ICE - LHA Compression Library for Windows

ICE - the true compression/decompression library for Windows 3.1
ICE - creates archive files compatible with the industry-standard LHA 2.1x
ICE - interfaces with all Windows programming languages

Coming soon ICE:ZIP - the ZIP compression library for Windows

OVERVIEW

ICE.DLL provides the programmer with powerful functions to create and manipulate LHA 2.1x archives from within Windows applications. To include the ICE functionality in your programs add the language-specific file (e.g. ICE.BAS for VB users) to your project.

Add files to an archive with Freeze, either individually or based on a user-specified file pattern. Includes options to :

- preserve full or relative directory structure within the archive,
- recurse into directories,
- select the file(s) based on attributes,
- reset attributes after archiving,
- delete the file(s) after archiving

Extract files from an archive with the Thaw function, either individually or based on a user-specified file pattern. Includes options to :

- recreate stored directory structure (if it exists) or extract the file to another location,
- extract the file and optionally delete it from the archive,
- if the file already exists, either never overwrite, always overwrite, overwrite if newer or prompt the user for a decision

Technical information concerning the files compressed inside an archive can be retrieved into an array using the language-specific wrapper function ListArchiveContents around the ListArchive function. The following information is retrieved

path	the location of the file in the directory structure (if stored)
filename	the name of the compressed file
date	the date the compressed file was last accessed
time	the time the compressed file was last accessed
attributes	the date the compressed file was last accessed
original size	the original size of the file
compressed size	the compressed size of the file
ratio	the percentage redunction in file size
method	the compression method used
CRC	the file CRC value

Function reference

InitialiseICE	- register application with DLL
Freeze	- compress files into archive
Thaw	- expand files from archive
ListArchiveContents	- retrieve archive contents into an array
TestArchive	- test the integrity of an archive file
AddComment	- add a comment to the start of an archive
ShowComment	- retrieve the comment from an archive
MakeEXE	- make the archive self-extracting within Windows

InitialiseICE

The InitialiseICE function registers the calling application with the ICE library. This is used to set the parent window for any message boxes, and to optionally set the feedback window.

Syntax

Parameter	Description
hwndParent	Identifies the parent window of the calling program. If this parameter is NULL, any message boxes will have no parent window.
hwndDisplay	Identifies the window to receive feedback during file processing. If NULL, no feedback will be given.
fuOptions	Specifies the feedback to place in *hwndDisplay*. If NULL, no feedback will be given. This parameter can be a combination of the following (multiple values delimited by #)

Value	Meaning
ICE_PASSPERCENT	*hwndDisplay* will receive the a value representing how complete the compression/decompression of the current file is.
ICE_PASSFILENAME	*hwndDisplay* will receive the name of the current file being processed.

Comments

During calls to the functions Freeze and Thaw, the developer may wish to display some visual feedback of the function's progress. This function passes to *hwndDisplay* the text as defined by *fuOptions*. The Visual Basic programmer should pass the hWnd property of an invisible text box, and include code in the Change event to perhaps display a percent-complete bar (see the included VB example file). The C programmer should pass the handle to a window, then write code in the WM_SETTEXT message area.

Return value

This function has no return value.

FREEZE

The Freeze function adds files to an archive, creating the archive if one does not exist.

Syntax

VB: Declare Function Freeze Lib "ICE.DLL" (lpMask as String, lpLZH as String, fuOptions as Long) as Integer

Parameter **Description**

lpMask This is the file(s) to add to the archive. Standard DOS wildcard characters (* and ?) can be used to select multiple files. If this parameter is NULL, Freeze defaults to *.* i.e. all files. This parameter may also contain the starting directory, if no directory is given, Freeze uses the current directory.

lpLZH Points to the location and name of the archive file. If no archive file is found, Freeze will create a new one. If this parameter is NULL, Freeze exits with an error.

fuOptions Changes the way the function collects and stores files. This parameter can be a combination of the following

Value	**Meaning**
ICE_STOREFULLPATHS	the file will be stored with its absolute path. This is the default.
ICE_STORERELATIVEPATHS	the file will be stored with its absolute path from the starting directory.
ICE_STORENOPATHS	only the filename is stored.
ICE_RECURSIVE	if subdirectories are found in the starting directory, Freeze will look in these subdirectories for files matching lpMask.
ICE_MOVEFILES	aftering archiving the file(s) specified by lpMask, Freeze will attempt to delete them.
ICE_INCLUDEARCHIVEFILES	only files matching lpMask AND with the archive attribute set will be added to lpLZH.
ICE_INCLUDEREADONLYFILES	only files matching lpMask AND with the readonly attribute set will be added to lpLZH.
ICE_INCLUDESYSTEMFILES	only files matching lpMask AND with the system attribute set will be added to lpLZH.
ICE_INCLUDEHIDDENFILES	only files matching lpMask AND with the hidden attribute set will be added to lpLZH.
ICE_INCLUDENORMALFILES	only files matching lpMask AND with the hidden or system attribute *not* set will be added to lpLZH. This is the default.
ICE_TURNARCHIVEOFF	if a file added to lpLZH has the archive attribute set, then this attribute will be removed.
ICE_TURNREADONLYOFF	if a file added to lpLZH has the readonly attribute set, then this attribute will be removed.

ICE_TURNSYSTEMOFF   if a file added to lpLZH has the system attribute set, then this attribute will be removed.

ICE_TURNHIDDENOFF   if a file added to lpLZH has the hidden attribute set, then this attribute will be removed.

ICE_OVERWRITEALL   if a file in lpLZH matches exactly (path and filename) one specified in the current function call, the one about to be added will overwrite the original.

ICE_OVERWRITEIFNEWER   if a file in lpLZH matches exactly (path and filename) one specified in the current function call, the one about to be added will overwrite the original one ONLY if is is dated later than the original.

ICE_OVERWRITEQUERY   if a file in lpLZH matches exactly (path and filename) one specified in the current function call, the user will be prompted before the new one overwrites the original.

ICE_OVERWRITENEVER   if a file in lpLZH matches exactly (path and filename) one specified in the current function call, the one about to be added will be skipped

Return value

This function returns the number of files added to lpLZH if the function call was successful. An error code (see Error reference) is returned if a fatal error occurs.

Comments

By combining ICE_STOREFULLPATHS and ICE_RECURSIVE, Freeze is capable, in theory, of archiving entire drives. In practice this may not be possible due to a lack of resources when archiving a large number of files in one function call.

Examples

THAW

The Thaw function extracts (and optionally deletes) files from an archive.

Syntax

VB: Declare Function Thaw Lib "ICE.DLL" (lpMask as String, lpLZH as String, fuOptions as Long) as Integer

Parameter **Description**

lpMask This is the file(s) to extract from the archive. Standard DOS wildcard characters (* and ?) can be used to select multiple files. If this parameter is NULL, Freeze defaults to *.* i.e. all files. This parameter may also contain the starting directory, if no directory is given, Freeze uses the current directory.

lpLZH Points to the location and name of the archive file. If no archive file is found, or this parameter is NULL, Freeze exits with an error.

fuOptions Changes the way the function extracts files. This parameter can be a combination of the following

Value	Meaning
ICE_RESTOREDIRECTORIES	this option uses the path information stored in lpLZH. If the stored directories do not exist, they will be created. This option has no effect if no path information was stored when lpLZH was created.
ICE_DELETEFILES	the file(s) as specified by lpMask are deleted from the archive - no extraction occurs. Any path information in lpMask is ignored.
ICE_MOVEFILES	the file(s) as specified by lpMask are extracted, and deleted, from lpLZH.
ICE_OVERWRITEALL	if a file in lpLZH matches exactly (path and filename) one specified in the target directory, the extracted file will overwrite the target one.
ICE_OVERWRITEIFNEWER	if a file in lpLZH matches exactly (path and filename) one specified in the target directory, the extracted file will overwrite the target one, ONLY if the date of the file is later than the target's date.
ICE_OVERWRITEQUERY	if a file in lpLZH matches exactly (path and filename) one specified in the target directory, the user will be prompted before the extracted file one overwrites the target.
ICE_OVERWRITENEVER	if a file in lpLZH matches exactly (path and filename) one specified in the target directory, it is skipped.

Return value

On success, this function returns the number of files extracted. If a fatal error has occurred the function returns an error code (see Error messages).

Comments

None

Examples

ListArchiveContents

The ListArchiveContents function retrieves information concerning the files stored in an archive into an array of user-defined type ICEINFO. This function is defined in the language specific files (ICE.BAS or ICE.CPP) supplied with the ICE software not in the library itself.

Syntax

VB: Function ListArchiveContents (ByVal lpMask as String, ByVal LpLZH as String, lpInfo() as ICEINFO) as Integer

Parameter	**Description**
lpMask	This is the file(s) for which information is required. Standard DOS wildcard characters (* and ?) can be used to select multiple files. If this parameter is NULL, Freeze defaults to *.* i.e. all files. Any path information given is ignored.
lpLZH	Points to the location and name of the archive file. If no archive file is found, or this parameter is NULL, Freeze exits with an error.
lpInfo	An array of user-defined type ICEINFO.

Return value

On success, this function returns the number of files found matching the specified lpMask. On failure, this function returns an error code (refer to Error messages).

Comments

This function should be used to retrieve file information from an archive instead of the function ListArchive. Use of the ListArchive function is not documented.

The ICEINFO array is automatically dimensioned to be of the correct size.

Examples

Visual Basic

The following example compresses all the files in the root directory of drive C: into the archive TEST.LZH. The contents of the archive, as well as original and compressed sizes, are then displayed in a list box.

```
Dim iICE as Integer, I as Integer
Static aICE(10) as ICEINFO

iICE = Freeze("c:\*.*", "TEST.LZH", 0)
iICE = ListArchiveContents("*.*","TEST.LZH",aICE())
For I = 1 to iICE
List1.AddItem aICE(I).FileName & Chr$(9) & aICE(I).OriginalSize & aICE(I).CompressedSize
Next I
```

TestArchive

The TestArchive function tests the validity of an archive, returning the number of files contained in the archive if no error occurs.

Syntax

VB:	Declare Function TestArchive Lib "ICE.DLL" (ByVal LpLZH as String) as Integer

Parameter	**Description**
lpLZH	Points to the location and name of the archive file. If no archive file is found, or this parameter is NULL, Freeze exits with an error.

Return value

On success, this function returns the number of files found in the specified archive file. On failure, this function returns an error code (refer to Error messages).

Comments

None.

Examples

ADDCOMMENT

The AddComment function allows the user to add a comment to the start of an archive. This comment can then be displayed with the ShowComment function.

Syntax

VB:	Declare Function AddComment Lib "ICE.DLL" (ByVal LpLZH as String, ByVal lpComment) as Integer

Parameter	**Description**
lpLZH	Points to the location and name of the archive file. If no archive file is found, or this parameter is NULL, Freeze exits with an error.
lpComment	The comment to add to the archive.

Return value

On success, this function returns 1; on failure this function returns an error code (refer to Error messages).

Comments

This function uses the method outlined in the documentation of LHA to store the comment. The comment is stored in a file with the name ! at the beginning of the archive, and will be extracted as a file if the archive is expanded using other LHA decompressors.

Adding a comment to an archive file will automatically overwrite the previous comment (if one exists).

Examples

SHOWCOMMENT

The ShowComment function allows the user to display the comment stored in an archive. This comment should have been added using the AddComment function.

Syntax

VB: Declare Function ShowComment Lib "ICE.DLL" (ByVal LpLZH as String) as Integer

Parameter	Description
lpLZH	Points to the location and name of the archive file. If no archive file is found, or this parameter is NULL, Freeze exits with an error.

Return value

On success, this function displays the comment and returns 1, if no comment is found 0 is returned; on failure this function returns an error code (refer to Error messages).

Comments

Please refer to the AddComment function for a description of how ICE stores comments.

Examples

Error reference

Errors can be broken into 2 categories; fatal errors which result in the abnormal termination of the function (returning an error code), and warning messages which are displayed but the function continues.

Fatal Error Messages

The following is a complete list of all the fatal error codes returned by the functions defined in the ICE library. When a fatal error occurs the function will abort. If an error is encountered which does not appear in this list please contact the auther immediately.

-1 Cannot open archive

The specified archive file does not exist, is not in the specified location or is spelt incorrectly.

-2 Cannot open temporary file

The current drive is full, or you do not have write access to the current directory. Free up some space on the current drive, or change the current drive/directory before calling this function.

-3 Cannot close temporary file

The temporary file used in the creation of an archive has got an internal error - the file has probably been deleted, closed by another application, or another application is trying to write to this file. Do not delete temporary files from within Windows.

-4 Header sum error

The CRC value for the file being extracted, tested or deleted, did not match the value stored in the archive. The file is probably corrupted and cannot be recovered.

-5 File input/output error

An error has occured in trying to read/write to a file. The file has been deleted, or has been locked by another application.

-6 Out of memory

There is not enough memory to continue processing the archive - close all open applications and try again. Remember that the more files stored in an archive, the more memory needed to process it.

-7 Bad table

The file is corrupted or is not a LHA-compatible archive. Files that are damaged in this way are not recoverable.

-8 Destination drive is full

The drive containing the archive file is full. While the archive file has not been created, the temporary file has not been deleted and can be recovered.

-9 Could not write to archive file

When trying to copy the temporary file to the named archive, an I/O error has ocurred. You can usually recover the archive from the temporary file, which, because of this error, has not been deleted.

Warning Messages

The following is a list of all non-fatal errors (warnings) possibly encountered when using the ICE library. These warning messages appear as they occur, but function execution will continue when the user presses OK.

Cannot open ??????.??? - file skipped

During compression; the user does not have read access to the specified file or is locked by another application.

Cannot create ??????.??? - file skipped

During extraction; the specified file could not be created. The probable cause is that the user has no creation rights for a network drive.

Cannot expand ??????.??? - unsupported compression method

The file was compressed in a method not recognised by this version of ICE. The archive was either created by a later version of LHA, or the archive file is corrupted.

Cannot expand ??????.??? - cannot read extended header

The file header information contains extended information not compatible with this version of ICE. The archive was either created by a later version of LHA, or the archive file is corrupted.

Cannot expand ??????.??? - corrupted archive file

The specified file is corrupted in the archive. This may indicate a corrupt archive or one that has been tampered with.

Cannot delete ??????.??? - the file is probably read-only

The ICE_MOVEFILES flag was specified in the call to Freeze, but the function could not delete the file. The file is either read-only, is on a network directory with no permission to delete or the file is locked by another application.

TERMS AND CONDITIONS OF USE

License Agreement

This is the FREEWARE version of ICE, and is for private, non-business use only (subject to the distribution restrictions outlined below). Use in a business environment requires registration.

Distribution

The FREEWARE version of ICE may be distributed by any means subject to the following restrictions :-

- the software, its documentation and example files are distributed together
- the software and all its documentation is distributed free of charge (apart from handling costs)
- no modification is made to the software or its documentation
- the software is not distributed as part of another package (without the express permission of the author)

Copyright

ICE.DLL and its documentation remain the sole property of the author, and as such is copyright Stephen Darlington © 1995, all rights reserved.

Any other copyright(s), either expressed or implied, are duly acknowledged.

Disclaimer

The ICE compression software is supplied "as is", and as such the author cannot accept responsibility for any loss/damage of data resulting from its use.

You are automatically agreeing to the above terms by using the ICE library.

TECHNICAL SUPPORT/CONTACTING THE AUTHOR

Technical support for the FREEWARE version of ICE is offered *only* by email to TCRSTEPHEN@EPIDEM.ICR.AC.UK. Registered users also receive technical support by telephone or mail.

The development and enhancement of ICE.DLL is on-going. If you have any bug-reports, suggestions or comments please contact the author at the above email address.

REGISTRATION

Whilst registration is not necessary for the private user (see Terms and conditions of use), you are strongly recommended to do so. Registration provides a **single user** license, with the following benefits:

- automatic receipt of future versions of ICE when they become available
- receipt of the unregistered ZIP-compatible version (when available)
- a license to distribute ICE.DLL royalty-free with your own software
- a single user license to use ICE in a business environment
- written/verbal technical support

ICE can be registered by sending 15 pounds sterling/30 US dollars (preferably in cash, but money orders/crossed cheques drawn on a UK bank will be accepted) to the following address

Stephen Darlington
5 Clowser Close
Watergardens
Sutton
Surrey
United Kingdom
SM1 4TP

Please make all cheques/money orders payable to *S. Darlington*, and also you are advised to send your registration via registered post (especially for cash registrations).

RELEASE HISTORY

Beta release 3 - version 1

Added the TestArchive, AddComment and ShowComment functions. Made some internal changes to the library for compatibility reasons. Fixed other minor bugs.

Released 12th July 1995

Beta release 2 - version 2

Rewrote the ListArchive function to make it compatible with languages other Visual Basic. Wrote wrapper function ListArchiveContents. Completed help file (as far as possible) and provided a Visual Basic demonstration of ICE. Fixed problem with 0 length files and the ListArchiveContents function. Fixed problem encountered when creating archives on diskettes/network drives. Fixed minor bugs.

Release 3rd May 1995 to VISBAS-L subscribers.
Updated 5th May 1995 - not widely released.
Updated 2nd June 1995 - not widely released.

Beta release 2 - version 1

Expanded Freeze and Thaw functions to include DOS wildcards, and added various options to recurse directories, store full or relative path and to overwrite files. Included ListArchive function to retrieve file information into an array (VB only). Included means of providing feedback of Freeze and Thaw operation.

Released 19th April 1995 to VISBAS-L subscribers.

Beta release 1 - version 1

Bare-bones release to gain feedback from users. Freeze and Thaw only compress one file at a time, and have no options to alter operation.

Release 24th February 1995 to VISBAS-L subscribers.

LIMITATIONS

The following are the known limitations of the ICE:LHA library:

Freeze - the number of files compressed during a single call to ICE:LHA is limited by the amount of free memory. This should not be an issue for the majority of users.

Thaw - ICE:LHA is currently not compatible with LHARC-created archives. Whilst you can use ICE:LHA to view the contents of such archives, you will not be able to extract the files. A future version of ICE:LHA will correct this problem.

ListArchiveContents - Visual Basic users may experience problems when viewing the contents of archives that contain many files. This is due to the limitation of string length within Visual Basic. A future version of ICE:LHA will correct this problem.

Cyclic Redundancy Check