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

The NCBI C++ Toolkit

Release Notes (April 16, 2004)

Release Notes (April 16, 2004)

Download Location

ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/2004/Mar_31_2004/

Source Archive Contents

Source Code Archives

- ncbi_cxx_unix--Mar_31_2004.tar.gz-- for UNIX'es (see the list of UNIX flavors below) and MacOSX/GCC
- ncbi_cxx_unix--Mar_31_2004g.tar.gz-- for UNIX'es (see the list of UNIX flavors below) and MacOSX/GCC
- ncbi_cxx_win_msvc6--Mar_31_2004.exe-- for MS-Windows / MSVC++ 6.0 (self-extracting)
- ncbi_cxx_win_msvc6--Mar_31_2004.zip-- for MS-Windows / MSVC++ 6.0
- ncbi_cxx_mac_cw--Mar_31_2004.tar.gz-- for MacOS 10.2 / CodeWarrior DevStudio for MacOS 9.1
- ncbi_cxx_mac_gcc--Mar_31_2004.tar.gz-- for MacOS 10.2 / GCC 3.3

New Development

Portability and Application Framework (corelib)

1. Bug fixes to low-level atomicity support.
2. FixedCProcess::Killmethod to handle zero timeouts.
3. FixedCDirEntry::DeleteTrailingPathSeparator()andCDirEntry::CreatePath() functions to avoid creating empty directories with disk name in the case if specified path contains it.
4. Added some fixes concerning .NET 2003 compiler support.

Data streaming, Networking, and Dispatching (connect)

1. Buffers and memory streams can now efficiently accumulate large quantities of data. BUF API insertion sped up by keeping and using "last" pointer in the chain.
2. Fixed error in theCConn_PipeStreamwith implicit loop in destructor.
3. CPipe: new creation flags, add GetProcessHandle(), changed exception handling.
4. CONN_Write(): new paramater "write_mode".

5. New XML content types added.
6. SendMail API extended to allow list of recipients, custom or no headers, and better body handling.
7. CConnStreamBufis made seekable to the end, which allows one to obtain the size of the stream viatellp().

HTML

1. UpdatedCPopupMenuclass to use new version 2.4 of the Sergey Kurdin's JavaScript popup menu.
2. Added possibility to use Kurdin's popup menus (CPopupMenuclass) with onClick JavaScript event. Use milliseconds instead second to set auto-hide timeout.
3. Added HTML tracing ability to all classes on exceptions.
4. Fixed function to strip HTML tags. Added code to strip special HTML numeric and named entities.
5. Improved performance ofCHTMLHelper::HTMLEncode().
6. Added NOWRAP attribute support.
7. Added TagMapper`s functions and class methods which used a data parameter.
8. AllowCNCBINodeclass to repeat stored context.
9. Implemented HTML template library support (CHTMLPage::LoadTemplateLib*()). Added demo apllication (html/demo/demo_html_template.*).
10. Fixed some errors in the library classes. Added missed export specifiers.

UTIL

1. Added templatesCStaticArrayMap<>andCStaticArraySet<>to provide convenient map<>/set<> access to a statically-defined array, while making sure that the order of the array meets sort criteria in debug builds.
2. Few more patches toReadsome()iterface.
3. AbstractIReaderandIWriteras well asIRWStreambufrefined.
4. CStreamUtilscode refined

Berkeley DB API (bdb)

1. Added support for BerkeleyDB transactions.
2. Added simple query API.

Miscellaneous Libs (additions and improvements)

- `xcgi_redirect` (misc/cgi_redirect). Implemented framework for CGI redirect applications. Added standard default CGI redirect application.

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.

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

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 7) 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 7

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 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.

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.

Platforms (OS's, compilers used inside NCBI)

Unix

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Table 8

Unix OS's and Supported Compilers

Operating System	Platform	Compilers
Linux	INTEL	GCC 2.95.3, 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 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 3.3.2
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, not in this distribution)

Mac OS X

The NCBI C++ Toolkit Book

The NCBI C++ Toolkit Book

Table 9

Mac OS, and Supported Compilers

Operating System	Compilers
MacOS 10.2	GCC 3.3
MacOS 10.X	CodeWarrior 9.1

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, 1.1.4 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 April 16, 2004