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

The NCBI C++ Toolkit

Release Notes (August 1, 2003)

Release Notes (August 1, 2003)

Download Location

ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/2003/Aug_01_2003/

Source Archive Contents

Source Code Archives

- ncbi_cxx_unix.tar.gz -- for UNIX'es (see the list of UNIX flavors below)
- ncbi_cxx_win.exe -- for MS-Windows / MSVC++ 6.0 (self-extracting)
- ncbi_cxx_win.zip -- for MS-Windows / MSVC++ 6.0
- ncbi_cxx_mac_cw.sit -- for MacOS 10.X / CodeWarrior 8.0 Update 8.3
- ncbi_cxx_mac_cw.tar.gz -- for MacOS 10.X / CodeWarrior 8.0 Update 8.3
- ncbi_cxx_mac_gcc.tar.gz -- for MacOS 10.X / GCC
- ncbi_cxx_mac_gcc.sit -- for MacOS 10.X / GCC

Other Files

- timestamp -- when the sources were checked

New Development

CoreLib (corelib)

1. Added CStringUTF8 class to support UTF-8 encoded strings.
2. Fixes in exception tracing to avoid printing of NUL chars ('\0') to log file/terminal.
3. Minor bug fixes for mutexes and threads.
4. Added complete set of constructors for C*MutexGuard.
5. Added typedefs TReadLockGuard and TWriteLockGuard to mutexes and rwlock.
6. Added inplace operators CObject::new and CObject::delete.
7. Fixes and extensions
    - Pipe API: rewritten the Unix version of the CPipeHandle class. Added CPipe/CPipeHandle::CloseHandle(). Added flushing a standard output streams before it redirecting.

- Fixed CNcbiRegistry to handle the entries with leading spaces.
- Fixed some memory leaks in the CTime class.

Data streaming, Networking, and Dispatching (connect)

1. Refined datagram socket API.
2. Implemented C++ datagram socket API.
3. UNIX socket acceptance in SOCK_CreateOnTop[Ex]() calls [still experimental].
4. Implemented 0-time connection timeouts for sockets and connectors.
5. Few fixes in HTTP connector code.
6. Fixed a bug in the CONNECT library that discouraged using functions from this library compiled as DLL.

Database Connectivity (DBAPI) (dbapi)

1. The ODBC DBAPI driver, which had been specific to Windows, now also supports Unix.

Berkeley DB API

1. Implemented BerkeleyDB abstraction layer library (BDB library). This includes classes to create local flat file databases and indexes. Those classes are tightly integrated with C++ Toolkit and it's error reporting and debugging facilities. As part of the library implemented first version of db_map and db_multimap template classes designed to imitate map and multimap templates coming with STL. Disk based templates offer controllable memory footprint and data persistence between program sessions.
2. Local data storage. Implemented set of classes allowing you to create a local database of serialized biological objects. The database keeps meta-information about files, formats, types of objects, sequences stored and in files, annotations. Developed simple query API to retrieve the annotation information.
3. Developed object manager dataloader to work with local data storage.

CGI/FastCGI Framework (cgi)

1. Fixed watch-file timeout logic.
2. Allowed to run FastCGI without web server in standalone mode.

Data Serialization (serial)

1. Added data initialization verification mechanism, which protects against accidental accessing or serialization of an uninitialized object data members.
2. New specification of IsSet() like methods of generated classes. See CanGet() methods too.
3. Added possibility to reference XML schema when serializing an object in XML format.
4. Added support of different encodings in XML streams. Two encoding schemata are implemented: UTF-8 and ISO-8859-1.
5. Allow SET OF to be implemented as vector<>.
6. Fixed thread safeness of object stream hooks.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Object Libraries

1. Minor bug fixes: compiler warnings, missing include statements, build order and dependencies, access to uninitialized members, CSeq_entry::Parentize(), Seq_feat::Compare(),CSeq_loc::GetTotalRange().
2. vector<>is used instead of list<> in several places for lesser memory usage.

Object Manager (objects/objmgr)

1. Added possibility to set priorities for data sources.
2. Significantly improved GetBestOverlappingFeat() and supporting functions.
3. Added CSeqVector_CI class for faster sequential access to the sequence data.
4. Improved processing of multi-ID seq-locs in the object manager functions.
5. Fixed interaction of GenBank data loader garbage collector and TSE locking mechanism.
6. Added CSeq_annot_CI class to fetch seq-annot objects rather than individual features, alignments and graphs.
7. Rewrote object manager internal structures to reduce memory usage and object dependency.
8. Fixed interaction of GenBank data loader garbage collector and TSE locking mechanism.
9. Rewrote GenBank readers: use pipelined data stream (with compression), SNP data are loaded separately, try to do strings' representation compression where possible to reduce memory usage.
10. Fixed locking of object manager classes in multi-threaded application.
11. Fixed reconnection to GenBank servers on timeout.
12. Added CSeqMap::FindResolved() with strand argument and constructor of CSeqMap from CSeq_loc.
13. Fixed backward traversal of CSeqMap_CI.
14. New implementaion of CSeqVector uses CSeqVector_CI.
15. Fixed indexing and removal of multi-location annotations.
16. Fixed depth of recursion of annotation iterator.
17. Reduced memory usage by features' indexes.
18. Rewrote and simplified CSeq_id_Mapper and CSeq_id_Handle.
19. Added several CXxx_ScopeInfo classes for CScope related information.
20. CBioseq_Handle now uses reference to CBioseq_ScopeInfo.
21. Added public CScope::GetSynonyms() method.
22. Fixed conflict resolution in CScope::GetTSESetWithAnnots().
23. CPriority_I made to use less memory allocations/deallocations.
24. As part of the object manager utilities implemented new class CObjSniffer. Class can be used for reading ASN.1 and XML serialized objects of uncertain structure.

Alignment Manager (objects/alnmgr)

1. Changes in CAlnMap:
    - Interface: optionally a chunk can be created for each segment

- Internal: speed optimization via const refs to Dense-seg members; reworked some of the internal inefficient code; added caching for the end segments of sequences

2. CAlnVec: optimized method for displaying sequences: GetWholeAlnSeqString
3. Changes in CAlnMix:
   - Several bug fixes and optimizations
   - Optional truncation or deletion for overlapping segments
   - Optional filling of the unaligned regions
   - Optional use of ObjMgr
4. Demo programs added:
   - alnvwr demonstrates various display methods using CAlnMap and CAlnVec
   - alnmrg shows CAlnMix usage

Miscellaneous Libs (additions and improvements)

1. `xutil' (util)
   - [md5] Computes MD5 digests (adapted from the version in the C Toolkit).
   - [checksum] Supports MD5.
2. `xobjread' (objtools/readers) -- NEW
   - [fasta]
     - Split out of Seq_entry.[ch]pp.
     - Properly handles comments and multiple-defline entries.
     - Optionally reports lowercase letters' location.
   - [readfeat] Newly added code for reading Sequin-style feature tables, courtesy of Jonathan Kans.
3. `xobjutil' (objects/util)
   - [genbank] Replaced with a trivial wrapper around the new generator in objtools/flat.
4. `xhtml' (html)
   - Now handles large templates much faster, especially when not adding JavaScript menus..
5. `compress' (util/compress)
   - Implemented C++ wrappers for popular compression libraries (zlib, bzip2).
   - Added compression based i/o streams.
6. `bzip2' (util/compress/bzip2)
   - bzip2 library (see LICENSE) has been embedded into the Toolkit.
7. `zlib' (util/compress/zlib)
   - zlib library (see LICENSE) has been embedded into the Toolkit.
8. cgi (cgi)
   - Added optional parameter to the URL_Encode() to enable mark characters encoding and also enlarged a test for this function
9. Split up the "network client" part off for the objects libs in the MSVC++ static and DLL projects trees.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Build Framework

Configuration on Unix

The Unix `configure' script now checks for several additional third-party libraries, including in particular the image-format libraries the new `ximage' library needs. To deal with the hassle of finding lots of third-party libraries that may not all be in the same place, it also supports obtaining their locations from a file called `config.site'; `config.site.ex' is an example version showing the default settings. (It is still also possible to specify these locations on the command line or in the environment.)

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 (except testvalidator). We also build the fltk library's GUI editor, fluid.

All apps are built as application bundles except gbench_plugin_scan and datatool which 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 1) 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 1

Third Party Libraries

Library	Property	Example
fltk	pFLTKRootFolder	"usr:local:src:fltk-1.1.4:"
bdb	pBdbRootFolder	"Users:myhome:mylibs:db-4.1.25"

Library	Property	Example
dlcompat	pDLRootFolder	"usr:"

The latest release of fltk, 1.1.4x should be used.

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.

Documentation

Document Location

The documentation is available at

http://www.ncbi.nlm.nih.gov/books/bv.fcgi?call=bv.View..ShowTOC&rid=toolkit.TOC&depth=2

as a book titled "The NCBI C++ Toolkit". This is an online searchable book.

The C++ Toolkit book also provides PDF version of the chapters. Look for the Adobe Acrobat icon next to the chapter title in the table of contents.

The older HTML documentation has been deprecated; it continues to be updated for the time being, but the future plan is to support only "The NCBI C++ Toolkit" online book at the previously listed URLs.

Document Content

The documentation has undergone a major reorganization. Previous documentation has been grouped into chapters and subsect1s that provide a more logical coherence and flow. New subsect1s and paragraphs continue to be added to update and clarify the older documentation or provide new documentation. A new overview chapter titled "Introduction to the C++ Toolkit" has been added that gives an overview of the C++ Toolkit. This chapter contains links to other chapters containing more details on a specific topic.

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:

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

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

A major effort is under way to document all source files, so that the DOXYGEN source browser can generate a detailed API reference for the classes/methods.

Backward Compatibility

Layout

The old `objects' tree has been split up; many of the headers are now under `objmgr' or `objtools'. In conjunction, the libraries `seqalign', `seqblock', `seqfeat', `seqloc', and `seqres' (but NOT `seqcode' or `seqset') have been merged into `seq'; this change should be transparent on Unix if you've been using the $(SEQ_LIBS) variable. Likewise, the libraries `mmdb1', `mmdb2', and `mmdb3' have been merged into a single `mmdb' library.

There are still forwarding headers in the old locations for code that moved to the objmgr or objtools tree; however, you should not rely on them, particularly given that some C++ compilers, including Sun WorkShop, fail to recognize the `#warning' directives they use. The headers for the merged seq* and mmdb libraries have not moved.

In addition, the FASTA reader has moved out of Seq_entry.[ch]pp into its own library (xobjread) along with the new feature-table reader, and the generated RPC clients have been split into their own little *cli libraries and had their sources and headers renamed to repeat the module name for global uniqueness.

OBJMGR_LIBS

Due to possible changes in functionality of object manager, the list of libraries it depends on may change and so will the list of libraries required by the application which uses object manager. To reduce changes in makefiles, the Toolkit's framework provides a variable OBJMGR_LIBS which specifies what libraries are used by object manager (including libraries with generated objects). So you can put $(OBJMGR_LIBS) in the definition of LIB variable in your makefile in place of xobjmgr and other libraries used by object manager library.

Platforms (OS's, compilers used inside NCBI)

Unix

Table 2

Unix OS's and Supported Compilers

Operating System	Platform	Compilers
Linux	INTEL	GCC 3.0.4, 3.3.1
Linux	INTEL	ICC 7.1
Solaris	SPARC	WorkShop 6 update 2 C++ 5.3 Patch 111685-13 (64-bit, 32-bit)
Solaris	SPARC	GCC 2.95.3, 3.0.4
Solaris	INTEL	WorkShop 6 update 2 C++ 5.3 Patch 111685-13
Solaris	INTEL	GCC 3.0.4
IRIX64	SGI-Mips	MIPSpro 7.3.1.2m (64-bit, 32-bit)
FreeBSD	INTEL	GCC 3.0.4
Tru64	ALPHA	GCC 2.95.3

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Operating System	Platform	Compilers
Tru64	ALPHA	Compaq C V6.3-029 / Compaq C++ V6.5-014

MS Windows

MSVC++ 6.0 Service Pack 5.....

Mac OS X

Table 3

Mac OS, and Supported Compilers

Operating System	Compilers
MacOS 10.1	GCC 3.1 (patched, see "doc/config_darwin.html)
MacOS 10.2	GCC 3.1 (patched, see "doc/config_darwin.html), GCC 3.3
MacOS 10.X	CodeWarrior 8.0 Update 8.3

Caveats and Hints

MacOS 10.X / CodeWarrior 8.0 Update 8.3

1. None of the test or demo applications are built.
2. The source code for the latest release of fltk, 1.1.x and berkeley database (4.x) should be present. See the release notes (or installation instructions) for details.

MacOS 10.2/GCC 3.3

GCC 3.3 update for Dec. 2002 Developers Tools required from Apple. All C++ Toolkit configurations will build and run just fine.

GCC 3.0.4

Detected two bugs in GCC-3.0.4 compiler:

- 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.
- STL stream uses locale in thread unsafe way which may result to segmentation fault.

  Fixed in GCC 3.3.

Last Updated

This section was last updated on August 28, 2003