



ArchiveFileStreamContext Class

Provides a basic implementation of the archive pack and unpack stream context interfaces, based on a list of archive files, a default directory, and an optional mapping from internal to external file paths.

Methods

Method	Description
<code>CloseArchiveReadStream(archiveNumber, archiveName, stream)</code>	Closes a stream where an archive was read.
<code>CloseArchiveWriteStream(archiveNumber, archiveName, stream)</code>	Closes a stream where an archive package was written.
<code>CloseFileReadStream(path, stream)</code>	Closes a stream that has been used to read a file.
<code>CloseFileWriteStream(path, stream, attributes, lastWriteTime)</code>	Closes a stream where an extracted file was written.
<code>GetArchiveName(archiveNumber)</code>	Gets the name of the archive with a specified number.
<code>GetOption(optionName, parameters)</code>	Gets extended parameter information specific to the compression format being used.

Method	Description
<code>OpenArchiveReadStream/archiveNumber, archiveName, compressionEngine)</code>	Opens the archive stream for reading.
<code>OpenArchiveWriteStream/archiveNumber, archiveName, truncate, compressionEngine)</code>	Opens a stream for writing an archive.
<code>OpenFileWriteStream(path, fileSize, lastWriteTime)</code>	Opens a stream for writing extracted file bytes.

Properties

Property	Description
<code>ArchiveFiles</code>	Gets or sets the list of archive files that are created or extracted.
<code>Directory</code>	Gets or sets the default root directory where files are located.
<code>EnableOffsetOpen</code>	Gets or sets a flag that enables creating or extracting an archive at an offset within an existing file. (This is typically used to open archive-based self-extracting packages.)
<code>ExtractOnlyNewerFiles</code>	Gets or sets a flag that can prevent extracted files from overwriting newer files that already exist.
<code>Files</code>	Gets or sets the mapping from internal file paths to external file paths.

Remarks

This class can also handle creating or extracting chained archive packages. `WixToolset.Dtf.Compression.dll` version `4.0.2+644ed0118bae1aa885bb6cc906dc6c0b904de4d9`

CloseArchiveReadStream(*archiveNumber*, *archiveName*, *stream*) Method

Closes a stream where an archive was read.

Declaration

```
public void CloseArchiveReadStream(  
    int archiveNumber,  
    string archiveName,  
    System.IO.Stream stream  
)
```

Parameters

Parameter	Type	Description
archiveNumber	int	The archive number of the stream to close.

Parameter	Type	Description
archiveName	string	The name of the archive being closed.
stream	System.IO.Stream	The stream that was previously returned by and is now ready to be closed.

CloseArchiveWriteStream/archiveNumber, archiveName, stream) Method

Closes a stream where an archive package was written.

Declaration

```
public void CloseArchiveWriteStream(  
    int archiveNumber,  
    string archiveName,  
    System.IO.Stream stream  
)
```

Parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
archiveNumber	int	The 0-based index of the archive within the chain.
archiveName	string	The name of the archive that was previously returned by .
stream	System.IO.Stream	A stream that was previously returned by and is now ready to be closed.

CloseFileReadStream(path, stream) Method

Closes a stream that has been used to read a file.

Declaration

```
public void CloseFileReadStream(  
    string path,  
    System.IO.Stream stream  
)
```

Parameters

Parameter	Type	Description
path	string	The path of the file within the archive; the same as the path provided when the stream was

Parameter	Type	Description
		opened.
stream	System.IO.Stream	A stream that was previously returned by and is now ready to be closed.

CloseFileWriteStream(path, stream, attributes, lastWriteTime) Method

Closes a stream where an extracted file was written.

Declaration

```
public void CloseFileWriteStream(  
    string path,  
    System.IO.Stream stream,  
    System.IO.FileAttributes attributes,  
    System.DateTime lastWriteTime  
)
```

Parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
path	string	The path of the file within the archive.
stream	System.IO.Stream	The stream that was previously returned by and is now ready to be closed.
attributes	System.IO.FileAttributes	The attributes of the extracted file.
lastWriteTime	System.DateTime	The last write time of the file.

Remarks

After closing the extracted file stream, this method applies the date and attributes to that file.

GetArchiveName(archiveNumber) Method

Gets the name of the archive with a specified number.

Declaration

```
public string GetArchiveName(  
    int archiveNumber  
)
```

Parameters

Parameter	Type	Description
archiveNumber	int	The 0-based index of the archive within the chain.

Return value

`string` The name of the requested archive. May be an empty string for non-chained archives, but may never be null.

Remarks

This method returns the file name of the archive from the «see `F:\WixToolset.Dtf.Compression.ArchiveFileStreamContext.archiveFiles`» list with the specified index, or an empty string if the archive number is outside the bounds of the list. The file name should not include any directory path.

GetOption(optionName, parameters) Method

Gets extended parameter information specific to the compression format being used.

Declaration

```
public System.Object GetOption(  
    string optionName,  
    System.Object[] parameters
```



```
)
```

Parameters

Parameter	Type	Description
optionName	string	Name of the option being requested.
parameters	System.Object[]	Parameters for the option; for per-file options, the first parameter is typically the internal file path.

Return value

`System.Object` Option value, or null to use the default behavior.

Remarks

This implementation does not handle any options. Subclasses may override this method to allow for non-default behavior.

OpenArchiveReadStream(**archiveNumber**, **archiveName**, **compressionEngine**) Method

Opens the archive stream for reading.

Declaration

```
public System.IO.Stream OpenArchiveReadStream(  
    int archiveNumber,  
    string archiveName,  
    CompressionEngine compressionEngine  
)
```

Parameters

Parameter	Type	Description
archiveNumber	int	The zero-based index of the archive to open.
archiveName	string	The name of the archive being opened.
compressionEngine	CompressionEngine	Instance of the compression engine doing the operations.

Return value

`System.IO.Stream` A stream from which archive bytes are read, or null to cancel extraction of the archive.

Remarks

This method opens the file from the «see `P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.ArchiveFiles`» list with the specified index. If the archive number is outside the bounds of the list, this method returns null.If the «see

P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.EnableOffsetOpen» flag is set, this method will seek to the start of any existing archive in the file, or to the end of the file if the existing file is not an archive.

OpenArchiveWriteStream(archiveNumber, archiveName, truncate, compressionEngine) Method

Opens a stream for writing an archive.

Declaration

```
public System.IO.Stream OpenArchiveWriteStream(  
    int archiveNumber,  
    string archiveName,  
    bool truncate,  
    CompressionEngine compressionEngine  
)
```

Parameters

Parameter	Type	Description
archiveNumber	int	The 0-based index of the archive within the chain.
archiveName	string	The name of the archive that was returned by .

Parameter	Type	Description
truncate	bool	True if the stream should be truncated when opened (if it already exists); false if an existing stream is being re-opened for writing additional data.
compressionEngine	CompressionEngine	Instance of the compression engine doing the operations.

Return value

`System.IO.Stream` A writable Stream where the compressed archive bytes will be written, or null to cancel the archive creation.

Remarks

This method opens the file from the «see `P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.ArchiveFiles`» list with the specified index. If the archive number is outside the bounds of the list, this method returns null.If the «see `P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.EnableOffsetOpen`» flag is set, this method will seek to the start of any existing archive in the file, or to the end of the file if the existing file is not an archive.

OpenFileWriteStream(path, fileSize, lastWriteTime) Method

Opens a stream for writing extracted file bytes.

Declaration

```
public System.IO.Stream OpenFileWriteStream(
```

```
string path,  
System.Int64 fileSize,  
System.DateTime lastWriteTime  
)
```

Parameters

Parameter	Type	Description
path	string	The path of the file within the archive.
fileSize	System.Int64	The uncompressed size of the file to be extracted.
lastWriteTime	System.DateTime	The last write time of the file to be extracted.

Return value

`System.IO.Stream` A stream where extracted file bytes are to be written, or null to skip extraction of the file and continue to the next file.

Remarks

This method opens a file using the following logic:

- If the «see P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.Directory» and the «see P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.Files» mapping are both null, the path is treated as relative to the

current directory, and that file is opened.

- If the «see P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.Directory» is not null but the «see P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.Files» mapping is null, the path is treated as relative to that directory, and that file is opened.
- If the «see P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.Directory» is null but the «see P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.Files» mapping is not null, the path parameter is used as a key into the mapping, and the resulting value is the file path that is opened, relative to the current directory (or it may be an absolute path). If no mapping exists, the file is skipped.
- If both the «see P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.Directory» and the «see P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.Files» mapping are specified, the path parameter is used as a key into the mapping, and the resulting value is the file path that is opened, relative to the specified directory (or it may be an absolute path). If no mapping exists, the file is skipped. If the «see P:WixToolset.Dtf.Compression.ArchiveFileStreamContext.ExtractOnlyNewerFiles» flag is set, the file is skipped if a file currently exists in the same path with an equal or newer write time.

ArchiveFiles Property

Gets or sets the list of archive files that are created or extracted.

Declaration

```
public IList<System.String> ArchiveFiles { get; set; }
```

Directory Property

Gets or sets the default root directory where files are located.

Declaration

```
public string Directory { get; set; }
```

Remarks

For details about how the default directory is used, see «see

M:WixToolset.Dtf.Compression.ArchiveFileStreamContext.OpenFileReadStream(System.String,System.IO.FileAttributes@,System.DateTime@)» and «see

M:WixToolset.Dtf.Compression.ArchiveFileStreamContext.OpenFileWriteStream(System.String,System.Int64,System.DateTime)» .

EnableOffsetOpen Property

Gets or sets a flag that enables creating or extracting an archive at an offset within an existing file. (This is typically used to open archive-based self-extracting packages.)

Declaration

```
public bool EnableOffsetOpen { get; set; }
```

ExtractOnlyNewerFiles Property

Gets or sets a flag that can prevent extracted files from overwriting newer files that already exist.

Declaration

```
public bool ExtractOnlyNewerFiles { get; set; }
```

Files Property

Gets or sets the mapping from internal file paths to external file paths.

Declaration

```
public System.Collections.Generic.IDictionary<System.String, System.String> Files { get; set; }
```

Remarks

For details about how the files mapping is used, see «see

M:WixToolset.Dtf.Compression.ArchiveFileStreamContext.OpenFileReadStream(System.String, System.IO.FileAttributes@, System.DateTime@)» and «see

M:WixToolset.Dtf.Compression.ArchiveFileStreamContext.OpenFileWriteStream(System.String, System.Int64, System.DateTime)» .