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

The NCBI C++ Toolkit

Release Notes (December 8, 2003)

Release Notes (December 8, 2003)

Download Location

ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/2003/Dec_08_2003/

Source Archive Contents

Source Code Archives

- ncbi_cxx_unix--Dec_08_2003.tar.gz-- for UNIX'es (see the list of UNIX flavors below) and MacOSX/GCC
- ncbi_cxx_win_msvc6--Dec_08_2003.exe-- for MS-Windows / MSVC++ 6.0 (self-extracting)
- ncbi_cxx_win_msvc6--Dec_08_2003.zip-- for MS-Windows / MSVC++ 6.0
- ncbi_cxx_win_msvc7--Dec_08_2003.exe-- for MS-Windows / MSVC++ 7.0 (semi-experimental)
- ncbi_cxx_win_msvc7--Dec_08_2003.zip-- for MS-Windows / MSVC++ 7.0 (semi-experimental)
- ncbi_cxx_mac_cw--Dec_08_2003.tar.gz-- for MacOS 10.2 / CodeWarrior DevStudio for MacOS 9.1
- ncbi_cxx_mac_gcc--Dec_08_2003.tar.gz-- for MacOS 10.2 / GCC 3.3

Other Files

- timestamp -- when the sources' snapshot was made

New Development

Portability and Application Framework (corelib)

1. CPipeclass moved to xconnect library.
2. New process management classCProcess.
3. TheCPIDGuardclass moved fromncbi_system.[ch]pptoncbi_process.[ch]pp.
4. AddedCMetaRegistryto centralize configuration file search logic and allow potential reuse of parsed contents.
5. Added a cross-platformCPIDGuardfor the sake of daemons.
6. Added optional symlink resolution toCDirEntry::NormalizePath.

7. Made it possible to alter the verbosity of trace messages.
8. AddedCDllResolver- class scanning for DLLs and looking for specified entry points.
9. corelib/plugin_manager.hpp- Developed initial version of plugin manager (cannot be considered as a true release, but rather a pre-beta version)
10. Fixes and extensions
    - File API: Fixed bug in the File APIGetType()with get type of a dir entry on UNIX. Replaced usingtmpnam()withtempnam()/mkstemp()for creating temporary files in theGetTmpName[Ex]. Added functionsDeleteTrailingPathSeparator()andNormalizePath()to theCDirEntryclass. Added month and day of week names conversion functions to theCTimeclass.
    - Added month and day of week names conversion functions to the CTime class.
    - Exec API: Used correct function for run a processes on Unix for the eV-mode.
    - DLL API: ChangedCDllclass to better conform C/C++ standards.
    - RevampedGetEntryPoint()method to avoid mixing a pointers to data and functions.
    - File API: AddedCDirEntry::SetTime()method to change file access and modification times.
    - CTimeclass: Added setters for various components of time, added functions to determine number of week in the year and month, number of days in the month, reimplementedAddYear(), changedDayOfWeek()function to use other computation method.

Data streaming, Networking, and Dispatching (connect)

1. ncbi_socket.[ch][pp]:
    - Bug-fixes for proper destruction of underlying SOCK in case ofClose() orAbort()methods got called, but theCSocketis still in use.
    - Stall protection against waiting indefinitely if SOCK that is being waited for is closed from other thread (Linux specific issue).
2. ncbi_http_connector.[ch]:
    - Do not callshutdown()after sending HTTP request. Although 100% valid this call could cause some buggy software and hardware to react inappropriately as if the connection is going to be closed soon.
    - Support VHosts by passing "Host: " tag in the HTTP headers (essential if VHosts are being used with Apache web server software).
3. Pipes and named pipes moved from CORELIB into CONNECT.
4. Abstract classesIReaderandIWriterin UTIL library. Implementation of stream buffer on top of these classes.
5. Fix forCStreamUtils::Readsome()andPushback()API. Make all basic Toolkit streambufs (including those in CONNECT) standard-compliant with respect toxsgetn ()operation.
6. Performance and standard-compilance of FCGI I/O streams: FCGI I/O streams (cgi library) are significantly improved by eliminating intermediary buffers, and making the streams fully buffered directly on top of FCGIX structures.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

7 Performance of fstreams on Windows: MS-Windows (MSVC/6.0): fstreams are significantly improved by making an ad-hoc fix to make fstreams buffered by means of direct use of FILE* buffer pointers, and thus eliminating 1-bytefgetc()calls on every I/O.

Database Connectivity (DBAPI) (dbapi)

1 util/cache: Added specification for local cache interface (Work in progress, current version looks unstable and likely to change)

2 Local Data Storage (LDS):

- Fixed number of bugs
- Implemented recursive objects directory scan, now all subdirs are part of the database (Improved LDS data management)
- Implemented multiple LDS databases. Changed genome workbench to reflect this new feature.
- Added PID protection againts running several instances of program with the same local database (not supported yet and caused crash)

Berkeley DB API (bdb)

1 Added class for Berkeley DB environment support

2 Implemented stream like class on top of BDB BLOB (CBDB_BLobStream).

3 AddedCBDB_FileDumperclass to dump BDB files into text format

4 Implemented local cache for biological objects.

Data Serialization (serial)

1 Data initialization verification mechanism in I/O streams has been elaborated to allow for handling (skipping or using defaults) the unset mandatory data members both on reading and writing.

2 Added support of XML ANY data type to theDATATOOL(C++ code generator) and XML serialization I/O streams.

3 Added possibility to specify XML namespaces for serial objects.

4 Added support of multiple namespaces in XML streams.

5 Implemented basic XML SOAP packager, which includes an API to create a SOAP message, serialize (read or write) it, and investigate its contents.

6 Added possibility to reset an object member from a read hook (including non-optional members).

7 RedesignedDATATOOLto use new approach for type aliases. Instead of using wrapper classes, the newDATATOOLderives alias class from the aliased one if the aliased type is a class, or from a special template class in case of primitive types.

Object Manager (objects/objmgr)

1 Added or modified several seq-loc related functions to handle circular locations correctly (GetStart(),GetEnd(),GetCircularLength(),TestForOverlap()).

2 Introduced new classCAnnot_CIwhich allows to iterate over seq-annot container objects rather than over individual annotations.

3 Implemented caching id1 reader (uses BDB library to keep BLOBs and id resolution information on local disk).

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

4 Integrated local caching in Genome Workbench

Alignment Manager (objects/alnmgr)

Added support for translated (nucleotide to amino acid) alignments.

Sequence Alignment (objects/seqalign)

1. ExtendedCDense_segwithm_Widthsto support translated alignments.
2. Added a few methods for Seq-aligns (mostly for the Dense-seg type):
   - Alignment transformers: Reverse, SwapRows forCSeq_alginandCDense_seg, andCDense_seg::RemapToLoc
   - Validators:CheckNumRows,CheckNumSegsand Validate with optional full segments check.
   - Methods for range calculation:GetSeqRange,GetSeqStart,GetSeqStop.
3. Added a format converter:CSeq_align::CreateDensegFromStdseg.
4. Added validators.
5. Added methods for range calculation.

Generic Sequence Alignment Algorithms (algo/align)

The following changes and updates have been made to the algo/align library.

1. Formatting-related functionality has been moved from the rootCNWAlignerto a newCNWFormatterclass. Code using old aligner's formatting members will work fine with minimal adjustment. For details please see the updated demo applications.
2. CNWAlignerMrna2Dnawas replaced with three classes to facilitate quality/ performance trade-off in case of distorted sequences:
   - CSplicedAligner. Abstract base for spliced aligners;
   - CSplicedAligner16. Accounts for three conventional splices and a generic splice; uses 2 bytes per backtrace matrix cell
   - CSplicedAligner32. Accounts for three conventionales and splices that could be produced by damaging one or more bps of any conventional; uses 4 bytes per backtrace matrix cell.

Use the 16-bit version of the above classes for low-error sequences, and the 32-bit version for sequences with higher error rates.

A demo new application called Splign was introduced. Splign is built on top of theCSplicedAlignerclasses and is a real-world tool for doing spliced sequence alignments.

Miscellaneous Libs (additions and improvements)

1. `xutil' (util): [regexp] Added wrapper for the Perl-compatible regular expression (PCRE) library andCRegexpclass --CRegexpUtil.
2. `xhtml' (html) : UpdatedCPopupMenuclass to use new version 2.3 of the Sergey Kurdin's JavaScript PopupMenu2. Added support for Sergey Kurdin's slide menu.
3. New MSVC 6.0 project file converter to expand a single configuration project file to multi-configuration project file (src\util\msvc\one2all.cpp).
4. tables' (util/tables) -- NEW. C API for working with score matrices, along with (generated) hard-coded copies of standard matrices.

5 `xobjread' (objtools/readers). [fasta] Now reads large FASTA sequences much more efficiently.

6 `xflat' (objtools/flat). Added support for GFF/GTF [semi-experimental].

7 `general' (objects/general). Added manipulators for CInt_fuzz.

8 `xobjutil' (objects/util). Improved relative-location support, which now handles fuzz and opposite-strand relative locations.

9 `seq' (objects/seq*). Added support for several new prefixes to CSeq_id::IdentifyAccession.

10 util/file_obsolete.h: AddedCFileObsoleteclass - removes local files based on file mask and creation/access time.

11 util/format_guess.hpp:CFormatGuess- implemented new rules of format recognition with much improved our work with binary ASN.1 files.

BLAST algorithms, C and C++ API (algo/blast)

Integrated BLAST's core "C" (shared with the NCBI C Toolkit) and "C++" API. Actual development done by the BLAST group.

Build Framework

Configuration on Unix

The Unix build framework adds make variables ({CC,CXX,LINK}_WRAPPER) for specifying wrapper programs such as Purify or ccache. As such, `configure' now looks for ccache (http://ccache.samba.org/), and takes advantage of it when present. NB: ccache does not support Sun's (WorkShop) C++ compiler, and requires users to set the environment variable CCACHE_EXTENSION to "i" when using SGI's MIPSpro compiler.

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.

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 4) 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.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Table 4

Third Party Libraries

Library	Property	Example
FLTK	pFLTKRootFolder	"usr:local:src:fltk-1.1.4:"
DBD	pBdbRootFolder	"Users:myhome:mylibs:db-4.1.25"
dlcompat	pDLRootFolder	"usr:"

The latest release of fltk, 1.1.4 should be used.

You do not have to build the FLTK or DBD 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 DBD 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

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; although these are not up to date in this release.

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 NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

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 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. A top-level meta module grouping exists under "Modules" on the Doxygen main page. This can be used to get a conceptual grouping of the different components.

Platforms (OS's, compilers used inside NCBI)

Unix

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Table 5

Unix OS's and Supported Compilers

Operating System	Platform	Compilers
Linux	INTEL	GCC 3.0.4, 3.3.1, 3.3.2
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
Tru64	ALPHA	Compaq C V6.3-029 / Compaq C++ V6.5-014

MS Windows

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

MSVC++ 7.0 (semi-experimental)

Mac OS X

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Table 6

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 9.1

1. None of the test or demo applications are built.
2. The source code for the latest release of FLTK, 1.1.4 and berkeley database (4.1) 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.

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:

- 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 at "http://lists.boost.org/MailArchives/boost/msg53004.php".

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Last Updated

This section last updated on December 16, 2003