BasicDriver + const TYPE_EXTENSION + const TYPE_UTILITIES + const TYPE_PURE_PHP + const COMPRESSION_NONE + const COMPRESSION_WEAK + const COMPRESSION_AVERAGE + const COMPRESSION_STRONG + const COMPRESSION_MAXIMUM + const OPEN + const OPEN_ENCRYPTED et 11 de plus... + static \$typeLabels # \$fileName # \$format + checkAbility(\$ability) + __construct(\$archiveFile Name, \$format, \$password=null) + getArchiveInformation() + getFileNames() + isFileExists(\$fileName) + getFileData(\$fileName) + getFileContent(\$fileName) + getFileStream(\$fileName) + extractFiles(\$outputFolder, array \$files) + extractArchive(\$outputFolder) + deleteFiles(array \$files) + addFiles(array \$files) + addFileFromString (\$inArchiveName, \$content) + getComment() + setComment(\$comment) + static getDescription() + static isInstalled() + static getInstallationInstruction() + static getSupportedFormats() + static checkFormatSupport (\$format) static createArchive (array \$files, \$archiveFile Name, \$archiveFormat, \$compression Level=self::COMPRESSION_AVERAGE, \$password=null, \$fileProgressCallable=null) static createArchiveInString (array \$files, \$archiveFormat, \$compressionLevel=self::COMPRESSION _AVERAGE, \$password=null, \$fileProgressCallable=null) + static wrapStringInStream (\$string) BasicPureDriver + const TYPE + const PACKAGE_NAME + const MAIN CLASS + static isInstalled() + static getInstallationInstruction() NelexaZip + const PACKAGE_NAME + const MAIN_CLASS # \$files __construct(\$archiveFile Name, \$format, \$password=null)

Cab

- + const PACKAGE_NAME
- + const MAIN_CLASS
- # \$cab
- __construct(\$archiveFile Name, \$format, \$password=null)
- + __destruct()
- getArchiveInformation()
- + getFileNames()
- + isFileExists(\$fileName)
- + getFileData(\$fileName)
- + getFileContent(\$fileName)
- + getFileStream(\$fileName) + extractFiles(\$outputFolder,
- array \$files)
- + extractArchive(\$outputFolder) + static getDescription()
- + static getSupportedFormats()
- + static checkFormatSupport
- (\$format) # open(\$archiveFileName)

Iso

- + const PACKAGE NAME
- + const MAIN CLASS
- # \$iso
- # \$files
- # \$filesData
- # \$filesSize
- # \$blockSize
- + __construct(\$archiveFile Name, \$format, \$password=null)
- + __destruct()
- + getArchiveInformation()
- + getFileNames()
- + isFileExists(\$fileName)
- + getFileData(\$fileName)
- + getFileContent(\$fileName) + getFileStream(\$fileName)
- + extractFiles(\$outputFolder,
- array \$files) + extractArchive(\$outputFolder)
- + static getDescription()
- + static getSupportedFormats()
- + static checkFormatSupport
- (\$format) # prepareForFileExtracting (\$fileName)

- # \$zip

- + getArchiveInformation()
- + getFileNames()
- + isFileExists(\$fileName)
- + getFileData(\$fileName)
- + getFileContent(\$fileName)
- + getFileStream(\$fileName)
- + extractFiles(\$outputFolder, array \$files)
- + extractArchive(\$outputFolder)
- + addFileFromString (\$inArchiveName, \$content)
- + getComment()
- + setComment(\$comment)
- + deleteFiles(array \$files)
- + addFiles(array \$files)
- + static getDescription()
- + static getSupportedFormats()
- + static checkFormatSupport (\$format)
- static createArchive (array \$files, \$archiveFile Name, \$archiveFormat, \$compression Level=self::COMPRESSION_AVERAGE, \$password=null, \$fileProgressCallable=null)
- static createArchiveInString (array \$files, \$archiveFormat, \$compressionLevel=self::COMPRESSION _AVERAGE, \$password=null, \$fileProgressCallable=null)
- # static createArchiveInternal (array \$files, \$password, \$fileProgressCallable=null)

SplitbrainPhpArchive

- + const PACKAGE_NAME
- + const MAIN_CLASS
- # \$archive
- # \$files
- # \$members
- + __construct(\$archiveFile Name, \$format, \$password=null)
- + getArchiveInformation()
- + getFileNames()
- + isFileExists(\$fileName)
- + getFileData(\$fileName)
- + getFileContent(\$fileName)
- + getFileStream(\$fileName)
- + extractFiles(\$outputFolder,
- array \$files) + extractArchive(\$outputFolder)
- + static getDescription()
- + static getSupportedFormats()
- + static checkFormatSupport (\$format) + static createArchive
- (array \$files, \$archiveFile Name, \$archiveFormat, \$compression Level=self::COMPRESSION_AVERAGE, \$password=null, \$fileProgressCallable=null) + static createArchiveInString
- (array \$files, \$archiveFormat, \$compressionLevel=self::COMPRESSION _AVERAGE, \$password=null, \$fileProgressCallable=null)
- static createArchiveInternal (array \$files, \$archiveFileName, \$archiveFormat, \$compressionLevel =self::COMPRESSION_AVERAGE, \$fileProgressCallable=null)

TarByPear

- + const MAIN CLASS
- # \$pearCompressionRatio
- # \$pearFilesIndex
- # \$pureFilesNumber
- __construct(\$archiveFile Name, \$format, \$password=null)
- + getArchiveInformation() + getFileNames()
- + isFileExists(\$fileName)
- + getFileData(\$fileName)
- + getFileContent(\$fileName)
- + getFileStream(\$fileName)
- + extractFiles(\$outputFolder,
- array \$files)
- + extractArchive(\$outputFolder) + addFiles(array \$files)
- + addFileFromString
- (\$inArchiveName, \$content)
- + static getDescription()
- + static getInstallationInstruction() + static getSupportedFormats()
- static checkFormatSupport
- (\$format) static createArchive (array \$files, \$archiveFile Name, \$archiveFormat, \$compression Level=self::COMPRESSION_AVERAGE,
- \$password=null, \$fileProgressCallable=null) # open(\$archiveType)