



# Greenleaf ArchiveLib 2.12

## Data Compression, Archiving, & Encryption for Windows & DOS

### Easily the Best...

- Greenleaf ArchiveLib is unique: It starts with PKZip 2.x compression and decompression engines, adds full PKZip archive file capability, and tosses in encryption for your Windows and DOS applications.
- There are three compression engines and you can compress and archive to/from memory, file, serial port.....etc.
- Unlike PKWare libraries, ArchiveLib is fully compatible with PKZip. Adjustable compression parameters.
- 100% C++ class library plus thin C shell = two APIs.
- Three compression / decompression engines:
  - PKZip 2.x compatible (Deflate engine)
  - Greenleaf proprietary lossless engine, dictionary based, 5-level.
  - Copy engine for those files too small to bother compressing.
- Use our archival templates (one is PKZip format) or brew your own with the tools included.
- Supports Windows XP, 2000, 9x, NT 4, Me, 2003, and 3.z plus DOS and Extended DOS (using Borland C++ 5.02 or Visual C++ 1.52). Also offers support for OS/2 Presentation Layer.
- Several types of progress displays for file compression status.

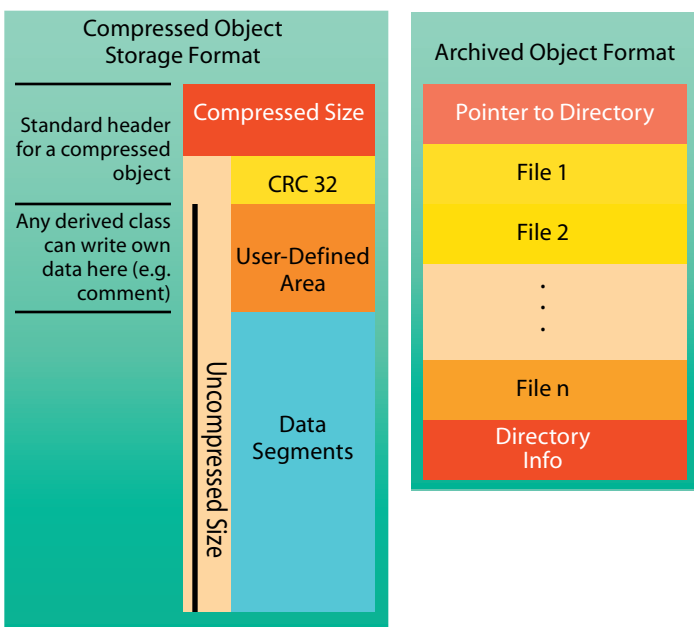
- Includes recursive subdirectory search engine with wildcard expansion.
- Supports Visual C++ 5,6,7.1, Borland C++Builder 3,4,5, and 6, Visual Basic 5,6, and 7.1, Visual FoxPro 5,6, Borland Delphi 3-6, PowerBuilder 6, 6.5, and 7, and VBA for Microsoft Office application development.
- Libraries include DLL, release & debug, static libraries, and BUILD utility so you can recompile any library if need be.
- Dozens of example programs included, all with source,
- FREE source, no ROYALTIES, FREE web / email support.
- 40 classes, over 250 public methods.
- Granular coding allows significant reduction of application footprint.

### Top Level Functional Architecture



### Comparison: ArchiveLib 2.x vs. PKWare Libraries

PKZip 2.x compatibility	yes	No
C++ classes	> 25	None
Function Count	> 250	3
Same product supports Win32, Win16, OS/2, DOS, Ext'd DOS	Yes	No, must buy separate products, avg. \$350 each
Visual Basic, Pascal, Delphi	Yes	No
Archiving Functions	Yes	No
Compresses to / from memory	Yes	No unless you write glue function
Compresses to / from file(s)	Yes	Yes
User interface support classes	5	None
Support for Windows listboxes	Yes	No
Comments per file, archive	Yes	No
Archive entry list processing	Yes	No
Private data area in compressed obj	Yes	No
Buffered file I/O for max thrupt	Yes	No
File rename / delete / backup	Yes	No
Subdirectory recursion / wildcards	Yes	No
Save file date / time / length / attributes	Yes	No
Time / Date manipulation functions	Yes	No
Named memory buffers	Yes	No
Compression progress meter	Yes	No
Long filename support	Yes	Yes
CRC-32 calculation functions	Yes	No
FREE source code included	Yes	No
Debug library versions included	Yes	No
Example programs	> 35	1
Windows Help documentation	Yes	No



## Greenleaf ArchiveLib 2.1

Crafted by internationally recognized authorities on data compression, ArchiveLib is a full-featured tool that allows you to add archival storage to your applications. Features include three different compression engines, classes that allow you to produce and read PKZip files, encrypt portions of an archive, and support Windows, DOS, OS/2, and save you so much time your project will come in early, under budget, and excel in robustness and capability. ArchiveLib is a must-have.

### What Others Say

Often, having been led by the perception that PKWare libraries must be the best, customers will turn to Greenleaf and find to their amazement that ArchiveLib, not the PKWare libraries, are PKZip compatible. How this can be is not the point—the point is that with ArchiveLib you not only get ZIP compatibility but much more. People say our product is complete—because what good, really is compression without archiving? People tell us they like having encryption at their fingertips and having the ability to create special purpose archives structured their way rather than “the way it’s always been.”

### Supports Many Platforms

With ArchiveLib, you can write applications for Windows XP Pro, 2000, 2003, Me, 9x, 3.x, NT, and OS/2. ArchiveLib is a C++ class library. Using Visual C++ 1.52 or Borland C++ 5.02 you can also target MSDOS and extended DOS.

Compilers supported currently include Visual C++ 5, 6, and 7.1, Visual Basic 5, 6, and 7.1, C++Builder 1 thru 6, Borland C++ 5.02 (we can get you a free copy) is needed for targeting DOS or Extended DOS and OS/2. Borland’s Delphi is also supported to a certain extent, and on special request we can get you libraries for Watcom C++ 10 which people are using with Open Watcom. We cannot adequately support use of the product with open source software.

### Documentation

Most software is designed, tested, then the documentation is written. ArchiveLib’s development and documentation teams worked as an organic entity so that there are tips in the docs that a tech writer would not think to add—and code that writers suggested.

We ship a Windows Help manual with ArchiveLib; in addition there are numerous examples for each platform—source code that you can compile. It turns out that many beginner users find our examples do what they want, so they have little work to do.

### Commitment & Growth

Greenleaf ArchiveLib was developed at a time when the only data compression programmer tools on the market were the PKWare libraries—which weren’t compatible with ZIP format and provided no help for archiving. In that context, ArchiveLib was an instant success.

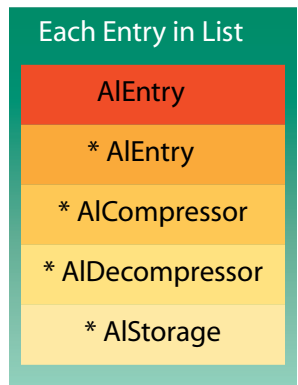
We were quite fortunate to have Mark Nelson on board at the time—Mark is an internationally known expert on data compression and now co-authors with Jean-loup Gailly (author of the ZLIB compression software and another internationally known expert) the book “The Data Compression Book.” which is in at least the second printing and has been translated to a number of languages. Mark’s intuitive sense and drop-dead C++ code-slinging expertise were invaluable in bringing this product to market. We plan to keep it there.

### What is an Archive?

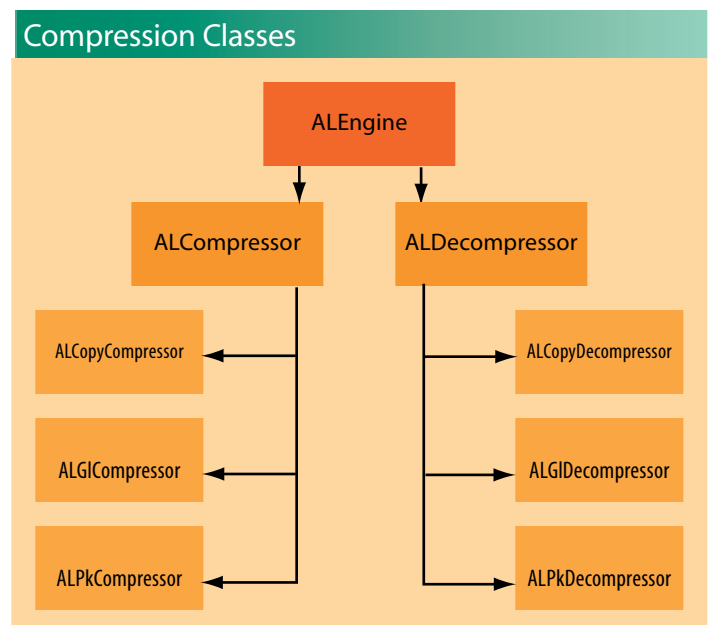
Each archive is an instance of either the ALGIArchive or AIPkArchive class.

As shown in the two illustrations to the right, each archive object is further described by an instance of the ALEntryList class. ALEntryList points to the first entry (object of class ALEntry) in the archive.

Each ALEntry, in turn, points to the next entry, the compressor and decompressor engines to use, and to storage (memory or file) for the entry. Typically, therefore, you can build an archive whole in memory of an unlimited number of entries, or from one or more entries targeted to a file via the class derived from ALStorage.



The illustration below shows the relationship between the various compression classes and, at the top, an abstract ALEngine class that provides the infrastructure for the derived compressor and decompressor engines shown here.



# Some of ArchiveLib's Classes and their Members

---

## ALStorage class

---

Open	Create
Clone	Close
Compare	Delete
Seek	Tell
Rename	UnRename
RenameToBackup	IsOpen
WritePkShort	WritePkLong
WriteString	ReadPkShort
ReadPkLong	WritePortableShort
WritePortableLong	ReadPortableShort
ReadPortableLong	ReadCopyright
LoadBuffer	FlushBuffer
InitCrc32	GetSize
GetCrc32	YieldTime

## ALMemoryBase, ALWindowsMemory, ALHugeMemory classes

---

Open	Create
Close	LoadBuffer
FlushBuffer	Seek
Rename	UnRename
Delete	RenameToBackup

## ALFile class

---

Open	Close
Create	Clone
Seek	Rename
UnRename	RenameToBackup
LoadBuffer	FlushBuffer
MakeTempName	

## ALCompressor and ALDecompressor

---

Clone	Compress
Decompress	

## ALGCompressor, ALGDecompressor

---

Clone	CompressionLevel
Compress	Decompress

## ALPkCompressor, ALPkDecompressor

---

Clone	Compress
Decompress	

## ALCopyCompressor, ALCopyDecompressor

---

Clone	Compress
Decompress	

## ALCompressedObject class

---

Insert	Extract
ReadHeaderData	WriteHeaderData
ALCompressionEngine	Compress
Decompress	WriteEngineData
ReadEngineData	

## ALEntry class

---

InsertBefore	SetMark
ClearMark	GetCompressedSize
GetComment	SetComment
Duplicate	GetNextEntry

SetMarkState	CompressionRatio
GetCrc32	GetMark

## ALEntryList class

---

GetFirstEntry	SetMarks
ClearMarks	ToggleMarks
DeleteUnmarked	SetMarksFromListBox
FillListBox	MakeEntriesFromListBox
AddWildcardFiles	UnmarkDuplicates

## ALArchive, ALGIArchive, ALPkArchive

---

Create	Append
Extract	Delete
GetComment	SetComment
ReadDirectory	WriteDirectory
GetVersion	GetStorageObject
FillListBox	

## ALEngine class

---

WriteEngineData	ReadEngineData
-----------------	----------------

## ALName class

---

GetName	GetOldName
GetSafeName	GetSafeOldName
ChangeExtension	ChangeTrailingChar
StripFileName	StripPath
WildcardMatch	

## ALWildcardExpander class

---

GetNextFile

## ALFileAttributes class

---

PackedAttributes	UserWrite
SetFromPackedAttributes	UserExecute
SetFromDosAttributes	GroupRead
ReadOnly	GroupWrite
System	GroupExecute
Hidden	OtherRead
Archive	OtherWrite
UnixBitsPresent	OtherExecute
UserRead	Directory

## ALTimeDate class

---

Valid	SetTimeDate
ToJulian	GetTimeDate
FromJulian	CreateUnixTime
GetUnixTime	SetFromUnixTime
GetDosTime	SetFromTmStruct
GetDosDate	

## ALMonitor class

---

Progress	ArchiveOperation
----------	------------------

## ALWindowsMessage, ALOs2Message, ALSpinner, ALBarGraph classes

---

Progress	ArchiveOperation
----------	------------------

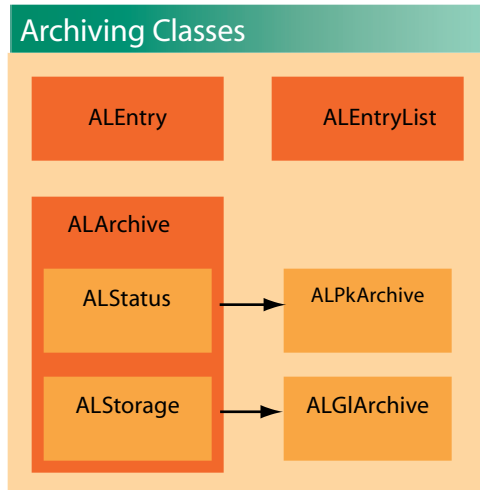
## ALStatus class

---

SetError	GetStatusString
GetStatusCode	GetStatusDetail

The implementation of archive building and reading classes, as shown below, starts with `ALEntry`, which binds everything needed to store a single file or buffer of data in an archive. One or more `ALEntry` instances are collected into an `ALEntryList` object and then the abstract `ALArchive` class takes over; derived classes create a ZIP compatible archive or a Greenleaf proprietary archive. Along the way, you can arrange for many options choosing which file attributes to store, time and date formats to use, and you can add a comment that applies to a single entry and/or one that applies to the entire archive.

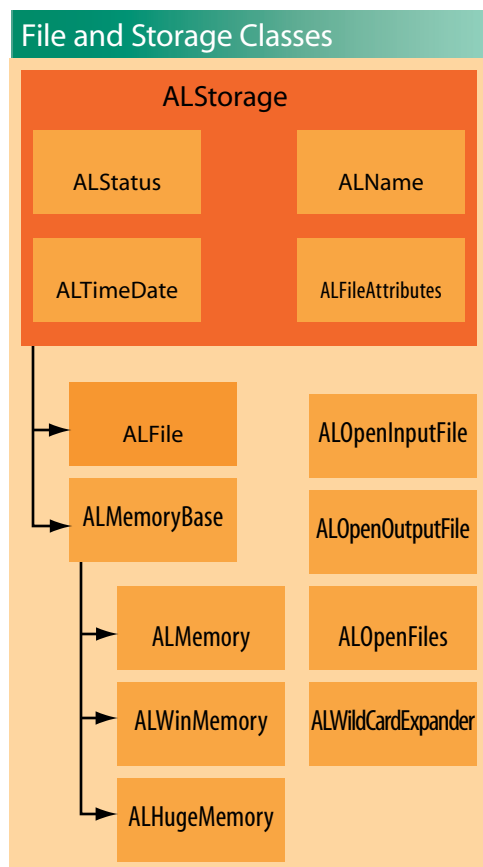
While data is still in memory, you can invoke the `GLCrypto` methods to encrypt entries in the archive.



## ALCrypto class

StartEncrypt  
StartDecrypt  
ClearError

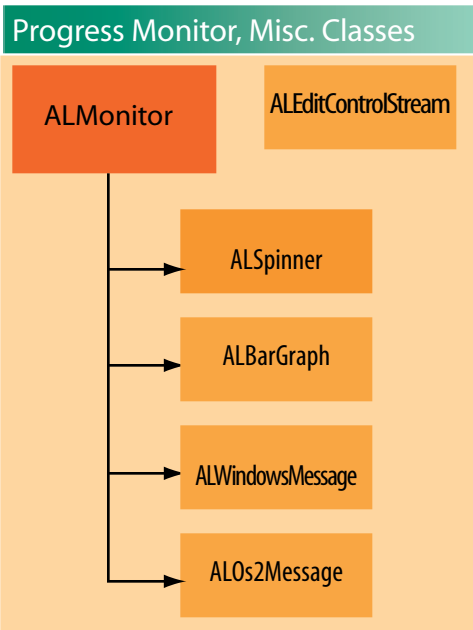
EndEncrypt  
EndDecrypt



## Memory, File, Serial Port, you can do it...

The illustration above shows classes used internally to work with memory buffers and files. Note the inclusion of a separate class `ALWildcardExpander` which is a subdirectory recursion engine that is aware of wildcard filename specifications.

As illustrated, ArchiveLib can build entire archive complexes in memory and, if desired, ship them to things other than files. For example, some customers are using this technique to send whole archives to another machine via a USB connection or a serial port.



## Monitoring Compression Progress

At left the illustration shows a collection of classes related to monitoring and displaying progress of compression.

You can choose the simple "spinner" display, the ubiquitous bar graph, a message box tailored for either Windows or OS/2.

As with most ArchiveLib parts, this group of classes starts with the abstract `ALMonitor` class from which you can derive your own method of display. Using one of the many graphical type libraries available, the choice is yours.



**Call Greenleaf Software Now:**

**1-877-218-7347**

**9550 Forest Lane  
Suite 309  
Dallas, TX 75243**

**+1 214-349-3005  
+1 214-349-3650 (FAX)**

**[www.GreenleafSoft.com](http://www.GreenleafSoft.com)**

**[sales@GreenleafSoft.com](mailto:sales@GreenleafSoft.com)**

- ❑ Greenleaf ArchiveLib supports Microsoft Visual C++ 5, 6, and 7.1, Borland C++Builder 3...6, Delphi 3-6 and Visual Basic 5 and 6.
- ❑ Develop applications, DLLs, etc. for Windows 9x, Me, 2000, XP Pro, and Server 2003.
- ❑ Many compilable example programs for each supported platform included, all with source for Windows.
- ❑ Coded for optimum granularity and speed.
- ❑ Product includes Windows Help documentation
- ❑ FREE Web and email based support with telephone support as needed.
- ❑ We accept MasterCard, VISA, American Express and Discover. Terms available upon credit approval.
- ❑ Did we forget to say THERE ARE NO ROYALTIES? Well, there are none.

**We do the grunt work...**