newsight
Zip Component
version 1.0

Documentation

CONTENTS

INTRODUCTION

Description

A collection of functions to use the Java Zip file API.

LiveDocs

http://livedocs.newsight.de/com/Zip/

Notice

For comments, bug reports or suggestions to optimise this component, feel free to send a E-Mail: info@newsight.de

License

THIS IS A OPEN SOURCE COMPONENT. YOU ARE FREE TO USE THIS COMPONENT IN ANY APPLICATION, TO COPY IT OR MODIFY THE FUNCTIONS FOR YOUR OWN NEEDS, AS LONG THIS HEADER INFORMATION REMAINS IN TACT AND YOU DON'T CHARGE ANY MONEY FOR IT. USE THIS COMPONENT AT YOUR OWN RISK. NO WARRANTY IS EXPRESSED OR IMPLIED, AND NO LIABILITY ASSUMED FOR THE RESULT OF USING THIS COMPONENT.

THIS COMPONENT IS LICENSED UNDER THE CREATIVE COMMONS ATTRIBUTION-SHAREALIKE LICENSE. FOR THE FULL LICENSE TEXT PLEASE VISIT: http://creativecommons.org/licenses/by-sa/2.5/

AddFiles

Description

Add files to a new or an existing Zip file archive.

Returns

True, if files been add to Zip file; False, otherwise.

Function Syntax

```
AddFiles(zipFilePath [, files, directory, filter, recurse, compression, savePaths])
```

See also

DeleteFiles, Extract, List

Arguments

Argument	Req/Opt	Type	Default	Description
zipFilePath	Required	string		Pathname of the Zip file to add files.
files	Optional	string		| *(Chr(124))* delimited list of files to add to the Zip file. Required if argument `directory` is not set.
directory	Optional	string		Absolute pathname of directory to add to the Zip file. Required if argument `files` is not set.
filter	Optional	string		File extension filter. One filter can be applied. Only if argument `directory` is set.
recurse	Optional	boolean	no	Get recursive files of subdirectories. Only if argument `directory` is set.
compression	Optional	numeric	9	Compression level (0 through 9, 0=minimum, 9=maximum).
savePaths	Optional	boolean	no	Save full path info.

Usage

Use the `recurse` argument carefully. If your specified directory have many subdirectorys or files your system may hang up.

Notice: Special characters in filenames will not be supported. Filenames including special characters will be false encoded. That is a general problem on the Java Zip API.

Example

```
<h3>AddFiles Example</h3>

<cfscript>

 /* Create an instance of a component object */
 zip = CreateObject("component", "Zip");

 /* Add files to Zip file */
 status = zip.AddFiles("c:\myFile.zip", "myFile_1.txt|myFile_2.txt|myFile_3.txt");

</cfscript>

<cfdump var="#status#">
```

DeleteFiles

5

Description

Delete files from an existing Zip file archive.

Returns

True, if files been deleted from the Zip file; False, otherwise.

Function Syntax

```
DeleteFiles(zipFilePath, files)
```

See also

AddFiles, Extract, List

Arguments

Argument	Req/Opt	Type	Default	Description
zipFilePath	Required	string		Pathname of the Zip file to delete files from.
files	Required	string			(Chr(124)) delimited list of files to delete from Zip file.

Usage

To delete a file from a Zip file, you have to specified the full entryname. The entryname contains the filename and the path info stored in the Zip file. To get the entryname you can also use the `List()` function.

Notice: Because there is no function in the Java Zip API to delete files from a Zip file, a workaround been created. At first a new temporary Zip file will be created, where all files excluded the deleted will be saved. Then the orginal Zip file will be deleted and the temporary Zip file be renamed.

Example

```
<h3>DeleteFiles Example</h3>

<cfscript>

 /* Create an instance of a component object */
 zip = CreateObject("component", "Zip");

 /* Delete files from Zip file */
 status = zip.DeleteFiles("c:\myFile.zip", "myFile_1.txt|myFolder\myFile_2.txt");

</cfscript>

<cfdump var="#status#">
```

Extract

Description

Extracts a specified Zip file into a specified directory.

Returns

True, if all files has been extract; False, otherwise.

Function Syntax

```
Extract(zipFilePath [, extractPath, extractFiles, useFolderNames, overwriteFiles])
```

See also

AddFiles, DeleteFiles, List

Arguments

Argument	Req/Opt	Type	Default	Description
zipFilePath	Required	string		Pathname of the Zip file to extract.
extractPath	Optional	string	*{local Directory}*	Pathname to extract the Zip file to.
extractFiles	Optional	string		I *(Chr(124))* delimited list of files to extract.
useFolderNames	Optional	boolean	yes	Create folders using the path info stored in the Zip file.
overwriteFiles	Optional	boolean	no	Overwrite existing files.

Usage

Notice: Because the special character encoding is a general problem on the Java Zip API, the extraction of all files including special characters in their names will be skiped.

Example

```
<h3>Extract Example</h3>

<cfscript>

 /* Create an instance of a component object */
 zip = CreateObject("component", "Zip");

 /* Extract Zip file */
 status = zip.Extract("c:\myFile.zip", "c:\myFolder");

</cfscript>

<cfdump var="#status#">
```

List

Description

List the content of a specified Zip file.

Returns

An query.

Function Syntax

```
List(zipFilePath)
```

See also

AddFiles, DeleteFiles, Extract

Arguments

Argument	Req/Opt	Type	Default	Description
zipFilePath	Required	string		Pathname of the Zip file to list the content.

Usage

`List` returns the following result columns:

- `entry`   The name of Zip file entry.
- `date`   The modification date/time of the entry.
- `size`   The uncompressed size of the entry data.
- `packed` The compressed size of the entry data.
- `ratio` The compression ratio in percent.
- `crc`   The CRC checksum of the uncompressed entry data.

Notice: Special characters in file/pathnames will not be displayed.

Example

```
<h3>List Example</h3>

<cfscript>

 /* Create an instance of a component object */
 zip = CreateObject("component", "Zip");

 /* Get Zip file content */
 content = zip.List("c:\myFile.zip");

</cfscript>

<cfdump var="#content#">
```