContent values

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

1 Allow use of separate scopes in standard prefetch actions.

2 Implemented waiting in CStdAction.

3 Implemented CPrefetchSequence for real limited prefetch.

4 Added CScope::GetObjectManager().

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

5 Added limiting range argument to CSeqMap_CI constructor.

6 Added overloaded CScope::GetObjectHandle() & CScope::GetObjectEditHandle().

7 Added CSeq_annot_ftable_CI.

8 CSeq_loc_Mapper — fixed mapping of std-segs, gaps in alignments, order of intervals in mapped locations, duplicate mappings, etc. Make it convert the mapped seq-loc-mix into packed-int if possible.

9 TestForOverlap() — new type eOverlap_CheckIntRev, to fix overlap testing in GetBestXXXForCds()

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

1 Added description of STS external annotations.

OM++ DEMO program (objmgr_demo)

1 id2_fetch — new option "-count" to issue repeated queries

BUILD FRAMEWORK (UNIX)

1 Configure supports a new "--with-flat-makefile" option that produces an alternative top-level Makefile.flat with rules for all configured applications and libraries

2 Meta-makefiles now can define XML-schema-based libraries by setting XSD_PROJ

PTB — Project Tree Builder for MSVC++ .NET

1 PTB now can work on UNIX. UNIX shell script 'scripts/create_flat_makefile.sh' builds PTB locally, and then uses it to generate flat makefile (Makefile.flat) in the build directory. The flat makefile allows for an effective use of parallel and distributed make, which considerably speeds up the build. It also makes it a breeze to build any single target application or library, with all needed dependencies.

2 Makefile.*.msvc tune-up files — now can redefine makefile macros on MS Windows platfrom

APPLICATIONS

1 NetCache

- Implemented cache cleaning function for ICache interface.
- Fixed bug in automatic session management shutdown.

2 NetSchedule

- Allow to give the status summary for the given affinity token
- Added return code to failure reporting
- Added output argument to PutFailure
- Fixed bug in job expiration/prolongation/restart algorithm
- Implemented methods to delay job expiration
- Implemented cout/cerr redirection for worker nodes
- Added option to immediately delete job when it is done
- More detailed log on the causes of server shutdown
- Added support for job flags (such as "exclusive" jobs)

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

3 NetBVStore — new (very experimental) server to provide a distributed storage of bit vectors, with convenient access to the selected parts of the bit vector data

GRID (DISTRIBUTED COMPUTING) FRAMEWORK

CONNECT/SERVICES — Components for the Network Grid Framework

1 CRADispatcherClient — new client API to remotely run command-line applications working under the control of the GRID framework, using HTTP (see also remote_app_dispatcher.cgi)

2 Added new configuration parameters for grid worker nodes:

- infinite_loop_time - maximum execution time for a job
- check_status_period - how often the node should check the state of jobs which are being processed

Grid Worker Node Implementation Framework

1 remote_cgi — an utility to run an arbitrary CGI application as a component of the Grid Framework

2 ns_remote_job_control — an utility to monitor and manage various components of the Grid Framework

3 remote_app and remote_cgi — provided with the new configuration parameters:

- keep_alive_period - how often to send notification messages to the queue server
- max_app_run_time - limit application's maximum run time
- non_zero_exit_action - what to do if the remote application returns a non-zero exit code

4 Remote Application — added support for an optional saving of STDOUT and STDERR into local files

5 Worker Node — now supports so-called Exclusive Jobs. The exclusive job is a job which is expected to occupy all resources of a worker node, so the worker node must not get any other jobs while an excusive job is being processed.

6 remote_app_dispatcher.cgi — a gateway CGI to give access to the Remote Applications via Web (see also its client API, CRADispatcherClient). In particular, it can allow non-NCBI clients to take advantage of the Remote Applications running on NCBI servers.

Documentation

Location

The documentation is available online as a searchable book "The NCBI C++ Toolkit": http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=toolkit.TOC&depth=2.

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

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

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 "Browsers" that appears on each page.

You can also access the CVS code repository via a Web interface. These links also appear in the sidebar box on each page.

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 public or in-house versions of LXR, Doxygen and Library. The Library search runs a CGI script that lists the library name where the symbol (such as function) is defined in.

Platforms (OS's, compilers used inside NCBI)

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, 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" for make on UNIX), can get you through.

Unix

Table 2. Unix OS's and Supported Compilers

Operating System	Architecture	Compilers
Linux-2.6.x (w/ LIBC 2.3.2 , 2.3.5)	x86-32	GCC 3.4.0 ICC 8.0 (build 20040520Z, package ID l_cc_pc_8.0.066_pe067.1) (GCC 3.0.4, 2.95.3, 3.4.2, **4.1. 1**- nominal support)
Linux-2.6.x (w/ LIBC 2.3.3 , 2.3.5)	x86-64	GCC 4.0.1 ICC 9.0 (build 20051201) (**GCC 4.1.1** - nominal support)
Solaris-8	SPARC	C++ 5.3 (WorkShop 6u2) patch 111685-23 (64-, 32-bit) (GCC 3.4.3 - nominal support)
Solaris-10	SPARC	Sun C++ 5.5 (Studio8) patch 113817-**18** GCC 4.0.1 (32-bit mode only)
Solaris-9	x86-32	C++ 5.3 (WorkShop 6u2) patch 111686-13 GCC 3.4.3
Solaris-10	x86-32	**Sun C++ 5.5 (Studio8) patch 113819-15** **GCC 4.1.1**
IRIX64-6.5	SGI-Mips	MIPSpro 7.3.1.3m (64-bit, 32-bit)
FreeBSD-4.10	x86-32	GCC 3.4.2
Digital Tru64 Unix 5.1 (aka OSF1)	ALPHA	GCC 3.3.2

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

MS Windows

Table 3. MS Windows and Supported Compilers

Operating System	Compilers
MS Windows-32	MS Visual Studio .NET 2003 (C++ 7.1). NOTE: We also ship an easily buildable archive of 3rd-party packages (including NCBI C Toolkit) for this platform.
MS Windows-32	MS Visual Studio .NET 2005 (C++ 8.0)
MS Windows-64	MS Visual Studio .NET 2005 (C++ 8.0)
Cygwin 1.5.18 - 32	GCC 3.4.4

Mac OS X

Table 4. Mac OS, and Supported Compilers

Operating System	Compilers
Darwin on MacOS X 10.4 ("Tiger")	GCC 4.0.1
Darwin on MacOS X 10.4 ("Tiger")	Xcode 1.5 - 2.2.1

Caveats and Hints

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 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 had been very good to us; it had a lot of very ugly bugs finally fixed.

1. Painfully slow linking in debug mode on Linux with GCC-3.3 compiler. — 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 significantly 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 (N^2). BINUTILS 2.15 fixes this. The link time still 2 times slower than without symbol merge, but the

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

resultant executable is about two times smaller in size, and no compiler patching is necessary. We are still testing it in-house. We had to patch GCC 3.3 in-house with the fix described at http://lists.boost.org/MailArchives/boost/msg53004.php.

2. Long-file support still broken.

GCC 3.4.x, 4.0.x

1. The "Painfully slow linking..." (see GCC3.3, [1] above) was an issue, and we had to patch it in-house to speed up, a la GCC 3.3 — until we finally upgraded to binutils 2.15.
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 (we didn't test whether 4.0.x fixed this).
3. At least on Linux, GCC 3.4.[0,1] optimizer (very rarely) generates incorrect code when comparing enumerated values in else-ifs. (Fixed in 3.4.2)
4. GCC 3.4.3, 3.4.4 (and maybe 3.4.5+) and 4.0 have 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 August 22, 2006.