

The NCBI C++ Toolkit

Release Notes (Version 7, May 2011)

Created: May 1, 2011.

Last Update: May 17, 2011.

-
- [Download](#)
 - [Third Party Packages](#)
 - [Build](#)
 - [New Developments](#)
 - [HIGHLIGHTS](#)
 - [CORELIB](#)
 - [SERIAL](#)
 - [DATATOOL](#)
 - [CONNECT](#)
 - [UTILITIES](#)
 - [BIO-OBJECTS](#)
 - [BIO-TOOLS](#)
 - [COBALT](#)
 - [PHYLOGENETIC TREES](#)
 - [BIO-OBJECT LIBRARIES](#)
 - [BIO-OBJECT MANAGER](#)
 - ◆ [New functionality](#)
 - ◆ [Improvements](#)
 - ◆ [Bug fixes](#)
 - [GENBANK DATA LOADER](#)
 - ◆ [New functionality](#)
 - ◆ [Improvements](#)
 - [OBJECT MANAGER TEST AND DEMO APPLICATIONS](#)
 - [FASTA](#)
 - [asn2flat utility](#)
 - [XMLWRAPP](#)
 - [WindowMasker](#)
 - [BUILD FRAMEWORK \(UNIX\)](#)
 - [BLAST](#)
 - [APPLICATIONS](#)
 - ◆ [NETCACHE](#)
 - ◆ [GRID APPLICATIONS](#)
 - [Documentation](#)
 - [Location](#)

- [Content](#)
- [Supported Platforms \(OS's and Compilers\)](#)
 - [Unix](#)
 - [MS Windows](#)
 - [Mac OS X](#)
 - [Added Platforms](#)
 - [Discontinued Platforms](#)
- [Caveats and Hints](#)
 - [GCC 3.4.x](#)

Download

Download the source code archives at:

ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools++/ARCHIVE/7_0_0/

- [ncbi_cxx--7_0_0.tar.gz](#) — for UNIX'es (see the list of UNIX flavors below) and MacOSX
- [ncbi_cxx--7_0_0.exe](#) — for MS-Windows (32- and 64-bit) / MSVC++ 9.0 — self-extracting
- [ncbi_cxx--7_0_0.zip](#) — for MS-Windows (32- and 64-bit) / MSVC++ 9.0

The sources correspond to the NCBI production tree sources, which are originally based on the development tree source snapshot from October 25, 2010 but also include many hundreds of important and safe code updates made since then and through March 28, 2011 (and then some).

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](#)

Third Party Packages

Some parts of the C++ Toolkit just cannot be built without 3rd party libraries, and other parts of the Toolkit will work more efficiently or provide more functionality if some 3rd-party packages (such as BerkeleyDB which is used for local data cache and for local data storage) are available.

For more information, see the FTP README.

The following table shows the versions of 3rd party packages that are believed to be compatible with the C++ Toolkit.

Table 1. Compatible Versions of Third Party Packages

Package	FreeBSD 32	Linux 32	Linux 64	Mac OS X	SunOS x86	SunOS SPARC	Windows ^a (32, 64 bit)
BerkeleyDB	4.4.20, 4.6.21.1	4.6.21.1	4.6.21.1	4.5.20	4.6.21.1	4.5.20	4.6.21.NC

Boost Test	1.35.0	1.41.0	1.41.0	1.35.0	1.35.0	1.35.0	1.42.0
FastCGI	-	2.4.0	2.4.0	-	2.1	2.4.0	2.4.0
libbzip2	current	current	current	current	current	current	1.0.2
libjpeg	current	current	current	current	current	current	6b
libpng	current	current	current	current	current	current	1.2.7
libtiff	current	current	current	current	current	current	3.6.1
libungif ^b	current	current	current	current	current	current	4.1.3
LZO	-	2.02	2.02	2.02	2.02	2.02	2.02
PCRE							7.9
SQLite3	3.6.14.2	3.6.14.2	3.6.14.2	-	-	-	3.6.14.2
Sybase	-	12.5-32bit	12.5-64bit	-	12.5.1	12.0-EBF9209	12.5
zlib	current	current	current	current	current	current	1.2.3

^a Applies to MSVC 9.0

^b Only 4.1.3 version is supported currently.

Table 2. Currently Supported/Tested Versions of Third Party Packages

Package	Versions expected to work (obtained by build-environment inspection in some cases)	Versions known to work (used in-house on any platform)
BerkeleyDB	4.3.0 or newer	4.5.20, 4.6.21, 4.6.21.1, 4.7.25
Boost Test	1.35.0 or newer	1.35.0, 1.40.0, 1.41.0, 1.42.0, 1.45.0
FastCGI	All versions	2.1, 2.4.0
libbzip2	All versions	1.0.2, 1.0.3, 1.0.5, 1.0.6
libjpeg	All versions	6b
libpng	All versions	1.2.5, 1.2.7, 1.2.8, 1.2.10, 1.2.42, 1.2.44, 1.4.3
libtiff	All versions	3.6.1, 3.8.2, 3.9.2, 3.9.4
libungif	All versions	4.1.0b1 (libungif), 4.1.3, 4.1.6 (giflib)
libxml2	All versions	2.7.3, 2.7.6
libxslt	All versions	1.1.24, 1.1.26
LZO	2.x	2.02, 2.03
PCRE	All versions	4.4, 6.6, 7.8, 7.9
SQLite3	3.6.6 or newer	3.6.14.2
Sybase	All versions	12.0-EBF9209, 12.5.0.6-ESD13, 12.5.1, 12.5.1.10-ESD26
zlib	All versions	1.2.1, 1.2.3, 1.2.3.3

For Mac OS X and UNIX OS's, the user is expected to download and build the 3rd party packages themselves. The release's package list includes links to download sites. However,

the user still needs a list of the 3rd party packages and which versions of them are compatible with the release.

To facilitate the building of these 3rd-party libraries on Windows, there is an archive that bundles together source code of the 3rd-party packages, plus MSVC "solutions" to build all (or any combination) of them.

Table 3. Versions of Third Party Packages Included in the FTP Archive_

Package	Depends On	Included Version ^a
BerkeleyDB		4.6.21.NC
Boost Test		1.42.0
libbzip2		1.0.2
libjpeg		6b
libpng	zlib 1.2.3	1.2.7
libtiff	libjpeg 6b, zlib 1.2.3	3.6.1
libungif		4.1.3
LZO		2.02
PCRE		7.9
SQLite3		3.6.14.2
zlib		1.2.3

^a Applies to MSVC 9.0

Build

For guidelines to configure, build and install the Toolkit see [here](#).

New Developments

HIGHLIGHTS

- Added LDS2 (Local Data Storage v.2) which is based on SQLite3, has new features and better performance. Also implemented LDS2 data loader to use LDS2 from the Object Manager.
- XmlWrapp –this convenient XML handling API has been mostly finished (and even polished).
- Implemented tunneling and authorization of HTTP connections and tunneling of secure sockets, through HTTP proxies.
- CFormatGuess now allows distinguishing between GTF, GFF3, and GFF2. It's a possibly breaking change. For more details see below.
- Implemented major parts of CFeatTree, the class to organize features defined on a biological sequence into a hierarchy that reflects their parent-child relationships (based on the feature subtypes).

CORELIB

- Implemented locale-independent conversion of string to double and back; changed core libraries to use it.

- NStr::Justify() -- for formatting of paragraphs of text.
- CNcbiApplication -- make FindProgramExecutablePath static, and more robust; add a static higher-level GetAppName method. Look for global configuration files in more cases.
- CMetaRegistry::FindRegistry -- new method exposing the logic determining which file (if any) to load.
- CEnvironmentCleaner -- new class to discard unwanted environment variables.
- CFileIO -- back to original behavior: do not close the file handle if it's assigned via SetFileHandle().

SERIAL

- Serialization of AnyContent data objects -- fixed to recognize and properly process attributes in their values.
- Corrected the reading of XML data to assign to an element default value when it has no content.
- Added support for sequences of elements, where the element has a default value.

DATATOOL

- Corrected code generation of:
 - CHOICE data objects;
 - binary data types with attributes.
- Corrected conversion of double type values to preserve more significant digits.

CONNECT

- Added keepalive socket option (fSOCK_KeepAlive).
- Added NCBI connectivity test (CConnTest).

UTILITES

- g_FindDataFile -- New function for locating data files in (configurable) standard locations.
- CChecksumStreamWriter - new class to compute checksum of the data written to a stream.
- g_GZip_ScanForChunks() - new API, to query compressed stream positions. Added implementation for getting positions for separate gzip-files inside concatenated gzip file.
- Added compression/decompression stream manipulators (include/util/compress/stream_util.hpp).
- CFormatGuess (util/format_guess.{h/c}.pp) updated, with a possibly breaking change. The purpose of this is to allow CFormatGuess to distinguish between GTF, GFF3, and GFF2. Currently it lumps all of those formats into a one 'eGtf' value. The old 'eGtf' value (3) is being replaced with 'eGtf_POISONED', and will not be returned again. The new value for 'eGtf' (21) will mean a file that should be read with CGtfReader (objtools/readers/gtf_reader.hpp). The new value 'eGff3' (22) is for files meant to be read with CGff3Reader (objtools/readers/gff3_reader.hpp), and 'eGff2' (24) is for files meant to be read with CGff2Reader (include/objtools/readers/gff2_reader.hpp).

BIO-OBJECTS

- `CBioseq::GetNonLocalId` -- New method to help place sequences imported from FASTA files with range specifications in more context; wrapped by `CBioseq_Handle::GetNonLocalIdOrNull` (likewise new).
- `CSeq_id::IdentifyAccession` -- Implement or improve recognition for more prefixes (GA, HH, HI, HO-HU, JA-JO, EAAA-EZZZ, and IAA-IZZ, some of which correspond to the new possibility of DDBJ TPA WGS data) and mixed-in TPA protein accessions (mostly from EMBL, but some from GenBank too).
- Distinguish WGS master accessions by a new flag bit. Relax over-strict PDB recognition logic.
- `CSeq_id::IsValidLocalID`, `CSeq_id::ParseIDs` -- New functionality for working with plain-text sequence identifiers, factored out of `CFastaReader` and generalized somewhat.
- `SSeqIdRange` -- New type (complete with parser and on-the-fly "iterator") for working with Seq-id ranges, as present in some FASTA define source modifiers.

BIO-TOOLS

- `CFastaOstream` -- Optionally accept custom titles for single sequences. Tag negative-strand ranges with leading 'c's.
- `CFastaReader` -- Support negative-strand ranges and Sequin's compact define-style gap syntax ("`>?N`" where N is a number; or "`>?unk100`").

COBALT

- Added command-line option `-num_domain_hits` that limits number of conserved domains per sequence used in computing alignment constraints.

Phylogenetic trees

- Added higher level interface for computing phylogenetic tree from sequence alignments (for example BLAST and COBALT results). Class `CPhyTreeCalc` computes phylogenetic tree, and `CPhyTreeFormatter` prints the tree in Newick and Nexus format.

BIO-OBJECT LIBRARIES

- Implemented `CheckNumRows()` and other methods for sparse alignments.
- To reduce memory footprint: added read-hooks to reduce memory used by alignments after deserialization; Na-strand now uses one byte of memory where possible; `Score.value` choice is now embedded in `CScore`.
- Capitalize accession in `CSeq_id::GetLabel()`.

BIO-OBJECT MANAGER

New functionality:

- Added getter methods for boolean fields in `CTableFieldHandle`.
- Added `GetBestGeneForFeat()` based on `CFeatTree`.
- Implemented `GetBestOverlappingFeat()` on `CFeatTree`.
- Added `fast CScope::GetTaxid()`.
- Implemented bulk loading for `acc/ver`, `gi`, `label`, and `taxid`.

- Added zero-length gaps check to CSeqMap and CSeqVector.
- Implemented GetLength() and GetCoverage() for bond locations.

Improvements:

- Added helper method to fill CFeatTree on location.
- Sped up mapping of simple CSeq_loc_mix locations in CFeat_CI.
- Stricter sorting of features in CFeat_CI to avoid ambiguities.
- CSeq_feat_Handle getters now work with Seq-table features too.
- Seq-table features now support multi-level user fields.
- Non Seq-feat Seq-tables are now recognized even if located in split chunk.
- Sped up CBioseq_Handle::AddId().
- Optimized CScope::AttachXxx().
- Support split of named annotation.
- CSeqVector and CSeqVector_CI's CanGetRange() now return false instead of throwing an exception.
- Allow to specify how to deal with existing handles in ResetHistory().
- Optimized re-parenting if more features are added to CFeatTree.
- Added possibility to debug CScope creation/deletion.
- Many changes to the C++ cleanup functionality to imitate the cleanup functionality which already exists in C. There is still more work to be done with BasicCleanup, but significant progress has been made. Little work has been done for ExtendedCleanup as of yet.
- CSeq_loc_Mapper can now be initialized with a GC-Assembly.

Bug fixes:

- Fixed mapping of mix locations on minus strand in CFeat_CI.
- Many fixes in the way CFeatTree links features.
- Several thread-safety fixes.
- Fixed typo preventing adding aligns and graphs to CSeq_annot_EditHandle.
- Safeguard against exceptions when sorting features in CFeat_CI.

GENBANK DATA LOADER

New functionality:

- Registered HPRD external annotations.
- Added optional exclude_wgs_master param in pubseqos/pubseqos2 readers.
- Implemented bulk loading for acc/ver, gi, label, and taxid.
- Added CGBDataLoader::CloseCache().

Improvement:

- Use bulk loading requests in CScope::GetBioseqHandles().
- Separate reader statistics by type of loaded blobs.
- Added timestamp to GenBank debug messages.

- Use IConnValidator for opening PubSeqOS connections.
- Added split-version to chunk requests and chunk subkeys in GenBank cache to avoid using wrong chunks when blob split state is changed in ID.
- Added secondary less confusing param names for open timeout.
- Do not multiply retry count by number of connections.

OBJECT MANAGER TEST AND DEMO APPLICATIONS

- id2_fetch_simple -- added -id options for arbitrary Seq-id's.
- test_bulkinfo -- new test application.

FASTA

- C++ feature table functionality has been made more functional such as for part of the BankIt project.

asn2flat utility

- Huge number of changes to flatfile formatter to bring it much closer to release-ready state (possibly release ready at this point, although some relatively minor issues remain).

XMLWRAPP

- Fixed segmentation fault in case of taking a reference to XPath expression running results.
- Added helpers to get public ID, system ID and DTD name for external and internal subsets.
- Added methods to lookup node attributes.
- Fixed execution of XPath expression: it now starts from the given node.
- Fixed searching attributes (including default) when a namespace is provided.
- Added ability to run XPath expression without necessity to register namespaces explicitly.
- Added ability to provide containers for collecting errors and warnings while parsing documents.
- Added ability to modify values and namespaces of node's default attributes.
- Added ability to test if an attribute is default.
- Added ability to insert or remove attributes while taking into account their namespaces.
- Added ability to strip XML declaration when a document is saved.

WindowMasker

- Added a new input format, "seqids"; with this input format, the input is a file containing a sequence id on each line, and the algorithm uses the Bio-Object Manager to look up the sequences.
- Added a new class CWinMaskConfig, for storing all the WindowMasker configuration parameters. The class can be used to add the needed command-line arguments to CArgDescriptions, and then get the configuration parameters from the command-line arguments.

BUILD FRAMEWORK (UNIX)

- Interpret command-line specifications of APP_PROJ or LIB_PROJ as a cue to clear out other *_PROJ settings not also provided there. (Requires GNU Make; builds with Sun make continue to work as before.)
- Supply more targets in subdirectories: *_f (using local flat makefiles produced on demand, ignoring dependencies on other parts of the tree), *_fd (wrapping the top-level Makefile.flat), clean_sources and purge_sources.
- Configure and its convenience scripts (compilers/unix/*.sh):
 - Noteworthy new flag --without-3psw – to not use with any 3rd-party software.
 - Added a check for GLEW.
 - Improved checks for Boost and OpenGL.
- Support specifying run paths on Darwin (Mac) systems with modern toolchains.

BLAST

- On Darwin (Mac OS X), build only for Intel processors even in otherwise universal builds due to a PowerPC toolchain limitation.
- Added support for retrieving NCBI Taxonomy IDs for which WindowMasker support is available.
- Allow the specification of a query sequence along with multiple sequence alignment file in psiblast.
- Added database hard-masking support.
- Added database soft-masking for translated searches.
- Added support for btop (BLAST traceback operations) and query and subject length in the tabular report.
- Command-line applications -- allow psiblast to search multiple queries, added optional -input_type for makeblastdb
- Allow use of best hit and XML in blast2sequences mode.
- Improved formatting performance for remote searches.
- makembindex can now build masked MegaBLAST index directly from a BLAST nucleotide database using the masking information stored in the BLAST database. This is accomplished by new command line option -db_mask to makembindex. The option accepts the integer id of the filtering algorithm supported by the BLAST database. The option can only be applied in conjunction with -ifformat blastdb.
- To assist a user in finding out the numeric ids of filtering algorithms supported by a BLAST database, the flag -show_filters is introduced. Applying the flag with -ifformat blastdb and BLAST database as an input causes makembindex to output a list of available filtering algorithms and exit.

APPLICATIONS

NETCACHE

- NetCache is reworked to include the following features:
 - better management of disk space;
 - lock-less work with blobs, versioning is used instead;
 - multi-port listening and per-client settings differentiating.

- NetCache and ICache APIs:
 - Use Uint8 everywhere for blob size.
 - Allow partial blob retrieval.
 - Introduced blob password protection; empty passwords are treated as no password.
- Worker node APIs:
 - New parameter for terminating the worker node if its memory consumption exceeds the specified limit (parameter "total_memory_limit").
 - New parameter for terminating the worker node if its run time exceeds the specified limit (parameter "total_time_limit").

GRID APPLICATIONS

- netscheduled
 - Fixed a bug that caused no reply to the queue deletion command.
- remote_app
 - New configuration parameter ("tmp_dir") to control how temporary directory name is generated - to reduce its length.
 - Log blob writing error.
- netcache_control
 - Allow partial blob retrieval.
 - New command -remove to delete blobs by their ids.
 - New parameter -auth to specify authentication string to use.
 - New commands -reconf and -reinit for use by NetCache administrators.
- netschedule_control
 - Enabled compatibility mode to make netschedule_control work with older worker nodes.
- cgi2rcgi.cgi
 - Do not create an empty NetCache blob as a placeholder for the progress message.
 - Log Grid errors that are reported to the user.
 - Allow spaces in the job ID parameter.
 - Support output of the job status information in JSON format.
 - Allow custom HTML templates to be defined for GRID errors and other events.
 - Added no-cache HTTP headers to avoid caching of intermediate results.
- ncfetch.cgi
 - New parameter to access password-protected blobs.
 - Interpret extra parameter "filename" as a file name for the downloaded file.

Documentation

Location

The documentation is available online as a searchable book "The NCBI C++ Toolkit": <http://www.ncbi.nlm.nih.gov/books/NBK7160/>.

The C++ Toolkit book also provides PDF version of the chapters. 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 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.

Supported Platforms (OS's and Compilers)

- [UNIX](#)
- [MS Windows](#)
- [Mac OS X](#)
- [Added](#)
- [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 and some were added. 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.

In cases where multiple versions of a compiler are supported, the mainstream version is shown in **bold**.

UNIX

Table 3. UNIX OS's and Supported Compilers

Operating System	Architecture	Compilers
SuSE-9.x (LIBC 2.3.5)	x86-32	GCC 3.4.2 , 4.1.2 ^a , 4.2.3 ^a , 4.3.3 ^a , 4.4.2 ^a , 4.6.0 ^a ICC 10.1 , 11.1 ^a
SuSE-9.x (LIBC 2.3.5)	x86-64	GCC 4.0.1 , 4.1.2 ^a , 4.2.3 ^a , 4.3.3 ^a , 4.4.2 ^a , 4.6.0 ^a ICC 10.1 , 11.1 ^a
CentOS 5.4 (LIBC 2.5)	x86-64	GCC 4.4.2
CentOS 5.4 (LIBC 2.5)	x86-32	GCC 4.4.5
Ubuntu 9.04 ("jaunty") (LIBC 2.9)	x86-32	GCC 4.3.3
Solaris 10	SPARC	GCC 4.1.1 ^b Sun Studio 12 (C++ 5.9) , Sun Studio 12 Update 1 (C++ 5.10) ^a Oracle Studio 12.2 (C++ 5.11) ^a

Solaris 10	x86-32	GCC 4.2.3 Sun Studio 12 (C++ 5.9) , Sun Studio 12 Update 1 (C++ 5.10) ^a Oracle Studio 12.2 (C++ 5.11) ^a
Solaris 10	x86-64	Sun Studio 12 (C++ 5.9) , Sun Studio 12 Update 1 (C++ 5.10) ^a Oracle Studio 12.2 (C++ 5.11) ^a
FreeBSD-8.0	x86-32	GCC 4.2.1
Darwin 8.x, 9.x (Mac OS X 10.4.x, 10.5.x)	Native, Universal	GCC 4.0.1

^a some support

^b 32-bit only

MS Windows

Table 4. MS Windows and Supported Compilers

Operating System	Architecture	Compilers
MS Windows	x86-32	MS Visual C++ 2008 (C++ 9.0), 2010 (C++ 10.0) – nominal support only. NOTE: We also ship an easily buildable archive of 3rd-party packages (including NCBI C Toolkit) for this platform.
MS Windows	x86-64	MS Visual C++ 2008 (C++ 9.0), 2010 (C++ 10.0) – nominal support only. NOTE: We also ship an easily buildable archive of 3rd-party packages (including NCBI C Toolkit) for this platform, although not all libraries are available for x86-64 architecture.
Cygwin 1.5.25	x86-32	GCC 4.3.4, GCC 3.4.4 - nominal support only.

Mac OS X

Table 5. Mac OS and Supported Compilers

Operating System	Architecture	Compilers
Mac OS X 10.5, MacOS x 10.6	Native (PowerPC <u>or</u> x86-32 <u>or</u> x86-64)	Xcode 3.0 - 3.2.6
Darwin 8.x, 9.x, 10.x	Native (PowerPC <u>or</u> x86-32 <u>or</u> x86-64), Universal (PowerPC <u>and</u> x86-32)	GCC 4.0.1 GCC 4.2.1 (only available under Darwin 10.x)

NOTE: the correspondence between Darwin kernel versions and Mac OS versions:

Darwin 8.x = Mac OS 10.4.x

Darwin 9.x = Mac OS 10.5.x

Darwin 10.x = Mac OS 10.6.x

Added Platforms

Table 6. Added Platforms

Operating System	Architecture	Compilers
MS Windows	x86-32, x86-64	MS Visual C++ 2010 (C++ 10.0) – nominal support only.
CentOS 5.4 (LIBC 2.5)	x86-32	GCC 4.4.5

Solaris 10	SPARC	Sun Studio 12 (C++ 5.9) , Sun Studio 12 Update 1 (C++ 5.10) ^a Oracle Studio 12.2 (C++ 5.11) ^a
Solaris 10	x86-32	Sun Studio 12 Update 1 (C++ 5.10) ^a Oracle Studio 12.2 (C++ 5.11) ^a
Solaris 10	x86-64	Sun Studio 12 Update 1 (C++ 5.10) ^a Oracle Studio 12.2 (C++ 5.11) ^a
Mac OS X 10.5, MacOS x 10.6	Native (PowerPC <u>or</u> x86-32 <u>or</u> x86-64)	Xcode 3.2.3 - 3.2.6

^a some support

Discontinued Platforms

Table 7. Discontinued Platforms

Operating System	Architecture	Compilers
MS Windows	x86-32, 64	MS Visual C++ 2005 (C++ 8.0)
FreeBSD-6.1	x86-32	GCC 3.4.6

Caveats and Hints

GCC 3.4.x

- 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).
- GCC 3.4.4 has 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 May 31, 2011.