General page information

aspEasyZIP

Last modified : 12/11/2004 (Version 3.3
)

What's aspEasyZip

This is an ASP component for use with the IIS Server from Microsoft , the main feature is that can zip, unzip and show the contents of a zip file in ASP language, even you can do that in VB and VB Scripting. Read the documentations and experience with some examples to learn more about it.

This component is **freeware** but it has a small limitation, when zipping and unzipping it will display a message on the html page and it will put a reminder file inside the zipped file, if you want to remove the message and the reminder file then you will have to pay 20 US$. This is a small limitation because everyone can use it with no limitation of time and functionality.

The professional aspEasyZIP is a new product that started from version 2.0 and includes several features that does not include the standard version; the encryption and the auto extract function for Win9x/ME/NT/2000/XP. This version is called professional and you have to pay 69 US$ to register it, those option are not on the freeware even on the standard registered.

For additional information contact us at: support@mitdata.com or visit our web site at: www.mitdata.com

Prices & Conditions

All products when registered have the following features:

- Free email support for all of our products
- Minor, Major upgrades and bug fixes
- Notification by email when newer versions are released

Prices and conditions

Visit our page to order the full version

To order one product go to http://www.mitdata.net/online
We process orders in 5 minutes with Visa Credit card

Product	**Code to Order it**	**Price per/ Server**	**Site License**	Full Mail Support	Free Updates for minor releases	Free Updates for major releases	Special Prices for major releases	Bug fixes e-mail notifications
aspEasy ZIP	32839	$ 20 USD	$ 100 USD	**Yes**	**Yes**	**No**	**Yes**	**Yes**
aspEasy ZIP Pro	44318	$ 6 9USD	$ 14 9USD	**Yes**	**Yes**	**Yes**	**N/A**	**Yes**

N/A = Not applicable

We have also special upgrades prices between versions, contact us to get more information.

Email us for special conditions and quantities prices.
Schools and Universities (Educational Services) have 35% off from regular price

The site license enables you to install it on all servers / machines from your company with out any restrictions or develop your own selling software with our component included

History

aspEasyZIP

Last modified : 12/11/2004

History

3.30 (12/11/2004)
'' Added rename function
3.22 (04/04/2004)
'' Bug removed when decoding large MIME strings
'' Bug removed Encoding and Decoding Binary strings
3.21 (04/02/2004)
'' Sometimes the ZipString and UnZipString was hanging, added exception controls on the code.
3.2 (02/02/2004)
'' compress and encode strings / decode decompress string
3.1 (20/11/2003)
'' Added new functions that controls the corruptions and ZIP repairs ZipCheck and ZipFix
'' Calling destroy several times does not hang the component, even if this is called on the ASP page.
3.01 (20/07/2003)
'' Removed the warning error code 10502
3.00 (17/02/2003)
'' New version library
'' Zip Items
'' Extract base dir
'' RootDir property was locking the directory
'' Debug is compatibles with VB messages, no more html codes
'' Solves the nasty bug D:\ Write permissions. Thanks to Nicolas Sporn to be so patient to test it until we found it.
'' More debug options. Debug the DLL libraries
'' Dynamic DLL loading when needed
'' List contents of a ZIP doesn't need the libraries any more.
'' You can set a different directory for the ZIP.DLL
'' New installation program
'' New help file

2.31 (12/05/2001)
'' Small bug found that when unzipping was locking the whole directory making it undeletable after finishing the unzip method.
'' New installation program to make it more easy to install and using it, even registering is more easy with aspEasyREG program.

2.3 (27/04/2001)
'' Added the GetZipItem_CDateTime that returns the date time in windows format and not the Zip format returned by GetZipItem_DateTime easyZip.txt.
'' Added the EncodeUU and DecodeUU, using the standard that uses WinZIP. (Exchange zip files into different platforms)
'' Removed a bug from the registered version that was causing by RemoveZIP when selecting files that were not in.
'' Changed the way that is stored the easyZip.txt on the freeware version, this removes the full directory that is stored when using the root property.

2.21 (11/03/2001)
'' Removed a small bug that was ignoring the extract options.

2.2 (09/01/2001)
'' Added a new property "RootDir" for working with relative paths on the zip archive.

2.1 (18/12/2000)
'' Fixed a bug when Unzipping and not specifying the ExtrBaseDir

" Added a new property to specify the temp directory to be used for zipping and unzipping.
" Added debug information on some methods and properties that I forget to put on the 2.0 version.
" For the unregistered version now adds a reminder file when zipping and unzipping, these has been done because some people were able to hide the reminder messages. Sorry for the inconvenience.

2.0 (01/08/2000)
" Major new functionalities on the zip, like see the contents of a zip file and delete items on the zip.
" Also a new professional version where you can handle encryptions, with password protected. And create self auto extract option with zip files, converting a zip file to an exe one. (Only Win32)

1.01 (19/06/2000)
" Revision 01, added a new property LastMessage

1.0 (01/02/2000)
" First version

AddOptions

AddOptions is a Set of options to modify the default action of the ZIP method. The value is an integer and you can combine multiple options.

AddDirNames (value = 1)
Saves the pathname with each filename.
NOTE: the root directory name is never stored in a pathname; in other words, the first character of a pathname stored in the zip file's directory will never be a slash.

AddZipTime (value = 2)
Set Zip timestamp to that of the newest file in the archive.

AddRecurseDirs (value = 4)
Subdirectories below EACH given file specification will be included in the Zip archive.
Defaults to False.
WARNING: This is potentially dangerous if the user does this from the root directory. The hard drive may fill up with a huge Zip file!

AddHiddenFiles (value = 8)
Files with their Hidden or System attributes set will be included in the Add operation.

AddEncrypt (value = 16)
Add the files with standard zip encryption. You must set the password property. (only professional version)

AddSeparateDirs (value = 32)
Add separate entries to the zip archive that will hold the name of each directory from root. (No data for these entries).NOTE: - To use this, you must also have AddDirNames set.

AddMove (value = 256)
After adding to archive, delete original file. Potentially dangerous. Use with caution!
NOTE: Freshen and Update can only work on pre-existing archives. Update can add new files to the archive, but can t create a new archive.

AddFreshen (value = 1024)
Add newer files to archive (only for files that are already in the archive).

AddUpdate (value = 2048)
Add newer files to archive (but, any file in an file specification that isn't already in the archive will also be added).

Syntax
ZIP .AddOption = OptionValues

Parameter
(Read / Write) Integer

Example
Will compress al files from the actual directory getting recursive to all subdirectories that it will find and storing full path inside the zip file:

```
<%
 const AddDirNames = 1
 const AddRecurseDirs = 4
 set ZIP = server.createobject("aspZip.EasyZIP")
Zip.ZipFileName = "backup.zip"
Zip.AddOptions = AddDirNames + AddRecurseDirs
Zip.ArgsAdd( '*.*' )
```

```
 Zip.Zip
  set ZIP = nothing
%>
```

Debug

Sets the component in debug mode, when you made any operation then it displays the debug on the screen. Default is set to False, so there is no debug information when running up.
When running the component from another language different from the ASP, it will display a windows popup a message evey call on any intruction of aspEasyZIP.

Syntax
ZIP .Debug = True / False

Parameter
(Read / Write) Boolean

Example

```
<%
  set ZIP =
server.createobject("aspZip.EasyZIP")
 ZIP.Debug = True
 ' Do all stuff here
%>
```

Debug DLL

Sets the component in debug mode for the DLLs. The aspEasyZIP communicates to the unzdll.dll and zipdll.dll, every call is intercepted and printed out on the screen, or displayed when running out from the ASP.

Syntax

ZIP .DebugDLL = True / False

Parameter

(Read / Write) Boolean

Example

```
<%
  set ZIP = server.createobject("aspZip.EasyZIP")
 ZIP.Debug = True
 ZIP.DebugDLL = True
 ' Do all stuff here  now we will get more information on how is going on
%>
```

DLL Directory

It could happen that you cannot use the system32 directory from windows to put the additional libraries, (zipdll.dll and unzdll.dll). With this property you can set a special directory out of the system directory, or actual path, and the component will search and run it from there.
This is very important if you place the component on an ISP that doesn't want to use external DLL on the Windows / System directory.

Syntax

```
ZIP .DLLDirectory = "Set Directory"
```

Parameter

(Read / Write) String

Error

Number of the error when you call the ZIP or UNZIP method, it returns 0 when there are no errors. Use the LastMessage property to get an explanation of the error.

List of comom errors:

-1	File not found
2	Unexpected end of file
3	Invalid structure of Zip file
4	Out of memory
5	Internal error
6	Temporary file error (check write permissions or use a different temporary directory)
10	Temporary file error (check write permissions or use a different temporary directory)
11	Error reading file
13	Missing Zip File or unable to write zip (check permissions on the INET_xxxx user)
14	Error writing to the a file
15	Error creating a file
16	Bad zip option specified (check the addOption)
17	File not found or not read permissions

Syntax
Error = ZIP .Error

Parameter
(Read) Integer

Example

```
<%
  set ZIP = server.createobject(" aspZip.EasyZIP ")
 Zip.ZipFileName = " backup.zip "
 Zip.ArgsAdd( " *.asp " )
 Zip.Zip
 if Zip.Error <> 0 then
    response.write " Error : " & Zip.LastMessage &
" <br> "
 else
    response.write Zip.SuccessCnt & " files
zipped<br> "
 end if
  set ZIP = nothing
%>
```

ExtrBaseDir

This base directory applies only to UnZIP operations. The component will make this directory the current directory before extracting any files. If you don't specify a value for this property, then the directory of where it reside the Zip file will be used as the base directory for extractions.

Syntax

ZIP .ExtrBaseDir = DirectoryName

Parameter

(Read / Write) String

ExtrOptions

ExtrOptions is a Set of Options used to modify the default actions of the UnZip method. You can combine multiple values.
ExtrDirNames (value = 1)
Extracts and recreates the relative pathname that may have been stored with each file. Empty directories stored in the archive (if any) will also be recreated.

ExtrOverWrite (value = 2)
Overwrite any pre-existing files during Extraction.

ExtrFreshen (value = 4)
Extract newer files from archive (only for files that already exist). Won't extract any file that isn't already present.

ExtrUpdate (value = 8)
Extract files that don t already exist and if ExtrOverWrite is also set to true, it will extract newer files from the archive.

ExtrTest (value = 16)
Only test the archive to see if the files could be successfully extracted. Testing is done by extracting the files, but NOT writing the extracted data to the disk.

Syntax
ZIP .ExtrOptions = Options

Parameter
(Read / Write) Integer

LastMessage

Returns the explanation of the last action, if you get an error on the last action performed then you will get a brief explanation with that function of what happened.

Syntax

DisplayMsg = Zip .LastMessage

Parameter

(Read) String

Example

```
<%
  set ZIP = server.createobject(" aspZip.EasyZIP ")
 Zip.ZipFileName = " backup.zip "
 Zip.ArgsAdd( " *.asp " )
 Zip.Zip
 if Zip.Error <> 0 then
    response.write " Error : " & Zip.LastMessage &
" <br> "
 else
    response.write Zip.SuccessCnt & " files
zipped<br> "
 end if
  set ZIP = nothing
%>
```

MaxFiles

**** Only for Professional version property ****

Specifies the maximum files that can be inside the zipped file.

Syntax

Zip .MaxFiles = number

Parameter

(Read / Write) Integer

Example

```
<%
  set ZIP = server.createobject(" aspZip.EasyZIP
")
 Zip.ZipFileName = " backup.zip "
  Zip.MaxFiles = 10
 Zip.ArgsAdd( " *.asp " )
 Zip.Zip
  set ZIP = nothing
%>
```

MaxFileSize

**** Only for Professional version property ****

This property specifies the maximum size of an individual file.

Syntax

Zip .MaxFileSize = number (bytes)

Parameter

(Read / Write) Integer

Example

```
<%
  set ZIP = server.createobject(" aspZip.EasyZIP
")
 Zip.ZipFileName = " backup.zip "
 Zip.MaxFileSize = 10000
 Zip.ArgsAdd( " *.asp " )
 Zip.Zip
  set ZIP = nothing
%>
```

MaxZippedSize

**** Only for Professional version property ****

Limits the maximum size of the zipped file that can not be reach.

Syntax

Zip .MaxZippedSize = number (bytes)

Parameter

(Read / Write) Integer

Example

```
<%
  set ZIP = server.createobject(" aspZip.EasyZIP
")
 Zip.ZipFileName = " backup.zip "
 ' Limit the zip file to 100k
 Zip.MaxZippedSize = 100000
 Zip.ArgsAdd( " *.asp " )
 Zip.Zip
  set ZIP = nothing
%>
```

MaxUnzippedSize

**** Only for Professional version property ****

Limits the maximum size of the zipped file by the sum of all unzipped files.

Syntax

Zip .MaxUnzippedSize = number

Parameter

(Read / Write) Integer

Example

```
<%
  set ZIP = server.createobject(" aspZip.EasyZIP
")
 Zip.ZipFileName = " backup.zip "
  ' Limit the zip file to 500k
 Zip.MaxUnzippedSize = 500000
 Zip.ArgsAdd( " *.asp " )
 Zip.Zip
  set ZIP = nothing
%>
```

NVersion

Returns the version number of the component, you can check it to provide compatibility between different versions of the component.

Returning values

100 - this is version 1.0
110 - for version 1.10
200 - for version 2.00
300 - for version 3.00

Example

```
<%
if ZIP.NVersion < 300 then
   response.write " This version is not supported by our source code get a newer one "
end if
%>
```

Password

**** Only for Professional version property ****

Password is the key used when encrypting or decrypting a Zip archive. With version 3.0 you can protect individual files with the "<" character on the ArgsAdd specification.

Syntax

```
Zip .Password = "password"
```

Parameter

(Write) String

RootDir

RootDir sets the base directory for relative path file specifications.
The RootDir property determines where zipping will start for any filenames or wildcard entries in the ArgsAdd method.

Syntax

Zip .RootDir = Directory

Parameter

(Read / Write) String

Example

This will add the file D:\Program Files\Start\Dir1\MyFile.htm as Dir1\MyFile.html and all files D:\Program Files\Start*.h as *.h to the zip archive.

```
<%
Zip. RootDir  = " D:\Program
Files\Start "
Zip. ArgsAdd( " Dir1\MyFile.html ")
Zip. ArgsAdd("  *.h ")
Zip. Add
%>
```

SetProxy

**** Only for Professional version property ****

This property affects only to the AddURL method, when getting data trough http protocol you can set the proxy configuration if you are behind a proxy sever. You must specify the server address and the port.

Syntax

Zip .SetProxy = ProxyServer

Parameter

(Read / Write) String

Example

Will create google.zip with the logo.gif retrieved from internet. Remember to set the correct proxy configuration:

```
<%
  set ZIP = server.createobject(" aspZip.EasyZIP ")
 Zip.ZipFileName = " google.zip "
 Zip. SetProxy = " 10.0.0.0:80 "
 Zip. AddURL(" http://www.google.com/images/ logo.gif ", "
logo.gif ")
  set zip = nothing
%>
```

SFXCaption

**** Only for Professional version property ****

Is the caption of the SFX dialog box when you start executing a .EXE archive.

Syntax

```
Zip .SFXCaption = "Title"
```

Parameter

(Write) String

SFXCommandLine

**** Only for Professional version property ****

Can contain a command line to execute after extracting the executable.

Special symbols:
| (Vertical bar) is the command / argument separator.
>< Is the actual extraction directory selected by user.
Example: notepad.exe|><Readme.txt
This will run notepad to show "Readme.txt" in the actual extraction dir.

Syntax

Zip .SFXCommandLine = "command line to run"

Parameter

(Write) String

SFXDefaultDir

**** Only for Professional version property ****

The directory that will be used by the executable when extracting.

If you specify the special symbol >< here, then the user s temporary directory will be the default extraction directory.

Syntax

```
Zip .DefaultDir = "Directory"
```

Parameter

(Write) String

SFXOptions

**** Only for Professional version property ****

Set of options used to modify the default actions of the ConvertSFX method.

SFXAskCmdLine(value = 1)
Allows user (at runtime) to de-select the SFX program s command line checkbox. Once de-selected, the command line will not be executed.
NOTE: The checkbox doesn't appear unless there is a command line specified.

SFXAskFiles (value = 2)
Lets user (at runtime) modify the SFX program's list of files to be extracted.

SFXAutoRun (value = 4)
Extraction of the SFX contents will be performed automatically, no user actions are required.
NOTE : This works only if the SFX's filename starts with an exclamation mark (!) -for security reasons-For example: !AUTORUN.EXE
WARNING: Use this only in rare cases! We advise you NOT to use this because files will be extracted onto the user's disk without his knowledge.

SFXCheckSize (value = 8)
Then the size of the SFX executable will be checked before extracting. By default checking is set to True. If size checking is off (False) and the SFX file layout is wrong it is very likely you will get a system error.

SFXHideOverWriteBox (value = 16)
Does NOT show the user (at runtime) the SFX program s dialog box that lets him choose the overwrite action for files that already exist.

Syntax
Zip .SFXOptions = Values

Parameter
(Read / Write) Integer

SFXHideOverwriteBox

**** Only for Professional version property ****

Sets the the action for what the SFX program is supposed to do if it finds files that already exist.
OvrConfirm (value = 1)
Ask user when each file is found (The default).

OvrAlways (value = 2)
Aways overwrite existing files.

OvrNever (value = 4)
Never overwrite - skip those files.

Syntax
Zip .SFXHideOverWriteBox = Values

Parameter
(Read / Write) Integer

SFXBinPath

**** Only for Professional version property ****

Sets the binary file to use when creating exe file, actually the are two version, English and German languages.

Syntax
Zip .SFXBinPath = " Directory "

Parameter
(Write) String

SuccessCnt

Number of files that were successfully operated when you call ZIP or UNZIP method.

Syntax
number = ZIP .SuccessCnt

Parameter
(Read) Integer

Example

```
<%
  set ZIP = server.createobject(" aspZip.EasyZIP ")
 Zip.ZipFileName = " backup.zip "
 Zip.ArgsAdd( " *.asp " )
 Zip.Zip
 if Zip.Error <> 0 then
    response.write " Error : " & Zip.LastMessage &
" <br> "
 else
    response.write Zip.SuccessCnt & " files
zipped<br> "
 end if
  set ZIP = nothing
%>
```

Temp

Temp is the path and directory that will be used instead of the default temporary directory. The default temporary directory when zipping and unzipping is the same directory as where the zipfile is located.

Remember to set write permissions to the directory

Syntax

```
ZIP .Temp = " C:\Temp "
```

Parameter

(Read / Write) String

Version

Returns de version of the component and the version of the zip / unzip libraries. This is very useful because the libraries of zipdll and unzdll must be correct, if not then you will get unexpected results.

Important note:
For version 3.0 of aspEasyZIP you need version 1.60 or higher for the zip libraries
If you use 2.3 or lower version of the component then you must use version 1.52 or lower for the zip/unzip libraries

Syntax
Vers = ZIP .Version

Parameter
(Read) String

Example

```
<%
  set ZIP = server.createobject("aspZip.EasyZIP")
  if not isObject(ZIP) then
    response.write "EasyZIP is not installed.<br>"
  else
    response.write "EasyZIP version installed: " & ZIP.Version &
"<br>"
  end if
  set ZIP = nothing
%>
```

ZipFilename

You specified the name of the file to Zip or UnZIp file to work with. If you want to zip files to a new zip file, then it will create automatically the zip. If it exists then it will add the files to the zip.
When unzipping the file must exist.

Note:
You must set the full path directive, you can not use the Internet connotation, to do that use the **Server.MapPath** method.

Syntax
ZIP. ZipFileName = "zipfilename"

Parameter
(Read / Write) String

Example

```
<%
  set ZIP = server.createobject("aspZip.EasyZIP")
 Zip.ZipFileName = Server. MapPath
("../backup.zip")
 '...
%>
```

AddURL From version 3.0

**** Only for Professional version ****

This is on of the best features of the new version 3.0, it's able to retrieve an URL from internet and zip it. This will help to administer remote sites or make remote backups.
You must specify the full URL path, like http://www.internetplace.whatever, and the filename that you whish to appear on the zip filename. You don't have to use the ZIP method to zip it because it does it automatically.
You can set a password for this individual file item when specifying the filename, use the lower then character to separate the filename from the password property: "filename<pasword".

NOTES
If you are behind a proxy server you can use the SetProxy property to tell aspEasyZIP to use it to retrieve the URL.
It doesn't accept any patterns to include multiple files.

Syntax
ZIP .AddURL (URL, Filename)

Example

```
<%
  set ZIP = server.createobject("aspZip.EasyZIP")
 Zip.ZipFileName = "ASP_Backup.zip"
 Zip.AddURL( "http://www.google.com/images/logo.gif", "google.gif" )
  response.write Zip.Get_LastMessage
  set ZIP = nothing
%>
```

ArgsAdd

With this method you add the files you want to ZIP or Unzip. Note: You can use patterns to made multiple selections or use this function to add many files to the zip or the unzip.

If you want to set a password to individual files you can use the "<" character to specify a password for a selection or a single file. See example.

Now from version 3.0 if you add files with the following extensions, 'gif', 'png', 'z', 'zip', 'zoo', 'arc', 'lzh', 'arj', 'taz', 'tgz', 'lha', 'rar', 'ace', 'cab', 'gz', 'gzip', 'jar', 'exe', it will not try to zip it just store them on the file, this will improve speed when zipping.

NOTES
Remember to add the correct permissions to the user **IUSER_** ***MachineName***, if not you will get an error, usually number 10

Syntax
ZIP .ArgsAdd

Example

```
<%
  const AddDirNames = 1
  const AddRecurseDirs = 4
  set ZIP = server.createobject("aspZip.EasyZIP")
 Zip.ZipFileName = "ASP_Backup.zip"
 Zip.AddOptions = AddDirNames + AddRecurseDirs
 Zip.ArgsAdd( "*.asp" )
 Zip.ArgsAdd( "*.aspx" )
 Zip.ArgsAdd( "*.inc" )
 ' Professional version can use a password for individual files

 Zip.ArgsAdd( "*.key<mypassword' )
 Zip.Zip
  set ZIP = nothing
%>
```

ArgsExc From version 3.0

You can Exclude files from the ArgsAdd function. Use the DOS patterns to exclude files.

Syntax

ZIP .ArgsExc string

Example

```
<%
  const AddDirNames = 1
  const AddRecurseDirs = 4
  set ZIP = server.createobject("aspZip.EasyZIP")
 Zip.ZipFileName = "ASP_Backup.zip"
 Zip.AddOptions = AddDirNames + AddRecurseDirs
 Zip.ArgsAdd( "*.*" )
 ' Exclude bak files
 Zip.ArgsExc( "*.bak" )
 Zip.Zip
  set ZIP = nothing
%>
```

ArgsClear

Clear the list of files to zip or UNZip. When you set the ZipFileName then it performs an ArgsClear automatically.

Syntax
ZIP .ArgsClear

CheckWriteProtection From version 3.0

This is an easy way to check if the directory is write protected, is quite common on ASP and ASP.NET pages to not have write permission to write on the IIS directory. This is because of the security directive on the IIS default configuration.
If you want to check if the directory is write protected before doing a saving action, then you can use it to retrieve if it's able to write it or not.

Syntax
Boolean = *ZIP* .CheckWriteProtection Directory

ConvertSFX From version 2.0

**** Only for Professional version ****

ConvertSFX converts the Zip archive to a self-extracting executable. This functions needs a BIN file that is attached to the ZIP file and it manages all the decompressions routines, to change this module, specify the full directory or use a different language SFX runtime, use the SFXBinPath property.

Convert Zip archive to self-extracting .EXE. The resulting .EXE is a Win32 program. Uses the same base name and path already used by the ZipFileName - only the file extension is changed to "EXE".
This function returns:

0	On success
-1	Error in creation of destination.
-2	Read or write error during copy. (Could be: Not enough space on disk)
-3	Error in open of source
-9	General failure during copy

Example

```
<%
  set ZIP =
server.createobject("aspZip.EasyZIP")
 Zip.ZipFileName = "ASP_Backup.zip"
 Zip.SFXBinPath = "DZSFXUS.BIN"
 Zip.ConvertSFX
  set ZIP = nothing
%>
```

ConvertZIP From version 3.0

**** Only for Professional version ****

This is the reverse engine of the ConvertSFX function, if you want to transform the SFX EXE file to ZIP file again.

Syntax

ZIP .ConvertZIP

Create From version 3.0

The create method is designed for programming languages that does not use the OnStartPage method from the IIS, like VB, C++ or Delphi. This method will initialize the memory for use the EasyZIP
Warning, do no try to call this method in ASP and even call twice or more times in other languages, if you do so you will get an exception.

Syntax
ZIP .Create

Example in VBS

```
' This demostrates the easy to add it on the VBS
 set ZIP = createobject("aspZip.EasyZIP")
ZIP.Create
 WScript.Echo ZIP.Version
Zip.ZipFileName = "ASP_Backup.zip"
Zip.ArgsAdd( "*.asp" )
Zip.Zip

ZIP.Destroy
 Set ZIP = Nothing
```

DecodeUU From version 2.3

Decode from UU format.
Normally the file has an extension of . **uue** and the original file name is stored inside the coded file and will use that filename to restore it. It can be used to decode any UU file that has been previously coded with this format. This format is widely used for cross platform exporting.

Syntax
ZIP .DecodeUU Filename

Destroy From version 3.0

When you create the object you must destroy it when finishing working with, this will release all the memory that has take for the process when generating the bar code.
This is for programming languages that does not use the OnEndPage method inside, like VB, C++ or Delphi.
Warning, do no try to call this method in ASP and even call twice or more times in other languages, if you do so you will get an exception.

Syntax
ZIP .Destroy

DLLVersion From version 3.0

Returns more version information of the DLL.

Syntax
Integer = *ZIP* .DLLVersion (Kind)

Parameters

Kind parameter	
Integer Param.	**Description**
0	Returns: **0** - If the aspEasyZIP version is Shareware. **1** - For the standard version **2** - For the professional version
1	Returns the ZIPDLL.DLL version
2	Returns the UNZDLL.DLL version

EncodeUU

From version 2.3

Encode the filename, normally a zip file, in UU format. It uses the WinZip standard. And permits cross platforms exporting.

Syntax

ZIP .EncodeUU Filename

GetZipItem From version 3.0

Gets a full structure of a zip item

GetZipItem sCount

Return the number of files in the Zip archive.

GetZipItem sCountDir

Return the number of directories in the Zip archive.

GetZipItem _FileName

Gets the filename of one item on the ZIP contents.

Syntax
String = *ZIP* .GetZipItem_Filename (Index)

Example

```
<%
  set ZIP = server.createobject("aspZip.EasyZIP")
 Zip.ZipFileName = "ASP_Backup.zip"
 N = Zip.GetZipItemsCount
  Response.write "Zip contents:<br>"
  For f = 0 to N - 1
   Response.write Zip.GetZipItem_FileName(F) &
"<br>"
 Next
  set ZIP = nothing
%>
```

GetZipItem _FileComment

Gets the file comment of one item on the ZIP contents.

Syntax
String = *ZIP* .GetZipItem_FileComent (Index)

GetZipItem _DateTime

Gets the file date time stored of one item on the ZIP contents.

Syntax
Integer = *ZIP* .GetZipItem_DateTime (Index)

GetZipItem _CDateTime

Gets the file date time stored of one item on the ZIP contents. This differs from the GetZipItem_DateTime that the time is converted from the File DateTime to the Visual Basic date time field.

Syntax

Integer = *ZIP* .GetZipItem_CDateTime (Index)

GetZipItem _CRC32

Gets the checksum file stored of one item on the ZIP contents.

Syntax
String = *ZIP* .GetZipItem_CRC32 (Index)

GetZipItem _CompressedSize

Gets the compressed file size stored of one item on the ZIP contents.

Syntax
Integer = *ZIP* .GetZipItem_CompressedSize (Index)

GetZipItem _UncompressedSize

Gets the uncompressed file size stored of one item on the ZIP contents.

Syntax
Integer = *ZIP* .GetZipItem_UncompressedSize (Index)

RemoveZip

With the RemoveZip method you can delete specified files from the Zip archive.

NOTES
UNREGISTERED USERS: On the unregistered version you can not delete items.

Syntax
ZIP .RemoveZIP

Rename

With the Rename function you can rename a file to a different one.

WARNING: There is virtual no check on the destination specified.

This means that if you are not careful you can get multiple names or even wrong names that can't be extracted anymore.

Syntax

Integer = *ZIP* .Rename (Original name, New name)

Results:
0 All Ok.
-7 All known Rename errors.
-8 Memory allocation error.
-9 General unknown Rename error. (DS_ErrorUnknown)
-10 Destination should also be a filename when the source is a filename.

Size ZIP String

When Zipping a string without encoding it you may need to known the exact size of the screen, use this property to known the correct size.

UNZIP

Unzips the files passed by ArgsAdd , if you specify *.* then it will unzip all files. Also you can specify just a file.

NOTES
On the unregistered version it extracts a reminder file to the ExtrBaseDir directory, when you register the component this is removed. The filename is easyzip.txt.

Syntax
ZIP .UNZip

Example

```
<%
 set ZIP = server.createobject("aspZip.EasyZIP")
 Zip.Debug = true
 Zip.ZipFileName = Server.MapPath("test.zip")
 ' Extract all files
 Zip.ArgsAdd("*.*")
 ' Which directory to extract the files
 Zip.ExtrbaseDir = "c:\inetpub\wwwroot\test_dir\"
 Zip.ExtrOptions = 1
 Zip.UnZip
 response.Write "Files UnZipped=" & Zip.SuccessCNT & " with Error="& Zip.Error & " message=" & Zip.LastMessage
  set ZIP = nothing
%>
```

UN ZIP String

Decodes and Unzips a string and returns the original text. If you don't use the encode/decode string you should check the SizeZipString property to known exactly the size of the string.

When using it on ASP you should always set the encode / decode to true, encoding will make you zipped string larger, you should get a good relation when using large string.

Note: Zipping strings does not require additional DLL

Syntax

ZIP .UNZIPString (text as string , decode as boolean)

Example

Decode and Unzip " this is a test, test, test, test, test "

```
<%
  str =
"eNorycgsVgCiRIWS1OISHRwkAAIGDUY="
  set ZIP = server.createobject("aspZip.EasyZIP")
 decodedZIPstr = ZIP.UnZIPString( str, True )
  set ZIP = nothing
%>
```

Unzip with out decoding it " this is a test, test, test, test, test "

```
  str = "xÚ+ÉÈ,V"
  set ZIP =
server.createobject("aspZip.EasyZIP")
 UnZIPstr = ZIP.UnZIPString( str, False )
  set ZIP = nothing
```

UNZIPBinaryWrite From version 3.0

**** Only for Professional version ****

Syntax

ZIP .UnZIPBinaryWrite string

Example

```
<%
  set ZIP =
server.createobject("aspZip.EasyZIP")
 Zip.ZipFileName = "text.zip"
 ' We can extract only one file
 Zip.ArgsAdd("Expedicion.log")
 ' Send the file to the client with this content type
 Zip.UnZIPBinaryWrite( "application/txt" )
  set ZIP = nothing
%>
```

ZIP

Starts to Zip the files passed by the ArgsAdd , when success then the property SuccessCNT contains the number of ZIPped files.

NOTES
On the unregistered version it adds a reminder file to the zip file, when you register the component this is removed. The filename is easyzip.txt and is zipped.

Syntax
ZIP .ZIP

Example

```
<%
  set ZIP =
server.createobject("aspZip.EasyZIP")
 Zip.ZipFileName = "Backup.zip"
 Zip.ArgsAdd( "*.*" )
 Zip.Zip
  set ZIP = nothing
%>
```

ZIP Check

Checks if the Zipped file is damaged, on the result you will get the exact error if the zip file is damaged. If it's 0 then the file is OK.

Syntax
Error = *ZIP* .ZIPCheck

ZIP Fix

Tries to fix a damaged zip file by creating a new one, if there are files that can not be recovered then those will be deleted from the new zip file.

Syntax

ZIP .ZIPFix (newZipFile as string)

ZIP String

Zips a string and returns an encoded binary string or just the binary string.

When using it on ASP you should always set the encode / decode to true, encoding will make you zipped string larger, you should get a good relation when using large string.

Note: Zipping strings does not require additional DLL

Syntax

ZIP .ZIPString (text as string , encode as boolean)

Example

```
<%
  str = "this is a test, test, test, test, test"
  set ZIP =
server.createobject("aspZip.EasyZIP")
 encodedZIPstr = ZIP.ZIPString( str, True )
  set ZIP = nothing
%>
```

Include file

```
<%
' INCLUDE FILE aspEasyZIP ( www.mitdata.com )
'// Addoptions
const csAddDirNames = 1
const csAddZipTime = 2
const csAddRecurseDirs = 4
const csAddHiddenFiles = 8
const csAddEncrypt = 16
const csAddSeparateDirs = 32
const csAddDiskSpan = 64
const csAddDiskSpanErase= 128
const csAddMove = 256
const csAddFreshen = 1024
const csAddUpdate = 2048
'// ExtrOptions
const csExtrDirNames = 1
const csExtrOverWrite = 2
const csExtrFreshen = 4
const csExtrUpdate = 8
const csExtrTest = 16
'// AutoExtract Options SFX
const csSFXAskCmdLine = 1
const csSFXAskFiles = 2
const csSFXAutoRun = 4
const csSFXCheckSize = 8
const csSFXHideOverWriteBox = 16
const csOvrConfirm = 1
const csOvrAlways = 2
const csOvrNever = 4
%>
```

Sample

```
<%

    dim ZIP
    ' Create the object
    set ZIP = server.createobject(" aspZip.EasyZIP" )
    ' Debug, show what's going on
    ZIP.Debug = True
    ' This does the work
    ZIP.ZipFileName = Server.MapPath (" test.zip ")
    ZIP.ArgsClear
    ZIP.ArgsAdd( Server.MapPath (" Expedicion.Log ") )
    ZIP.Zip
    if ZIP.Error <> 0 then
       response.Write " Error: " & ZIP.LastMessage
    else
       response.Write ZIP.SuccessCnt & " files added to zip file "
    end if
    Set ZIP = nothing



%>
```