

APL64: Using ZIP

Table of Contents

Overview	1
Syntax	1
Actions	2
CreateFromDirectory.....	2
ExtractFile	2
ExtractFiles	4
ExtractToDirectory.....	5
FileProperties	7
GetAllFilePaths.....	8
Help(?).....	9
InsertFile.....	10
InsertFiles	11
RemoveAll.....	11
RemoveFile	12
RemoveFiles	13

Overview

The  ZIP APL64 system function is an interface to the cross-platform compatible [.Net System.IO.Compression](#) tools. The  ZIP system function actions involve zip-format files. Since file operations are involved, the user's workstation credentials must be appropriate for the actions to be performed by .

Microsoft System.IO.Compression toolkit applies case-sensitivity when identifying paths of files within a zip-format file. Setting path case from system function  PATHCASE has no effect on .

Microsoft System.IO.Compression toolkit does not support zip-format files with passwords.

Some of the examples in this document utilize an APL64 multi-line executable statement.

Syntax

[result] ← [lArg]  ZIP rArg

The type and structure of result, lArg and rArg depend on the programmer-selected ☐ ZIP action. The action element is the first element of rArg. The action element is a case-insensitive text element. 'Text' arguments can be character or string data types.

Actions

CreateFromDirectory

Create a zip-format file containing the files in the source folder and its sub-folders. The ZipFileName file cannot exist. The zipFileName folder must exist.

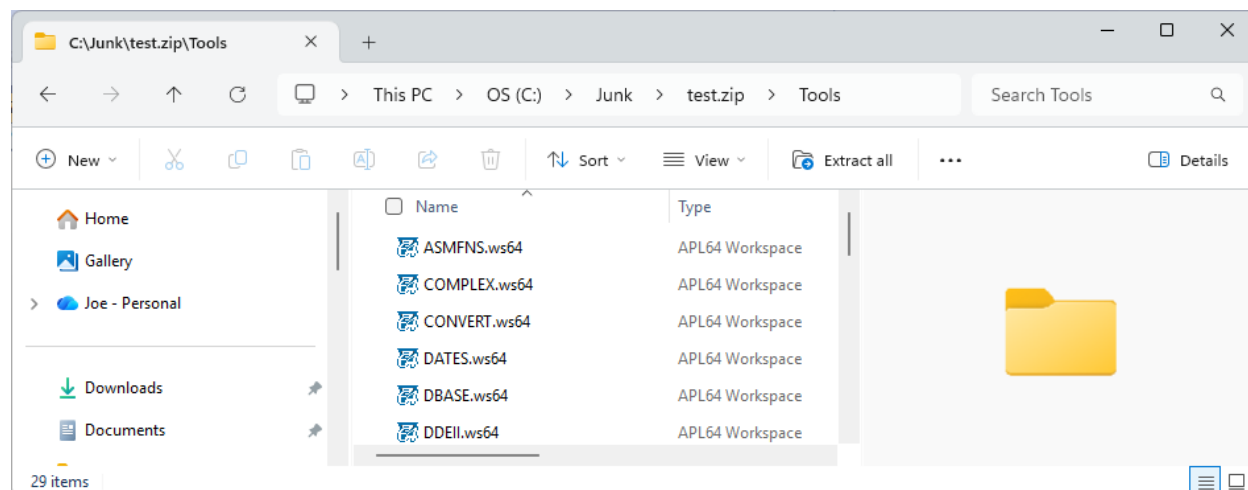
zipFileName ☐ ZIP 'CreateFromDirectory' srcDirName compLevel inclBaseDir [overwrite]

Argument	Type	Description
CreateFromDirectory	Text	Action
zipFileName	Text	Zip file to create
sourceDirectory	Text	Source directory path
compLevel	Text	Fastest, NoCompression, Optimal, SmallestSize
inclBaseDir	Boolean	Include base directory of sourceDirectory in zip file paths
overwrite	Boolean	Optional: Overwrite target file if it exists, default is false

Example:

```
sourceDirectory<'C:\Users\joebl\AppData\Roaming\APLNowLLC\APL64\Tools'
'c:\junk\test.zip' ☐ ZIP 'CreateFromDirectory' sourceDirectory 'smallestSize' 1 1
```

After execution:



ExtractFile

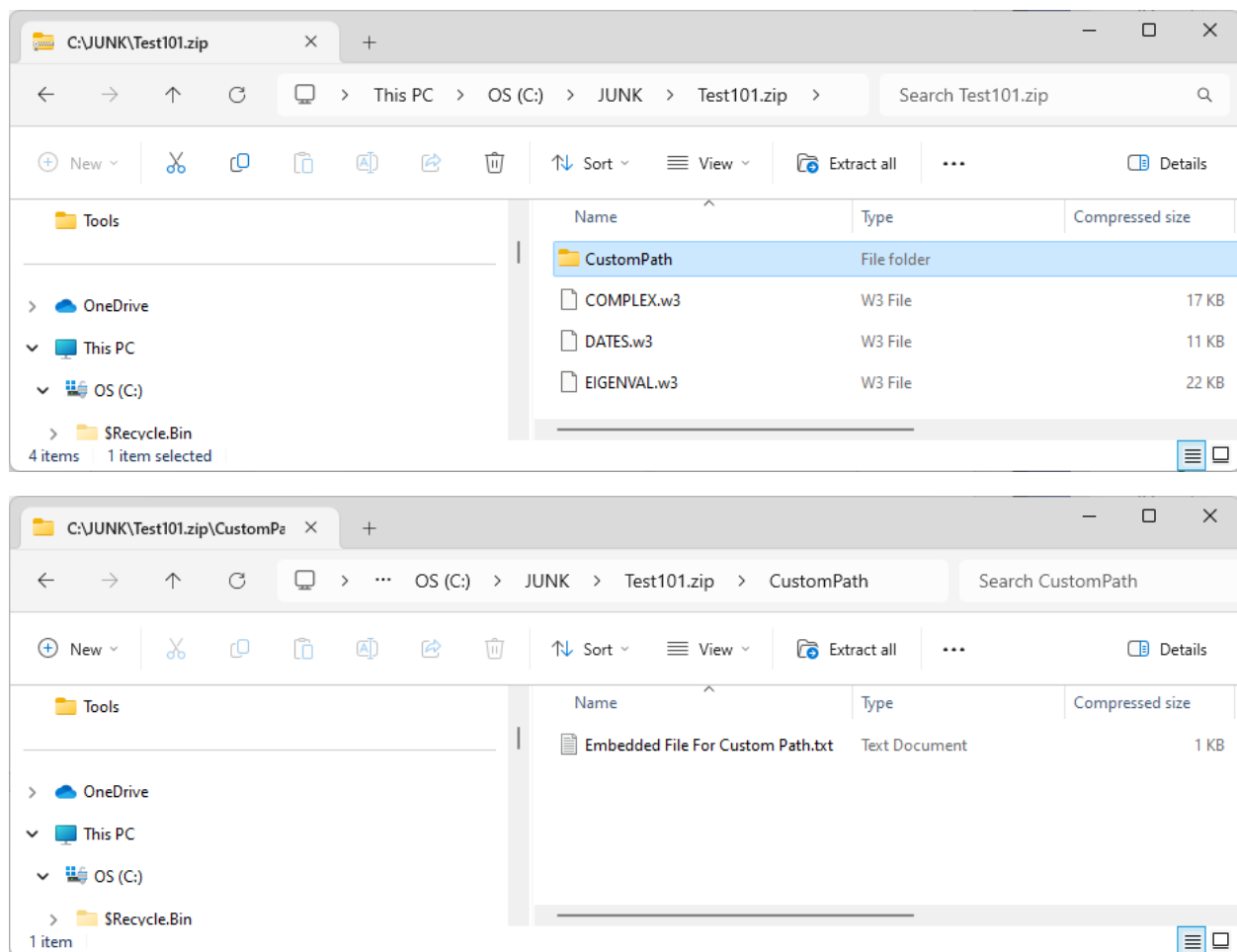
Extracts a file in a zip-format file to a target file.

zipFileName ☐ ZIP 'ExtractFile' zippedFilePath destFileName overWrite

Argument	Type	Description
ExtractFile	Text	Action
zipFileName	Text	Source zip-format file
zippedFilePath	Text	Path, relative to the root of the archive, that identifies the entry
destFileName	Text	File name to contain the entry content
overWrite	Boolean	Overwrite file if it exists

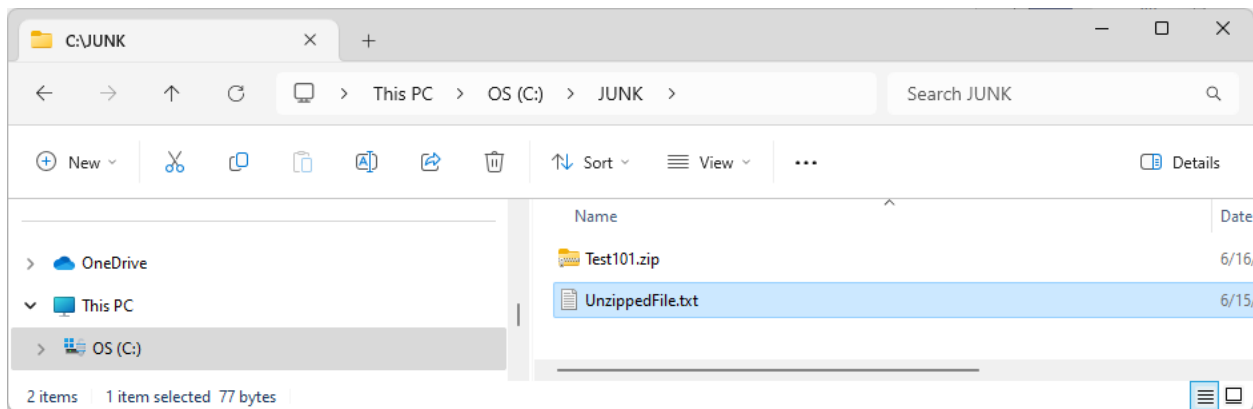
Example:

Existing zip-format file:



```
zippedFilePath< 'CustomPath/Embedded File For Custom Path.txt'  
destFileName<'C:\Junk\UnzippedFile.txt'  
'C:\Junk\Test101.zip' ☐ Zip 'ExtractFile' zippedFilePath destFileName 1
```

After execution:



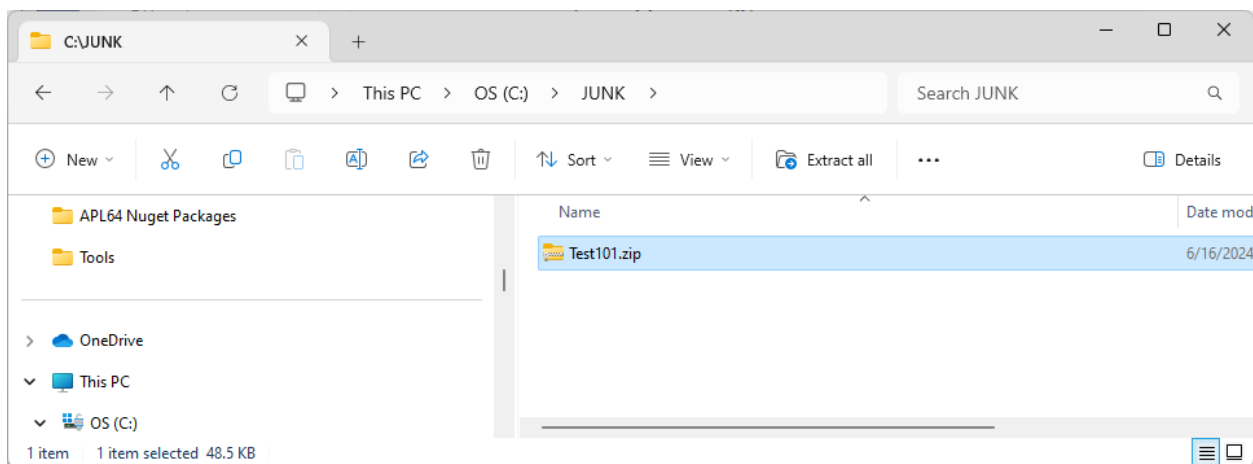
ExtractFiles

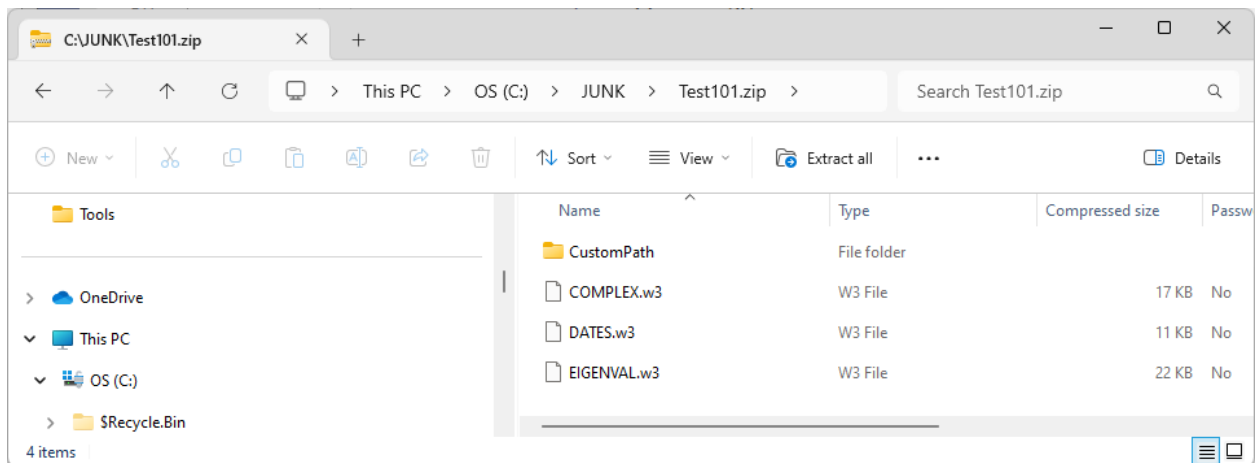
Extracts specific entry in a specified zip-format file to a specified file.

zipFileName ☐ ZIP 'ExtractFiles' zippedFilePaths destFileNames overWrite

Argument	Type	Description
ExtractFiles	Text	Action
zipFileName	Text	Source zip-format file
zippedFilePaths	String[]	Paths, relative to the root of the archive, that identifies the files to be extracted
destFileNames	String[]	Files name to contain the contents of the files to be extracted
overWrite	Boolean	Overwrite file if it exists

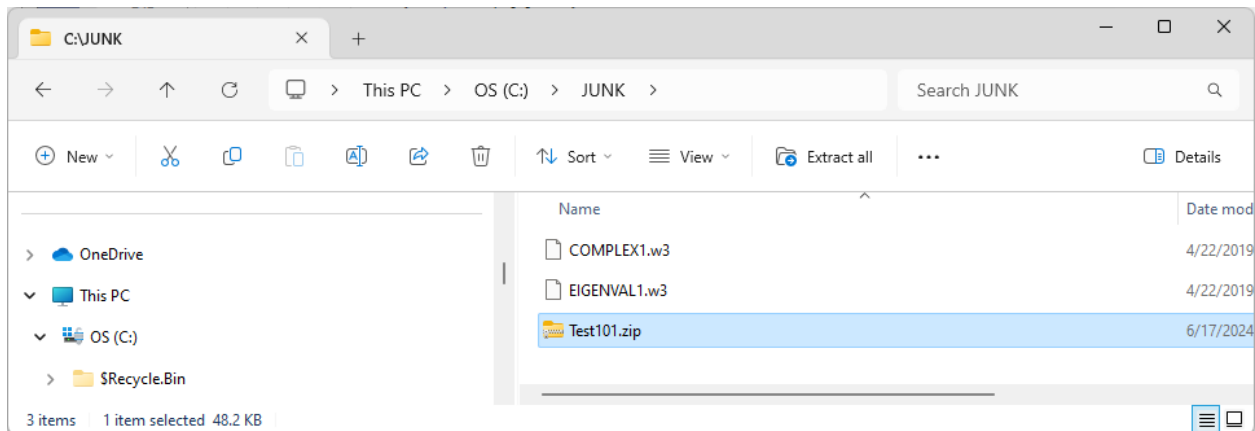
Before execution:





```
zipFileName←'C:\JUNK\Test101.zip'
zippedFilePaths←<'COMPLEX.w3' 'EIGENVAL.w3'
destFileNames←<'C:\JUNK\COMPLEX1.w3' 'C:\JUNK\EIGENVAL1.w3'
overWrite←1
zipFileName ☐ Zip 'ExtractFiles' zippedFilePaths destFileNames overWrite
```

After execution:



ExtractToDirectory

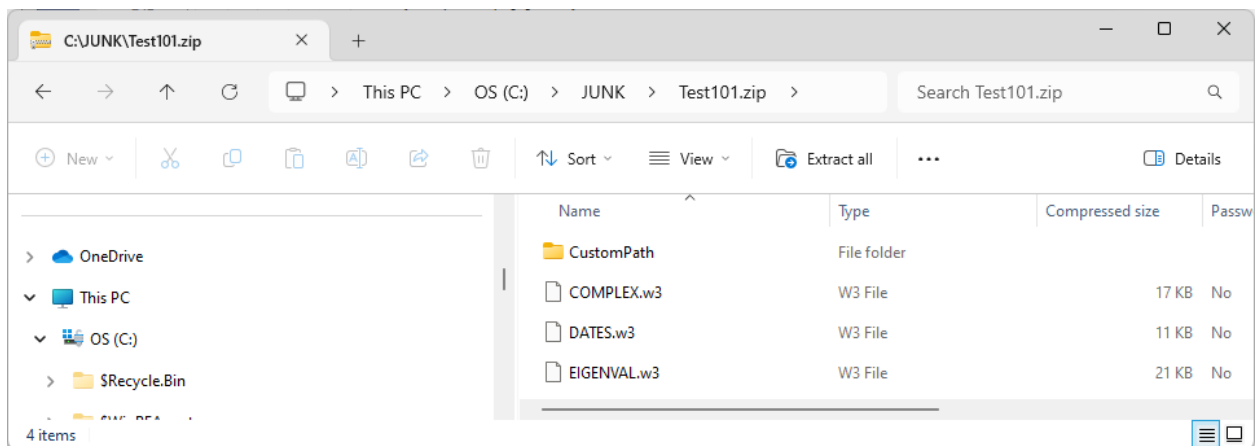
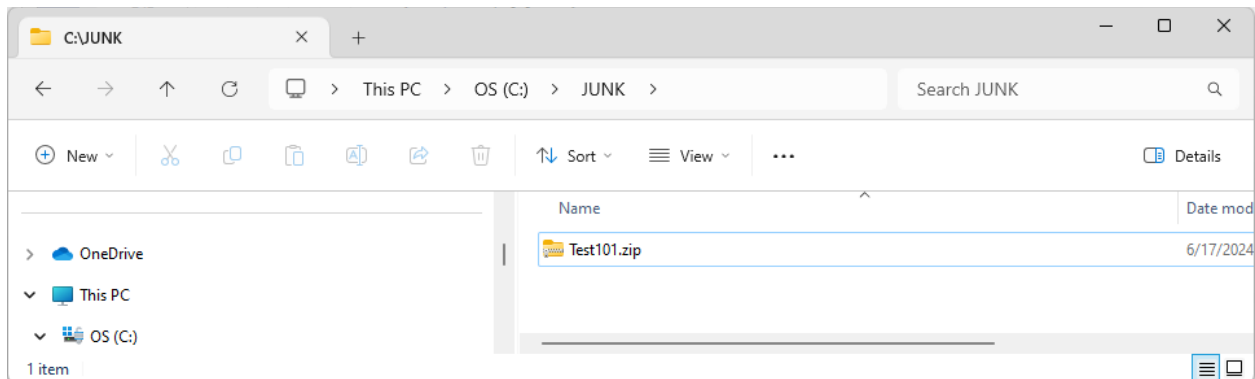
Extracts all the files in a specified zip-format file to a specified folder. The target folder(s) will be created if necessary. Extraction will continue until all files in the source zip-format file have been extracted or an exception occurs.

zipFileName ☐ ZIP 'ExtractToDirectory' destFolder overWrite

Element	Type	Description
ExtractToDirectory	Text	Action
zipFileName	Text	Source zip-format file
destFolder	Text	Target Folder
overWrite	Boolean	Overwrite files in target folder

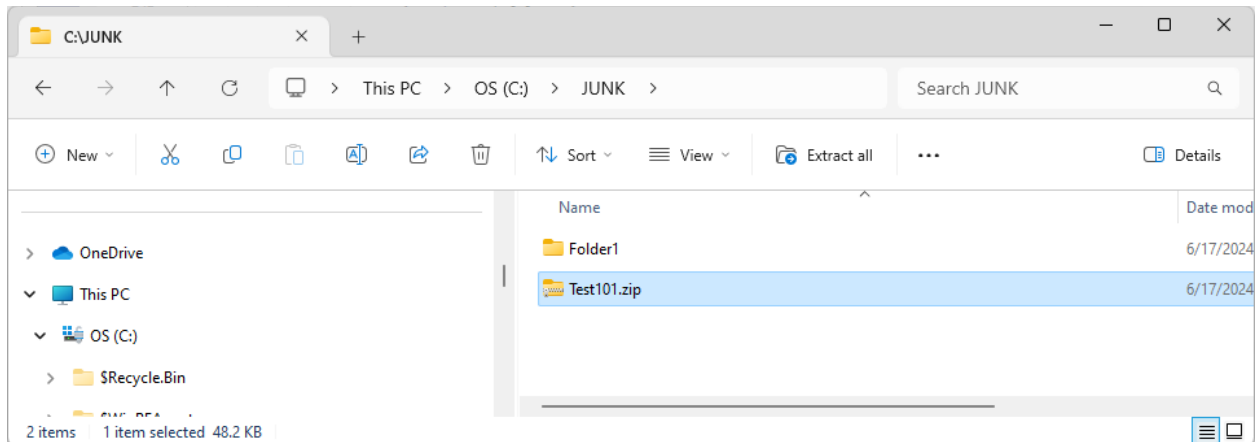
Example:

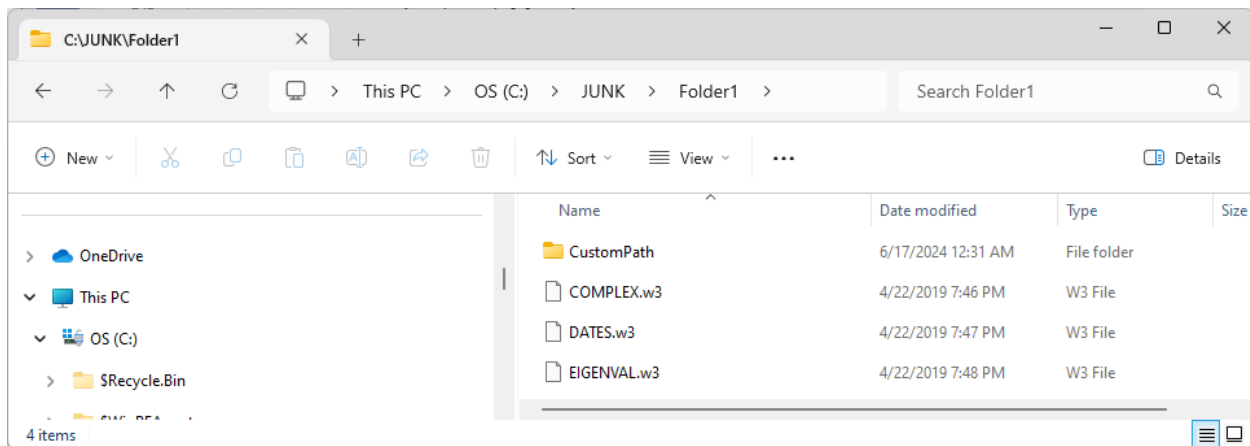
Before execution:



```
zipFileName<'C:\JUNK\Test101.zip'  
destFolder<'C:\JUNK\Folder1'  
overWrite<1  
zipFileName ☐ Zip 'ExtractToDirectory' destFolder overWrite
```

After execution:





FileProperties

Returns a vector of the available zip-format file entry properties.

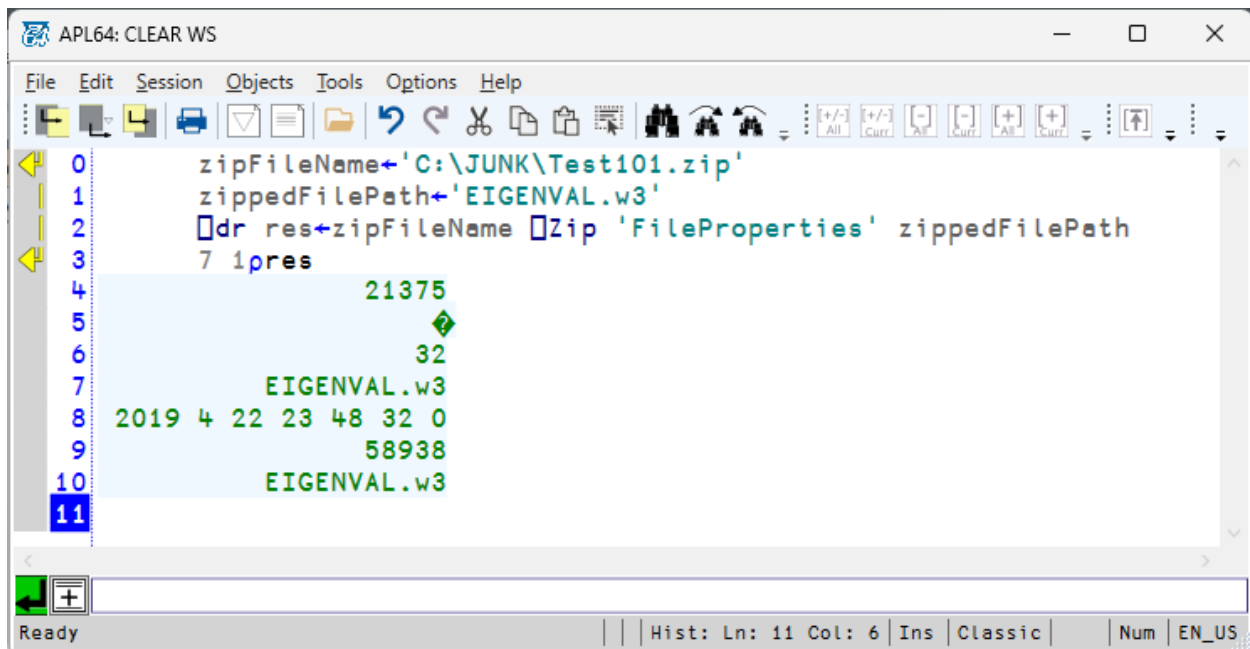
hetVector ← zipFileName ☐ Zip 'FileProperties' zippedFilePath

Argument	Type	Description
FileProperties	Text	Action
zipFileName	Text	Source zip-format file
zippedFilePath	Text	Path, relative to the root of the archive, that identifies the entry

File Property	Result Item#	Description
Compressed Length	1	Compressed #bytes
Crc32	2	Unsigned integer (4 bytes)
External Attributes	3	Int32: Host OS and Application dependent
Full Name	4	Relative path of the entry in the zip file
Last Write Time	5	Utc DateTime in ☐ TS format
Length	6	Uncompressed #bytes
Name	7	Name of entry in zip file without path

Example:

```
zipFileName←'C:\JUNK\Test101.zip'
zippedFilePath←'EIGENVAL.w3'
☐dr res←zipFileName ☐Zip 'FileProperties' zippedFilePath
7 1pres
```



GetAllFilePaths

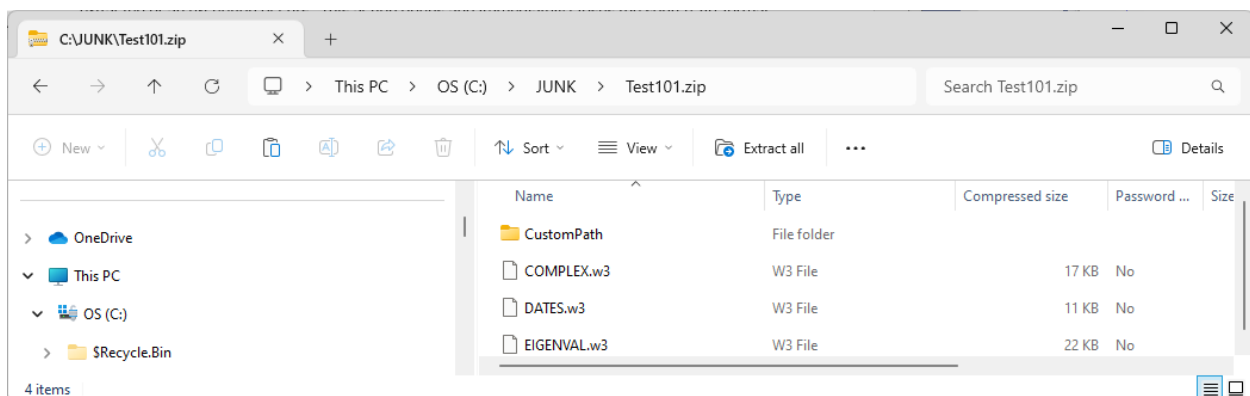
Get the zipped file paths for all files in a zip-format file. The result is a String[] vector of file paths relative to the root of the zip-format file.

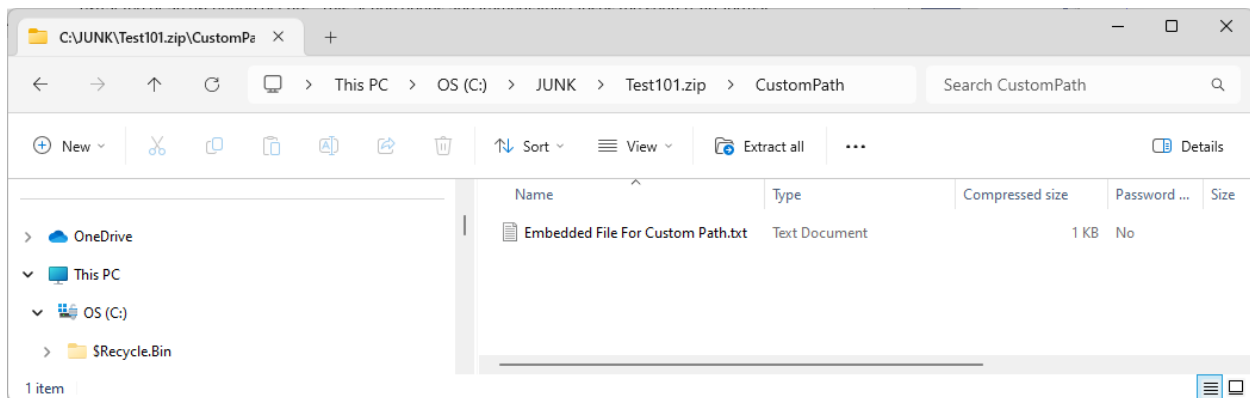
String[] ← zipFileName Zip 'GetAllFilePaths'

Argument	Type	Description
GetAllFilePaths	Text	Action
zipFileName	Text	Zip-format file

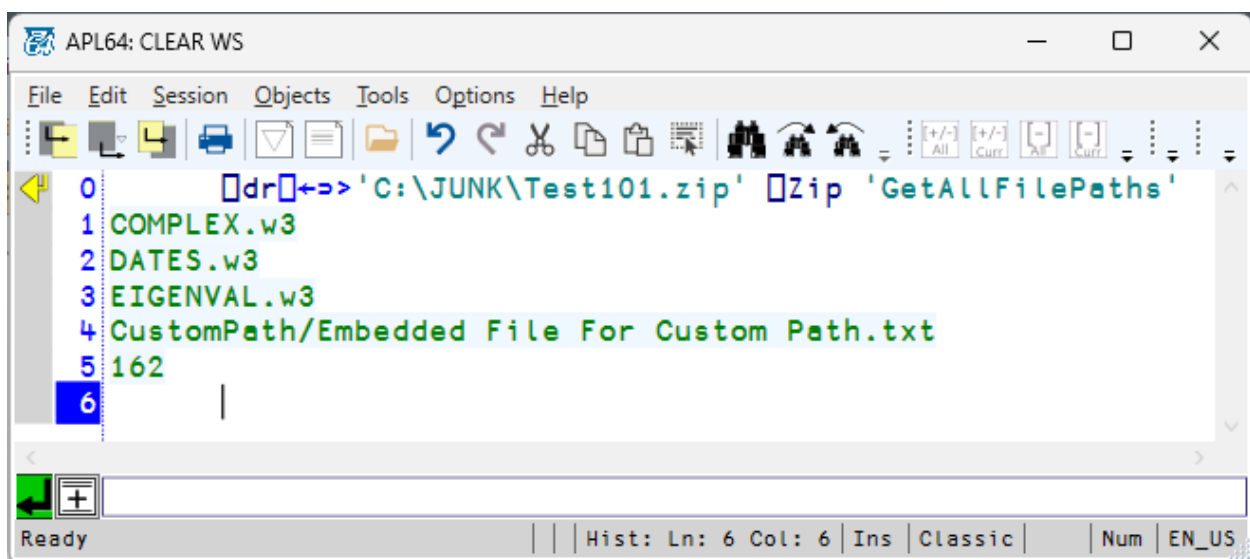
Example:

An existing zip-format file:





```
dr<=>'C:\JUNK\Test101.zip' Zip 'GetAllFilePaths'
```



Help(?)

Returns a character array with a summary of the Zip documentation.

```
charMat ← Zip ''
```

```
charMat ← Zip 'Help'
```

```

0  ⍺zip '?'
1  ⍺ZIP Documentation Summary
2      zipFileName ⍺ZIP 'CreateFromDirectory' srcDirName compLevel inclBaseDir [overWrite]
3      zipFileName ⍺ZIP 'ExtractFile' zippedFilePath destFileName overWrite
4      zipFileName ⍺ZIP 'ExtractFiles' zippedFilePaths destFileNames overWrite
5      zipFileName ⍺ZIP 'ExtractToDirectory' destFolder overWrite
6  hetVector ← zipFileName ⍺Zip 'FileProperties' zippedFilePath
7  String[] ← zipFileName ⍺Zip 'GetAllFilePaths'
8  CharMat ←      ⍺Zip 'Help'
9  CharMat ←      ⍺Zip '?'
10      zipFileName ⍺Zip 'InsertFile' zippedFilePath sourceFileName compLevel
11      ZipFileName ⍺Zip 'InsertFiles' zippedFilePaths sourceFileNames compLevel
12      zipFileName ⍺Zip 'RemoveAll'
13      zipFileName ⍺Zip 'RemoveFile' zippedFilePath
14      zipFileName ⍺Zip 'RemoveFilew' zippedFilePathw
15
16  compLevel: Fastest NoCompression Optimal SmallestSize
17  string[] : zippedFilePaths, destFileNames, sourceFileNames
18

```

InsertFile

Insert a source file into a zip.format file. If the ZipFileName file does not exist it will be created.

zipFileName ⍺Zip 'InsertFile' zippedFilePath sourceFileName compLevel

Argument	Type	Description
InsertFile	Text	Action
zipFileName	Text	Zip-format file to contain entry
zippedFilePath	Text	Zipped file path, relative to the root of the zip file, to insert
sourceFileName	Text	Source file path
compLevel	Text	Fastest, NoCompression, Optimal, SmallestSize

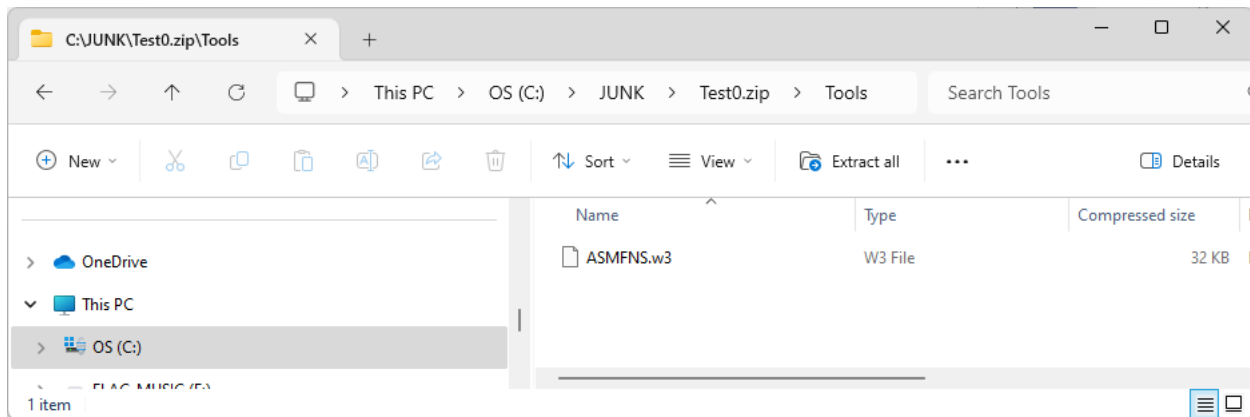
Example:

```

targetEntryId←'Tools\ASMFNS.w3'
sourceFileName←'C:\APLWIN19\Tools\ASMFNS.w3'
'C:\JUNK\Test0.zip' ⍺Zip 'InsertFile' targetEntryId sourceFileName 'Fastest'

```

After execution:



InsertFiles

Insert source files into a zip.format file. If the ZipFileName does not exist it will be created.

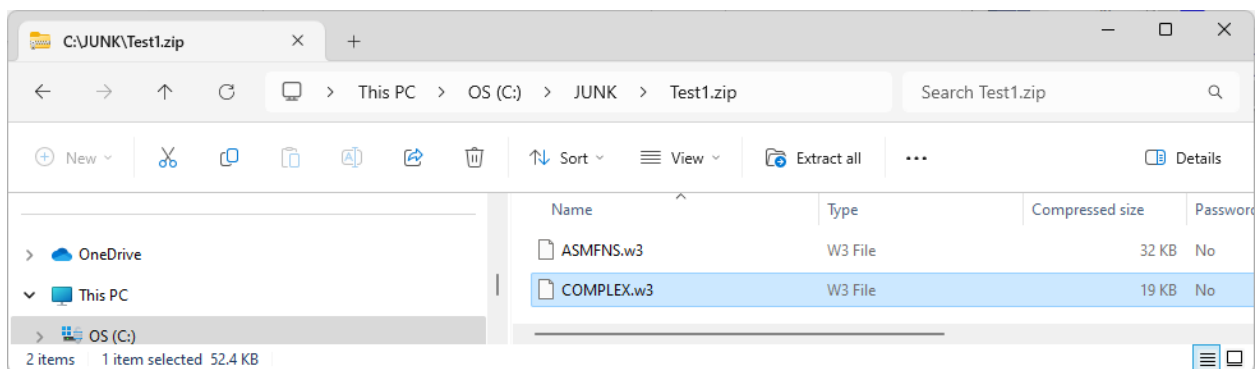
ZipFileName ☐ Zip 'InsertFiles' zippedFilePaths sourceFileNames compLevel

Argument	Type	Description
InsertFile	Text	Action
zipFileName	Text	Zip-format file to contain entry
zippedFilePaths	String[]	Zipped file paths, relative to the root of the zip file, to insert
sourceFileNames	String[]	Source file paths
compLevel	text	Fastest, NoCompression, Optimal, SmallestSize

Example:

```
targetEntryIds<«ASMFNS.w3» «COMPLEX.w3»
sourceFileNames<<'C:\APLWIN19\Tools\ASMFNS.w3' 'C:\APLWIN19\Tools\COMPLEX.w3'
'C:\JUNK\Test1.zip' ☐ Zip 'InsertFiles' targetEntryIds sourceFileNames 'Fastest'
```

After execution:



RemoveAll

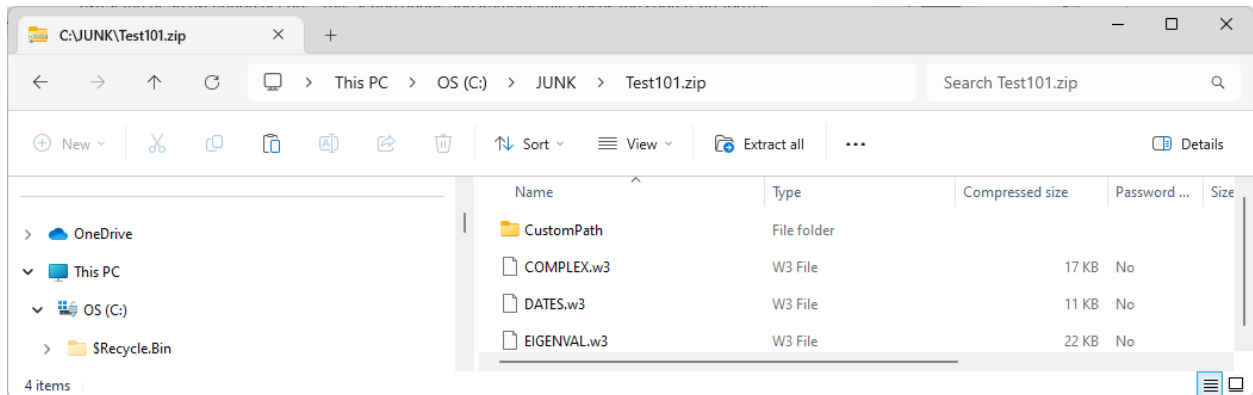
Removes all files in a zip-format file.

zipFileName ☐ Zip 'RemoveAll'

Argument	Type	Description
RemoveFile	Text	Action
zipFileName	Text	Zip-format file containing the file to remove

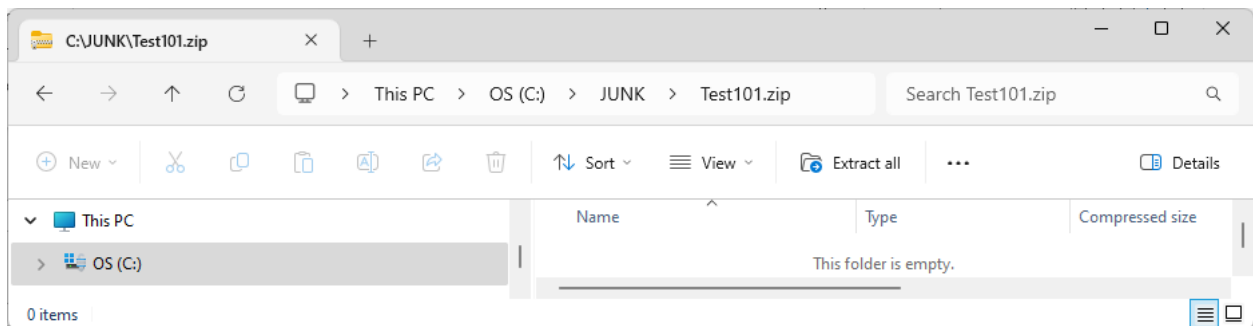
Example:

Before execution, using an existing zip-format file:



zipFileName ☐ Zip 'RemoveAll'

After execution:



RemoveFile

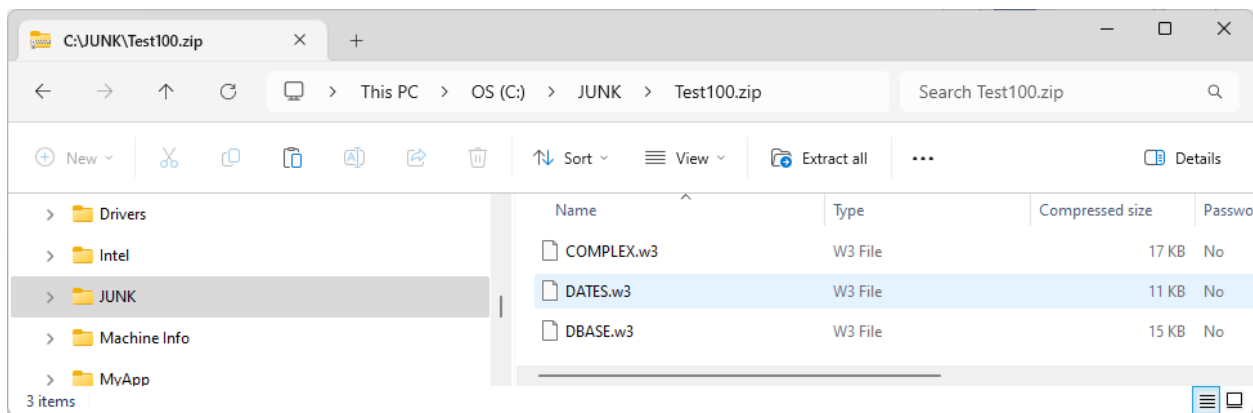
Remove a file from a zip-format file.

zipFileName ☐ Zip 'RemoveFile' zippedFilePath

Argument	Type	Description
RemoveFile	Text	Action
zipFileName	Text	Zip-format file containing the file to remove
zippedFilePath	Text	Zipped file, relative to the root of the zip file, to remove

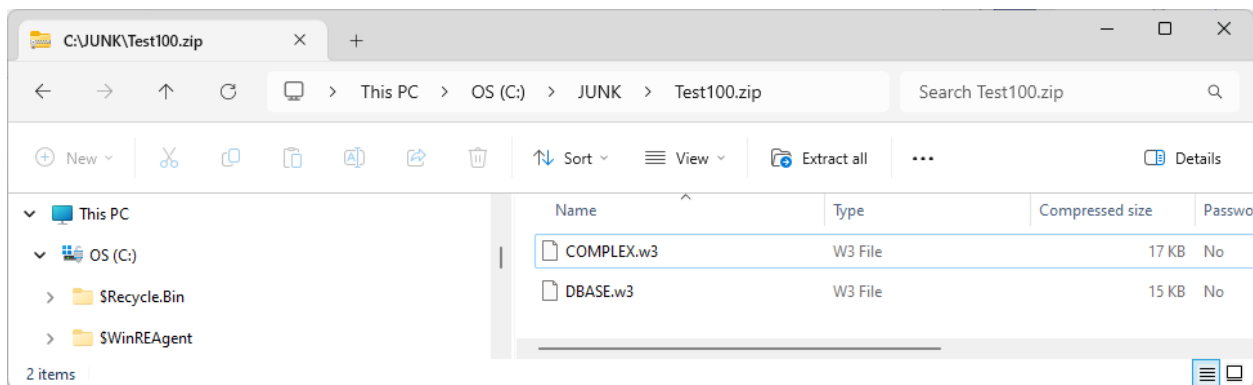
Example:

Before execution, using an existing zip-format file:



```
'c:\junk\Test100.zip' -Zip 'RemoveFile' 'DATES.w3'
```

After execution:



RemoveFiles

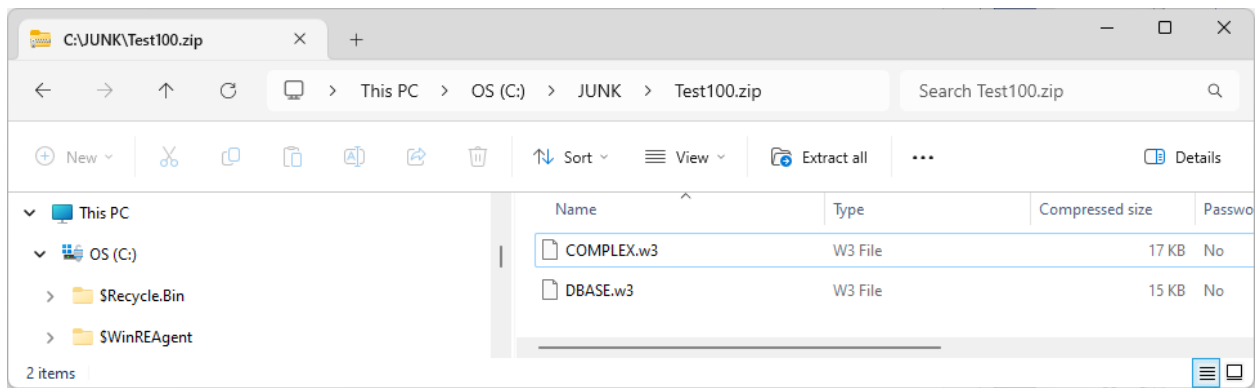
Remove files from a zip-format file. The remove process will continue until all specified zipped files have been removed or an exception is thrown.

zipFileName ☐ Zip 'RemoveFiles' targetEntryIds

Argument	Type	Description
RemoveFiles	Text	Action
zipFileName	Text	Zip-format file containing the file to remove
zippedFilePaths	String[]	Zipped files, relative to the root of the zip file, to remove

Example:

Before execution, using an existing zip-format file:



'c:\junk\Test100.zip' ☐ Zip 'RemoveFiles' (<'COMPLEX.w3' 'DBASE.w3')

After execution:

