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

# The NCBI C++ Toolkit

## Release Notes (December, 2008)

Created: December 24, 2008.
Last Update: December 24, 2008.

---

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

## Download

Download the source code archives at:

ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/2008/Dec_31_2008/

- ncbi_cxx-- Dec_31_2008.tar.gz — for UNIX'es (see the list of UNIX flavors below) and MacOSX
- ncbi_cxx-- Dec_31_2008.gtar.gz — for UNIX'es (see the list of UNIX flavors below) and MacOSX
- ncbi_cxx-- Dec_31_2008.exe — for MS-Windows (32- and 64-bit) / MSVC++ (7.1, 8.0) — self-extracting
- ncbi_cxx-- Dec_31_2008.zip — for MS-Windows (32- and 64-bit) / MSVC++ (7.1, 8.0)

The sources correspond to the NCBI production tree sources, which in turn roughly corresponds to the development tree sources from September 23, 2008.

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. CVersionInfo -- does not inherit from CObject anymore as to avoid double destruction problem. This is potentially backward incompatible change.
2. CWeakRef<> and CWeakIRef<> -- new templates for weak references. The templates work for the objects that derive from a new CObjectEx class.
3. Generic expression interpreter added. It is available via <corelib/expr.hpp>.
4. CLightString -- replaced with CTempString which already had nearly all functionality of CLightString.
5. NCBI-Boost unit testing framework -- new unit test framework which extends the Boost.Test library and adds useful features like automatic initialization, finalization, using of CNcbiApplication instance, disabling of test cases execution via ini-file, etc. (see corelib/test_boost.hpp)
6. CTime and CTimeSpan -- all the appropriate operators are const now (except operator-()).
7. CTime -- const operator-() deprecated. CTime:: DiffWholeDays() should be used instead.
8. CTime, CTimeSpan -- added support for non-strict date/time string parsing.
9. CTime -- allow constructing time object from the format string with partially represented time.

10 CTime -- operators to add/subtract/inc/dec days deprecated. Add*() methods should to be used instead.

11 File API -- error logging added for all File API classes.

12 CArgs:: GetAll() -- new method to get the list of all arguments passed in the command line.

13 CNcbiApplication -- allow to hide 'help' and 'xmlhelp' argument descriptions was added.

14 CNcbiApplication -- allow to specify an extended version information.

15 CNcbiApplication -- show a description of the possible arguments in case there was an error in argument parsing.

16 NStr:: JsonEncode() -- new helper method, for JSON string encoding.

17 CStringReader, ExtractReaderContents -- new utility class and function for efficient string/IReader inter-conversion.

18 CObject:: operator new() -- the filling of allocated memory became optional.

19 CSafeStaticRef<> -- memory leak was fixed.

20 Avoid exceptions while getting configuration parameters for efficiency.

21 Mutex's Lock()/Unlock() -- non-inline now for better code size and speed.

22 CDirEntry:: Remove() and CDir:: Remove() -- added an error logging: declared the FileAPILogging CParam for enabling file API error logging; introduced class CFileAPI - a name space for the file API-related global functions; defined CFileAPI:: SetLogging() for setting the FileAPILogging CParam.

23 CThread -- IRIX was added into the list of platforms for which PTHREAD_SCOPE_PROCESS and not PTHREAD_SCOPE_SYSTEM thread scope is set.

24 NStr -- URL encoding methods and string pairs parser class were added.

*Diagnostic framework*

1 CRequestContext -- new per-request diagnostic context class.

2 Allow limiting the rate of logging and log file size (including a basic log rotation).

3 Added idling API and default idler to allow execution of some code during application idle cycles. The default idler handles reopening of log files.

4 Added an option to duplicate a selected part of diagnostics into STDERR.

## CONNECT — Data streaming, Networking, and Dispatching

1 The default connection managing mode was changed to eKeepConnection.

2 CCgiUserAgent -- added detection of IceCat, Iceweasel and Google Chrome browsers.

3 The Grid services library has undergone major refactoring while keeping its public API almost intact (superfluous abstraction layers were eliminated; declarations of private class members were moved to the implementation part of the library; reference counting was introduced as a replacement for manual memory management; many internal identifiers got more meaningful names)

4 CNetICacheClient -- now supports LB service name resolution

5 CNetCacheAPI -- new method GetBlobSize(). See also the -size parameter of netcache_control utility.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

6 Protocol bug fix: SubmitJobAndWait() did not send the udp_port parameter to the server.

### CONNSERV

1 CNetCache_Key and CNetCacheKey -- merged into one, CNetCacheKey

### UTIL — Miscellaneous Low-Level APIs

1 CStdPoolOfThreads -- scheduled to be replaced with the new CThreadPool which is more robust and versatile

2 IScheduler and CScheduler_MT -- new interface and its multi-threaded implementation to schedule events.

3 CSyncQueue -- moved to UTIL along with minor improvements. It was located in CORELIB before.

4 XREGEXP library -- moved into a separate directory.

5 CRequestRateControl -- moved from UTIL to CORELIB.

6 CZipCompression:: DecompressFileIntoDir() -- new method.

7 value_convert.hpp and value_convert_policy.hpp -- two new files in 'include/util/'. These files implement Convert() and ConvertSafe() functions designed to simplify conversions among fundamental data types, std::string, and CTime.

8 NCBISM_GetStandardMatrix -- new interface for score matrix lookup by name.

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

1 Serial object streams -- enhanced support of XML serialization: fixed bugs in serialization of boolean and base64Binary type data, corrected namespace scoping, corrected parsing of 'xmlns' standard attributes.

2 SOAP library -- enhancements for better compliance to the standards

### DATATOOL — Code Generator and Data Converter Tool

1 Fixed bugs in DTD generation, DTD and XML Schema parsing, and C++ code generation.

### CGI — CGI and Fast-CGI Application Framework

1 FAST-CGI library -- MS Windows platform is now supported.

2 CCgiRequest:: fParseInputOnDemand -- new flag to parse input in streaming fashion, to be used in conjunction with various new methods, most notably CCgiEntry:: GetValueReader() and CCgiEntry:: GetValueStream().

### DBAPI -- Generic SQL Database Connectivity

1 FreeTDS-64 based driver:

- Sybase is fully supported now

- no 255 symbols limit anymore for Sybase

- TDS protocol version auto detection was added (MS SQL or Sybase). The supported versions are MS SQL (TDS 8.0) and Sybase (TDS 5.0).

- removed dependencies on OpenSSL library

- result set cursors implementation now supports several simultaneous cursors per connection and update of several blob fields per cursor row

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

2 ODBC and ODBCW drivers -- merged into one (ODBC) driver.

3 {CTLIB, DBLIB, etc.}_CreateContext() functions -- removed from DBAPI. This is potentially backward incompatible change.

4 CResultSet ::GetVariant() -- two methods merged into one which takes CDBParamVariant as an argument.

5 I_Connection and CDB_Connection -- two SendData() methods merged into one method.

6 I_Result and CDB_Result -- new default argument of type EGetItem added to the GetItem() method.

7 CDBConnParams:: GetParam() -- new method.

8 CDBConnParamsDelegate, CDBEnvConnParams, CDBInterfacesFileConnParams, and CCPPToolkitConnParams classes -- new classes.

9 CVariant -- now able to read BLOBs via GetString().

10 Max number of DBAPI connections is now configurable via ini file (section "dbapi", parameter "max_connection").

### Python DBAPI module

1 Attributes srv_errno and srv_msg were added to the DatabaseError class.

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

1 CSeq_id:: IdentifyAccession() -- recognizes more prefixes (AT_, DK, DL, FF-FL, FS-FZ, GA-GH, GM, GN, NZ_XX) and (mixed-in) EMBL TPA protein accessions.

2 SCigarAlignment -- support for segments with implicit lengths (of 1), and for callers to specify which variant of the format to expect.

3 Faster CSeq_id assignment in CSeq_id_Handle

4 CSeq_loc:: Assign() -- remember to clear Seq-id cache

### BLAST

1 Optimizations and MT bug fixes to local BLAST database data loader.

2 Introduced masking of subjects sequences.

3 Added support for smaller lookup tables for small queries.

4 (*) Moved seqdb and writedb libraries under objtools/blast.

5 Added blast2seq submission interfaces for CRemoteBlast.

6 Implemented a new method to compute effective observations and new entropy-based method to compute column-specific pseudocounts in PSI-BLAST.

7 Add support for WindowMasker in xblast library as well as BLAST+ command line binaries.

8 Improvements to legacy_blast.pl script.

9 Added src/app/blast/update_blastdb.pl script.

10 Added blast_formatter application.

11 Added support for comma-separated value output as well as support for custom output format specifiers in BLAST+ command line applications.

12 (*) ASN.1 output format in BLAST+ command line applications is of type Seq-annot.

13 (*) *-lcase_masking* BLAST+ command line option now applies to subject sequences as well as queries.

### Local data storage (LDS)

1 Fixed quadratic performance of LDS indexing.

2 Fixed MT-bottleneck in LDS loader.

3 Reuse open streams for files with several objects 4. Added PDB Seq-id indexing.

4 Reduce memory use in LDS indexer by processing entries separately.

5 All types of identifiers recognized by the Toolkit get indexed now.

### E-Utils client API

1 Added HTTP method selector to allow both GET and POST requests.

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

1 Added loading of named annotations.

2 Extended SNP features API: total range, quality data octet string.

3 Added support of "Extra" field to SNP table. Allow SNPs without strand.

4 Allow using shared CSeqMap object for simple CSeq_locs.

5 Allow changing CSeqVector and CSeqVector_CI strands.

6 Implemented loading accession.version which may be faster than full Seq-id resolution.

7 Added option fIgnoreUnresolved in CSeqMap and CSeqMap_CI.

8 Implemented breadth-first search for annotations on segmented sequences.

9 Added CFeat_id conversion class.

10 Added CScope::kPriority_Default to replace misleading kPriority_NotSet.

11 Added option to change priority of threads in prefetch manager.

12 Added API to retrieve GenBank loader statistics.

13 Avoid quadratic complexity when attaching split sequences.

14 Avoid slow setting of CSeq_id_Handle into CBioseq_Handle.

15 CBioseq_Handle:: GetSeqId() will try its best to find any id.

16 Many speedups by avoiding allocations and replacing STL map with vector.

17 Fixed broken annotation iterator after starting of sequence editing.

18 Use single map for all kinds of objects to reduce memory usage in index.

19 Simplify detection of cancel requests in CPrefetchManager.

20 Increase maximum allele length and count to accommodate larger SNP data in SNP features table.

21 Removed reference to deprecated headers objmgr/gbloader.hpp and objmgr/reader.hpp.

22 Moved Genbank readers one level up in build tree to shorten paths.

23 Store in the GenBank cache also blob state for ID2 data.

24 Remove trailing dots and spaces in sequence title.

25 Fixed duplication of features with non-exact Seq-id.

26 Treat virtual sequences w/o length as having zero length in CSeqMap.

27 Fixed feature duplication on edited sequences.

28 Fixed sequence:: GetId(..., eGetId_Best).

29 Set number of connect retries for ID1/ID2 service streams to 1, reader has its own retries.

30 Fixed loading external annotations via "gnl|Annot:x|gi".

### Object manager test and demo applications

1 test_objmgr_data -- added options to get accession, GI, or Seq-ids.

2 test_objmgr_data_mt -- added option to test single scope in MT.

3 test_objmgr_data_mt -- added blastdb loader to the MT-stress test.

4 objmgr_demo -- added options to retrieve annot types and names only.

5 objmgr_demo -- added option to work with named annotations' accessions.

## BIO-TOOLS

1 CFastaOstream -- flags renamed to preferred "fFlagName" syntax, and more added: fSuppressRange, fReverseStrand, fKeepGTSigns; passing out-of-range locations now yields appropriate exceptions.

## APPLICATIONS

*NetCache*

1 BerkeleyDB statistics are gathered now

2 Monitoring improvements (formatting, logging of opening and closing connections, always send message to monitor when blob is deleted as expired)

3 ClientIP and SessionID -- transferred from NetCache and ICache clients diagnostics to the NetCache server diagnostics.

4 A new (shared with NetSchedule) protocol parser is used. The new parser is in the CONNSERV library. The new parser allows parameters in a form of name=value.

5 Logging of request-start and request-stop entries for each incoming command was added

6 Protection against application hard-killing and attempt to restore database after one have taken place was added

7 A use case was added for PUT command. Within the use case NetCache sends a blob key only **after** the blob is written successfully

8 In waiting for a response, client is now notified even if an error occurred during writing to a database

9 Restored the code which cleans logs after cache cleaning

10 Transactions frequency is configurable now. The default value is 15 sec.

11 A new configuration flag added, that causes a database to be dropped if it was closed earlier incorrectly

*NetSchedule*

1 Instrumented with better request logging and timing trace

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

*Grid*

1. Client-side Grid tools moved from under app/netschedule/ to a new directory app/grid/.
2. Introduced versioning to the Grid package's applications.

*REMOTE_APP*

1. Improved logging -- log request numbers and the correct PIDs of child processes.
2. Removed the code that logged the entire STDOUT of the remote_app child processes.
3. Set the NCBI_NS_JID environment variable before running the child process.
4. Used the same environment for running the monitor as the app itself.

*NS_SUBMIT_REMOTE_JOB, NS_REMOTE_JOB_CONTROL*

1. Progress messages are sent and displayed now.

*NS_REMOTE_JOB_CONTROL*

1. The application now supports the *-stdout* parameter (dump stdout of the remote process for the specified job ID).
2. Bug fix: return non-zero exit codes in case of exceptions.

### BUILD FRAMEWORK (UNIX)

1. Configure frontends for various compilers are now available exclusively as compilers/unix/*.sh rather than compilers/*.sh.
2. This release drops support for IRIX, and for GCC 2.95.x on any platform.
3. Although default configurations continue to build some libraries in shared (dynamically loadable) form, they now arrange to favor static libraries over dynamic ones at link time, and to disable plugin autoloading by default. The existing --with(out)-dll flag and the new --with(out)-plugin-auto-load flag permit specifying alternative behavior.

### PTB -- Project Tree Builder for MSVC++ .NET

1. Added support (generation) for Microsoft Visual Studio 2008 projects.
2. Corrected analysis of DLL dependencies.
3. Changed configuration process on Windows to move site localization generated files into compiler and configuration name-specific location.
4. Modified CONFIGURE process to use prebuilt PTB by default, and to enable setting project list from an environment variable.
5. Modified generation of "flat makefile" on UNIX to allow directories as targets and to make it better report build errors.
6. Redesigned configuration process to move all logic into a separate standalone command line file.
7. In PTB, added protection against targets with the same name

## 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 NCBI C++ Toolkit Book

The NCBI C++ Toolkit 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.

### 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.

## Platforms (OS's, compilers used inside NCBI)

- Unix
- MS Windows
- Mac OS X
- 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; just because we do not use them inhouse 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.

**Please scroll down if you do not see tables.**

### Unix

**Table 2. Unix OS's and Supported Compilers**

| Operating System | Architecture | Compilers |
|---|---|---|
| Linux-2.6.x (LIBC 2.3.5) | x86-32 | GCC 3.4.**2**<br>ICC 8.0 (20040520Z, l_cc_pc_8.0.066_pe067.1)<br>(GCC 3.0.4, 4.1.2, 4.2.3- some support) |
| Linux-2.6.x (LIBC 2.3.5) | x86-64 | GCC 4.0.1, 4.1.2<br>ICC 9.0 (build 20051201)<br>(GCC 4.2.3 - nominal support) |
| Solaris-10 | SPARC, x86 (32/ 64-bit) | **Sun Studio 12 (C++ 5.9)** |
| Solaris-10 | SPARC | Sun Studio8 (C++ 5.5) 113817-19<br>GCC 4.1.1 (32-bit mode only) |
| Solaris-10 | x86-32 | Sun Studio8 (C++ 5.5) 113819-19<br>GCC 4.2.3 |

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

| Operating System | Architecture | Compilers |
|---|---|---|
| Solaris-10 | x86-64 | Sun Studio 11 (C++ 5.8) 121017-08<br>(nominal support only) |
| FreeBSD-6.1 | x86-32 | GCC 3.4.4 |

### MS Windows

**Table 3. MS Windows and Supported Compilers**

| Operating System | Compilers |
|---|---|
| MS Windows-32 | MS Visual Studio .NET 2003 (C++ 7.1).<br>(support to be discontinued in the next release)<br>NOTE: We also ship an easily buildable archive of 3rd-party packages (including NCBI C Toolkit) for this platform. |
| MS Windows-32 | MS Visual C++ 2005 (C++ 8.0)<br>NOTE: We also ship an easily buildable archive of 3rd-party packages (including NCBI C Toolkit) for this platform. |
| MS Windows-64 | MS Visual C++ 2005 (C++ 8.0) |
| Cygwin 1.5.1832 | GCC 3.4.4 (nominal support only) |

### Mac OS X

**Table 4. Mac OS, and Supported Compilers**

| Operating System | Architecture | Compilers |
|---|---|---|
| Darwin on MacOS X 10.4 - or newer | Native (PowerPC or x86-32),<br>Universal (PowerPC and x86-32) | GCC 4.0.1 |
| Darwin on MacOS X 10.5 | Native (PowerPC or x86-32) | Xcode **2.5, 3.1.2** |

### Table 5. Discontinued

| Operating System | Architecture | Compilers |
|---|---|---|
| Linux-2.6.x | x86-32 | GCC 2.95.3 (support to be discontinued in the next release) |
| IRIX64-6.5 | SGI-Mips | MIPSpro 7.3.1.3m (64-bit, 32-bit) |

## Caveats and Hints

### 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

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

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 avoiding 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 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 December 29, 2008.