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

The NCBI C++ Toolkit

Release Notes (October 2, 2004)

Release Notes (October 2, 2004)

Download Location

ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/2004/Oct_02_2004/

Source Archive Contents

Source Code Archives

- ncbi_cxx_unix--Oct_02_2004.tar.gz-- for UNIX'es (see the list of UNIX flavors below) and MacOSX/GCC
- ncbi_cxx_unix--Oct_02_2004.gtar.gz-- for UNIX'es (see the list of UNIX flavors below) and MacOSX/GCC
- ncbi_cxx_win--Oct_02_2004.exe-- for MS-Windows / MSVC++ 7.1 (self-extracting)
- ncbi_cxx_win--Oct_02_2004.zip-- for MS-Windows / MSVC++ 7.1
- ncbi_cxx_mac_cw--Oct_02_2004.tgz-- for MacOS 10.3.4 / CodeWarrior DevStudio for MacOS 9.2
- ncbi_cxx_mac_xcode--Oct_02_2004.tgz-- for MacOS 10.3.4 / xCode 1.[1-5]
- ncbi_cxx_mac_gcc--Oct_02_2004.tar.gz-- for MacOS 10.2, 10.3.4 / GCC 3.3.2

The sources correspond to the NCBI production tree sources from patch "CATCHUP_004", which in turn corresponds to the development tree sources from around the very end of August, 2004.

New Development

Some of the new development that may introduce potentially backward-incompatible changes are marked with an asterisk(*).

CORELIB -- Portability and Application Framework

1. GetVirtualMemoryPageSize()-- new function, to get the granularity with which virtual memory is allocated.
2. CNcbiApplication-- addedGetProgramExecutablePath()method.
3. CDllResolver-- extend class to make it also look in "standard" DLL paths. AddedAddExtraDllPath()method. Added accessory parameter toFindCandidates().
4. CDllResolver: added support for driver name (for Plugin Mananger)
5. PluginManager_ConvertRegToTree()- to convert application config file into a parameters tree.
6. CSimpleClassFactoryImpl-- added methods to extract information from parameters tree.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

7. CMemoryFileAPI -- added possibility to map parts of a file.
    - CMemoryFileSegment- new auxiliary class for mapping a memory file region.
    - CMemoryFileMap - new class to support a partial file memory mapping.
    - CMemoryFile - now is the same asCMemoryFileMapbut have only one big mapped segment with offset 0 and length equal to the length of file.
8. StringTo*() -- added additional parameter to specify the action to be performed on conversion error: to throw an exception, or just to return zero.
9. Fixed multithreading on BSD.
10. AddedCStopWatch::Restart()to reuse the same timer sequentially.
11. CTree<>-- added new methods:AddNode(),FindNode(),FindSubNode().

CONNECT -- Data streaming, Networking, and Dispatching

1. LSOCKandCListeningSocket::CListeningSocket()to accept flags (TLSCE_Flags), which extended from flags for simple logging to flags that also control binding of a socket being created: whether to bind locally (127.0.0.1) or promiscuously (any address) -- as before. Restricted binding helps limit incoming connections to loopback only, thus allowing to create local-only servers (which listen on and reply to localhost only).
2. Threaded Server Framework -- added a mechanism to request graceful termination.
3. Threaded Server Framework -- added support for running user-supplied code if a specified amount of time has occurred since the last client connected.

HTML -- HTML Generation Library

1. Added classCHTML_password(HTML tag <input type=password>)
2. CHTML_table-- replaced old unimplemented methodColumnWidth()with new oneSetColumnWidth().

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

1. CBZip2Compression, CZipCompression- fixed error inCompressFile()function with using incorrect file stream open mode.
2. ImplementedCNlmZipBtRdr::Pushback()for uncompressed data.

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

1. Completed first step in creation of infrastructure for developing .NET-compatible XML Web services. Created sample XML Web server and client that work on all platforms supported by the Toolkit.
2. Added possibility to scan input stream finding and processing objects of a specific type only, while skipping all other data: seeSerial_FilterObjectsfunction template.
3. Added (to "type iterators") the possibility to retrieve context information: type info, item info, object pointer: seeCTreeIteratorTmpl<>::GetContext,CTreeIteratorTmpl<>::GetContextData.
4. Modified serial library to support large files (greater than 4GB) on some 32 bits platforms - when compiler supports it.
5. RPC clients (for ID1, Entrez2, etc.) now support adjusting some additional parameters, including in particular timeouts.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

6 Added operators to read/writeCObjectInfo/CConstObjectInfo.

DATATOOL -- Code Generator and Data Converter Tool

1 Modified code generator to set and return primitive type data by value instead of by reference.

2 Corrected XML schema generation for named integers.

3 Added possibility to specify namespace name when generating XML schema.

4 Added possibility to specify extra headers to include into generated code; such headers are to be defined in DEF file.

CGI -- CGI and Fast-CGI Application Framework

The default implementation ofCCgiApplication::OnException()handler to issue a "400" HTTP error if it's the HTTP request itself that seems to be syntactically incorrect. "500" is still issued in all other cases.

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

1 New utility BDB dumper, to see BDB database content.

2 Bug fix in BLOB reading writing (cross platform compatibility).

3 Added log cleaning function (CBDB_Env::CleanLog()).

4 Added support for database recovery procedures when opening the Database.

5 Added new data field type "uint1" (unsigned char).

6 BDB_find_field improved to search in non-text fields.

7 BDB Cache(ICache):

1. support ASYNCronous write mode
2. improved page size control for better performance tuning
3. more efficient BDB environment join
4. cache verification method added (Verify())
5. added locking and transaction timeouts
6. added method Remove for key, version, subkey
7. added read-only cache

DBAPI -- Generic SQL Database Connectivity

1 AddedIReader/IWritersupport for BLOB I/O.

2 Added dedicated row counter for rowset results.

ALGO/ALIGN -- Generic Alignment Algorithms

New general purpose tree algorithms:

- TreeCompare - compare two trees using comparison functor
- TreeForEachParent - visit every parent of the tree node
- TreeTraceToRoot - find root of the tree
- TreeReRoot - tree re-rooting(rotation)
- TreeFindCommonParent - find common parent of two nodes
- TreeListToSet - convert list of nodes to bitset

- TreeMakeParentsSet - traverse all parents, create set of ids
- TreeMakeSubNodesSet - create set of subnode ids
- CTreeNonRedundantSet - compute non-redundant set of nodes
- CTreeMinimalSet - compute minimal set of nodes
- Functions to perform logical operations (AND, OR) on nodes

ALGO/PHY_TREE -- Phylogenetic and bio tree algorithms

CBioTree- New node design, improved usability.

ALGO/ALIGN -- Generic Alignment Algorithms

1. Supported low-case and zero-based nucleotide sequence characters.
2. Added a capability of parralel computing of parts of dynprog matrix inCNWAligner.
3. Synchronized compartment IDs for transcript alignments calculated in different query strands.
4. Provided a mechanism for elimination of weak/far terminal exons. At the preprocessing step, an empiric was introduced that limits external genomic search space based on the size of non-covered transcript ends. At the post-processing step, a heuristic was added to check for weak terminals.

BLAST

1. Consolidation of the traceback extension types.
2. Implemented stacks initial word container for all blastn extension methods.
3. Implemented uneven gap linking of HSPs for blastx.
4. Refactorings to PSSM engine.
5. Modularized code to link HSPs.
6. MT-support added toCDbBlast.
7. Added implementations of the BlastHspStream interface to collect HSPs according to the traditional BLAST algorithm and to collect all HSPs for on-the-fly processing.
8. CSeqDb: Added support for ISAM indices to allow lookup of sequences by GI, PIG, or String identifier; Allow memory bounds to be set; Added "chunk" interface to iterate over set of included OIDs.
9. The coordinates remapping for BLAST results when comparing subsequences or sequences from different strands was fixed.
10. Fixed compiler warnings and memory leaks, updated documentation, and removed obsolete defines.

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

1. Added support for converting generic serial objects to and from User-object trees.
2. CSeq_id-- support several new prefixes inIdentifyAccession()method.
3. Check strand inCSeq_loc::GetStart/End().

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

1. (*)CObjectManagermade singleton.
2. (*)CObjectManager-- added methods to manipulate Data Loaders.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

3. objmgr_sample.cpp updated to fit current APIs.
4. (*) Seq-id mapper moved from OBJMGR to SEQ library.
5. (*) Removed duplicate methods fromCSeqMap.
6. Modified annot iterators to better resolve synonyms. Do not cache synonyms in the scope's history.
7. Detect circular locations in Annot Iterators.
8. Added annotation mapping through annot.locs.
9. AddedCScope::GetIds()to fetch synonyms without fetching the whole bioseq.
10. CSeqVector-- added ncbi2na ambiguity randomizer.
11. CBioseq_set_Handle-- addedGetComplexityLevel()andGetExactComplexityLevel ().
12. Added operatorsbool()and!()to all Annot Iterators.
13. Revamped object manager: Changed TSE locking scheme; and, TSE cache is now maintained byCDataSource.
14. Fixed performance degradation ofGetTSESetWithAnnots(), used time was quadratic on amount of loaded blobs.
15. Fixed processing of trace assemblies: segment shift in Seq-annot.locs processing, and duplication of discontinuous alignments.
16. CAnnotNameandCAnnotTypeSelectormoved to separate headers.
17. Several fixes to make object manager thread-safe.
18. Understand "weight" param in qual field and string dbSNP value when parsing SNP features into table. Allow SNP Seq-entry in addition to SNP Seq-annot.
19. Added delayed loading of external annotations from satellite 26.
20. Fixed loading of missing split chunks whenGetCompleteXxx()method is called.
21. Added splitting of whole Bioseqs. Postpone indexing of retrieved split blobs to reduce memory usage by big nucprot sets.

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

1. (*)Data Loaders do not have public constructors anymore,RegisterInObjectManager ()should be used instead.
2. (*)Added Data Loader factories usingCPluginManager, fixed driver names.
3. CSeqrefis replaced byCBlobId.
4. Added loading of external annotations from 26 satellite. Use new split features to represent external annotation blobs (sat 26).
5. Allow SNP Seq-entry in addition to SNP Seq-annot. Load SNPs from satellite 26.

ID2

1. Added GENNBANK_READER_ID2_xxx macros to Makefile.mk.
2. Added exports for ID2 libraries. ID1 & ID2 moved to ncbi_seqext.dll on Windows.

BIO-TOOLS

Added preliminary support for importing and exporting GFF version 3 data, complete with gapped alignments.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

BUILD FRAMEWORK (MSVC++ 6.0)

Not supported anymore.

BUILD FRAMEWORK (MSVC++ .NET)

1. Allow for conditional macro definition.
2. Implemented optional dependency on a third-party library.
3. Do include EXPENDABLE projects to the build.
4. Added possibility to set up a precompiled header for a subdirectory which is not direct child of the root folder.
5. Corrected new_project and import_project Windows scripts to add more diagnostics, and to add the ability to import projects from the non-standard build tree.

APPLICATIONS

OBJMGR DEMO program (objmgr_demo)

1. Allow limited processing to have benefits from split blobs. Added new options to control tasks to be done:
   - count_types
   - reset_scope
   - limit_tse
   - used_memory_check
   - print_cds
   - range_from
   - range_to
2. UseCBlob_idinstead of obsoleteCSeqref. Removed obsolete code for old blob cache interface.
3. Seq-id cache is put in the same directory as blob cache. Tuned cache parameters on Windows.
4. Optional compilation with Berkley DB.

OBJMGR TEST programs (test_objmgr_data & test_objmgr_data_mt)

Added new options -no_seq_map, -no_named, -verbose, -adaptive -idlist.

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.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

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 "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.

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.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

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 14) are required to build some parts of 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 14

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 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.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Unix

Table 15

Unix OS's and Supported Compilers

Operating System	Architecture	Compilers
Linux-2.4.23 (w/ LIBC 2.2.5)	INTEL	GCC 2.95.3, 3.0.4, 3.4.0
Linux-2.4.26 (w/ LIBC 2.3.2)	INTEL	GCC 3.4.0
Linux-2.6.7 (w/ LIBC 2.3.3)	INTEL/**64**	GCC 3.4.0
Linux-2.4.23 (w/ LIBC 2.2.5)	INTEL	ICC **8.0**
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-**9**	INTEL	WorkShop 6 update 2 C++ 5.3 Patch 111685-13
Solaris-**9**	INTEL	GCC **3.3.3**
IRIX64-6.5	SGI-Mips	MIPSpro 7.3.1.2m (64-bit, 32-bit)
FreeBSD-4.5	INTEL	GCC 3.0.4
FreeBSD-4.10	INTEL	GCC **3.4.2**
Tru64 (OSF1) V5.1	ALPHA	GCC 3.3.2
Tru64 (OSF1) V5.1	ALPHA	Compaq C++ V6.5-014

MS Windows

Table 16

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 17

Mac OS, and Supported Compilers

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

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.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

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).

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.
3. At least on Linux, GCC 3.4.[1,2] optimizer (very rarely) generates incorrect code when comparing enumerated values in else-ifs. (Fixed in 3.4.2)

Last Updated

This section last updated on October 22, 2004