

RAR Component – Documentation

First of all: sorry for my poor English!

Contact

Author: Philippe Wechsler, Switzerland (Bern)

Web: www.PhilippeWechsler.ch

Mail: contact@PhilippeWechsler.ch

First release: 12.09.2008

What is RAR Component?

RAR Component is a library for Delphi to open, extract and test rar archives using the “unrar.dll”. This is not the first library for Delphi to unrar archives, but it’s the first that offers you all possible functionality from the “unrar.dll”!

- Open, Extract and Test all rar archives (new with support for Unicode filenames in d2009)
- Display the correct progress state (there are a lot of unrar libraries that do not show the progress state of the actual file, only the total progress state)
- Show all archive information’s:
 - * Rar format version
 - * Total files
 - * Dictionary size
 - * Solid flag
 - * Files or archive encrypted
 - * Host OS
 - * Compressed/uncompressed size
 - * File/archive comment present + comment itself
 - * Multivolume flag
 - * Signed flag
 - * Locked flag
 - * Recovery data flag
 - * Extract all or only custom files
 - * Extract files with/without restoring folders
 - * Event for “password” or “next volume” required and replace request
 - * ReadMultiVolumeToEnd
 - * Easy to use
 - * Free!

Requirements

The library was written and tested in Delphi 2007 and 2009, but it might run also in other Delphi versions. Please contact me if the library works in your Delphi version.

License

You can use my code for free; also for commercial projects (maybe you can send me a free copy of your software or donate me something?). If you find bugs or if you modify/extend the code please write me your changes. Your application should include a short message that you use code written by Philippe Wechsler and a link to my site (www.PhilippeWechsler.ch), also if you changed my code.

Please note also the license for the “unrar.dll” library:

The unrar.dll library is freeware. This means:

1. All copyrights to RAR and the unrar.dll are exclusively owned by the author - Alexander Roshal.

2. The unrar.dll library may be used in any software to handle RAR archives without limitations free of charge.

3. THE RAR ARCHIVER AND THE UNRAR.DLL LIBRARY ARE DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS SOFTWARE.

Thank you for your interest in RAR and unrar.dll.

Alexander L. Roshal

Install and use the library

First you have to install the library. If you have installed already an older version you have to remove it. Please note that the library was designed for Delphi 2007 and 2009, so it might not run in other versions. If it doesn't work please contact me to help you!

- Extract all files into any custom folder
- Add this Folder to the Delphi's library (Tools->Options->Library – Win32->Library path)
- Open the file “RARComponent_d2007.dpk” (“RARComponent_d2009.dpk” for later Delphi versions)
- Go to the Project Manager and right-click “RARComponent.bpl” and select install

Open an archive

Opening an archive is very easy. Just call RAR.OpenFile(“name of the archive including path”). This function will return a Boolean value. If this value is true, all worked fine, if not there were errors while opening the archive.

Opening an archive will also list the content. So for each file in the archive the event RAR.OnListFile will be fired.

Test an archive

To check if an archive is damaged or not you can call RAR.Test. If the returned value is true, the archive is not damaged.

Extract an archive

There are multiple ways' to extract files from an archive. If you simply want to extract all files you can call RAR.Extract(“location where extracted files will be written”,True,NIL)

Path: this is the location where the files will be written to. It doesn't matter if this path is terminated by a “\” or not!

RestoreFolders: this value determines if folders in the archive will be written or not. Files will be written like this:

File in the archive: *"folder\file.bin"*

Path: *"c:\\"*

The file will be written to *"c:\folder\file.bin"* if RestoreFolders is set to True, else it's *"c:\file.bin"*

Default is true!

Files: specifies the files that will be extracted. If you want to extract all files, you just can set this to "NIL", else you have to set all files you want to extract (advanced feature!).

Abort the current operation

Just call RAR.abort!

Archive information

If you want to read archive information's you first have to open an archive. After that all archive properties are listed under RAR.ArchivenInformation .

FileName: archive name, including path

ArchiverMajorVersion+ArchiverMinorVersion: minimum rar version to extract files in the archive.

DictionarySize: size of the used dictionary

Encryption: True if files are encrypted

Solid: True if archive is solid

HostOS: system on that the archive was created

TotalFiles: amount of files compressed in the archive

CompressedSize+UnCompressedSize: compressedsize/uncompressed size of all files in the archive (in Bytes)

HeaderEncrypted: True if archive is encrypted (password required for viewing the file names)

ArchiveComment: True if archive comment is present

FileComment: True if file comments are present

Comment: archive comment itself

Signed: True is archive is signed by the author

Locked: True if archive is locked

Recovery: True if archive contains recovery data

SFX: True if the archive is an self-extracting archive

ReadMultiVolumeToEnd

This is an advanced feature of the library. By default this is set to False. If you open a multivolume archive there will be listed only the files that are in the part you opened and the file size and crc will be incorrect if the file is splitted into other parts of the archive. To avoid this you can set the "ReadMultiVolumeToEnd" property to true. If you now open the same archive all files will be listed (from all parts) with correct size and crc, but the library needs all parts of the archive. This makes only sense if all parts are in the same location! If not you have to insert all disk to open the archive!

Pro: all files of the multivolume archive will be listed with correct size and crc

Contra: to open a multivolume archive all parts are required.

DLLName

Specifies the path and name of the “unrar.dll” library. For example you could set this to “C:\unrar.dll” to use the library in this path. The Default is “unrar.dll”, but for this you have to place the library in the same folder like your application.

OnListFile

This event will be thrown whenever a file had been read. You can use this to add the file to a ListView. All file properties are available through “FileInformation”.

FileName & FileNameW : name of the file (including path). Use FileNameW in Delphi 2009 and newer to display unicode name

CompressedSize: size of the file in the archive

UncompressedSize: size of the decompressed file

HostOS: the os that compressed the file (either „DOS“, „IBM OS/2“, „Windows“ or „Unix“)

CRC: the checksum of the file

Attributes: file attributes, see the demo for more informations and handling

Comment: the file’s comment, currently not set by the dll

Time: last time the file had been modified

CompressionStrength: the compression method that was used :

48 = stored, no compression

49 = compressed, fastest compression

50 = compressed, fast compression

51 = compressed, normal compression

52 = compressed, good compression

53 = compressed, best compression

ArchiverVersion: version of the archiver that compressed the file

Encrypted: True if the file is encrypted

OnPasswordRequired

This event will be thrown when a password is required to process.

HeaderPassword: specifies if the password is required to encrypt the filenames (True) or for an file in the archive (False)

FileName: this is the filename of the file for that the password is required. This can be the archive itself (filenames encrypted/HeaderPassword) or a file in the archive.

NewPassword: you should set this to the password that is required

Cancel: you can set this if you don’t know the correct password and want to abort the current operation

OnNextVolumeRequired

This event will be thrown when the next part of a multivolume archive is required.

FileName: you should set this to the next part of the required part or leave it empty if you just need to replace a disk.

Cancel: you can set this to “True” if you want to abort the current operation.

OnReplace

This event will be thrown when the library is about to overwrite a file.

ExistingData,NewData: offers you information about the file that exists on your drive and the one in the archive

Action: set this to specify what to do

rrCancel: abort the current operation

rrOverwrite: overwrite the file on the drive with that in the archive

rrSkip: keep the file on the drive

OnProgress

This is a very useful event that will be thrown after every change of the progress state.

FileName: file that currently is in process

Progress: display how many % of the actual operation are done

FileBytesTotal: size to process of the actual file (in Bytes)

FileBytesDone: processed size of the actual file (in Bytes)