

**xpkarchive**

**COLLABORATORS**

	<i>TITLE :</i> xpkarchive		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 1, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>xpkarchive</b>	<b>1</b>
1.1	xpkarchive.doc	1
1.2	xpkarchive.library/XarAddFileA	2
1.3	xpkarchive.library/XarCloseArchive	3
1.4	xpkarchive.library/XarCopyFileA	3
1.5	xpkarchive.library/XarExamine	4
1.6	xpkarchive.library/XarExtractFileA	5
1.7	xpkarchive.library/XarFindFileA	6
1.8	xpkarchive.library/XarGetFileData	7
1.9	xpkarchive.library/XarGetFileIndex	8
1.10	xpkarchive.library/XarGetFileName	8
1.11	xpkarchive.library/XarGetFileNote	9
1.12	xpkarchive.library/XarGetFileSize	9
1.13	xpkarchive.library/XarGetLock	10
1.14	xpkarchive.library/XarGetType	11
1.15	xpkarchive.library/XarIsXpkArchive	11
1.16	xpkarchive.library/XarModifyFileDataA	12
1.17	xpkarchive.library/XarNextGenerationA	13
1.18	xpkarchive.library/XarNextLock	14
1.19	xpkarchive.library/XarOpenArchiveA	14
1.20	xpkarchive.library/XarPrevLock	15
1.21	xpkarchive.library/XarRenameFileA	16
1.22	xpkarchive.library/XarSetFileNoteA	16
1.23	xpkarchive.library/--tags--	17
1.24	xpkarchive.library/--error--	21
1.25	xpkarchive.library/--generations--	21
1.26	xpkarchive.library/--BUGS--	21

---

# Chapter 1

## xpkarchive

### 1.1 xpkarchive.doc

XarAddFileA()  
XarCloseArchive()  
XarCopyFileA()  
XarExamine()  
XarExtractFileA()  
XarFindFileA()  
XarGetFileData()  
XarGetFileIndex()  
XarGetFileName()  
XarGetFileNote()  
XarGetFileSize()  
XarGetLock()  
XarGetType()  
XarIsXpkArchive()  
XarModifyFileDataA()  
XarNextGenerationA()  
XarNextLock()  
XarOpenArchiveA()  
XarPrevLock()

---

```

XarRenameFileA()

XarSetFileNoteA()

--tags--()

--error--()

--generations--()

--BUGS--

```

## 1.2 xpkarchive.library/XarAddFileA

NAME

XarAddFile

SYNOPSIS

```

arclock = XarAddFileA( tags )
D0                      A0

```

```

XarLock * XarAddFileA(struct TagItem *)

```

FUNCTION

XarAddFileA() adds a file to the archive using XPK to compress the file that has to be added to the archive. XarAddFileA() uses the same tags for compression as XPK plus some additional tags. In fact the XPK-tags are just passed on to XpkPack(). Since the output-tags are already set by the xpkarchive.library you must not supply these tags. The function internally calls XarNextGeneration() to prevent duplicate filenames with the same generationnumber by default.

You can suppress this behaviour by using the XAR\_AutoNextGen tag.

NOTE: a4 is preserved, therefore hooks called by this function need not use \_\_saveds. See xpk/--progress-- for further information.

TAGS

supported tags:

```

XAR_Error
XAR_InName
XAR_Archive
XAR_FileName
XAR_Generation
XAR_FileNote
XAR_TimeDay
XAR_TimeMonth
XAR_TimeYear
XAR_TimeHour
XAR_TimeMin
XAR_TimeSec
XAR_DateStamp
XAR_CurrentTime
XAR_CRC
XAR_Protection
XAR_InLen

```

XAR\_AutoNextGen

all XPK-Tags, that are supported by XpkPack(),  
without output Tags

INPUT

tags - Pointer to an array of struct TagItem. See xpkarchive/tags

RESULT

arclock - the ArcLock of the added file or NULL, if there was an  
error

SEE ALSO

XarOpenArchive(),  
XarExtractFileA()  
,XarNextGeneration(),  
xpkarchive/tags

### 1.3 xpkarchive.library/XarCloseArchive

NAME

XarCloseArchive

SYNOPSIS

```
err = XarCloseArchive( arc )
D0          A0
```

```
LONG XarCloseArchive(XarHandle *)
```

FUNCTION

Closes Archive that was opened with  
XarOpenArchiveA()

.

INPUT

arc - The ArchHandle obtained from  
XarOpenArchiveA()  
RESULT  
err - Xar error code

SEE ALSO

XarOpenArchiveA()

### 1.4 xpkarchive.library/XarCopyFileA

NAME

XarCopyFileA

SYNOPSIS

```
arclock=XarCopyFileA( tags )
```

```
D0
A0

XarLock *XarCopyFileA(struct TagItem *)
```

## FUNCTION

With XarCopyFileA() you can copy files between to archives without decompressing and newly compressing them. For this the two archives must be of the same type. You can determine the type of an archive using

```
XarGetType()
```

. The type of standard XpkArchive-archives is "XPKArchive". The function internally calls XarNextGeneration() to prevent duplicate filenames by default. You can suppress this by using the tag XAR\_AutoNextGen.

## TAGS

supported tags:

```
XAR_Error
XAR_ArcLock
XAR_Archive
XAR_DestArchive
XAR_AutoNextGen
XAR_NewName
XAR_FileNote
XAR_NewGeneration
XAR_ArcLock
XAR_Archive
XAR_FileIndex
XAR_FileName
XAR_Generation
```

## INPUT

```
tags    - Pointer to an array of struct TagItem. See xpkarchive/tags
```

## RESULT

```
arclock - The arclock of the file in the destination archive or
          NULL if an error happened.
```

## SEE ALSO

```
XarOpenArchiveA()
,XarNextGeneration(), xpkarchive/tags
```

## 1.5 xpkarchive.library/XarExamine

## NAME

```
XarExamine()
```

## SYNOPSIS

```
err = XarExamine( xarlock , xfib )
D0              A0      A1
```

```
LONG XarExamine( XarLock *, struct XpkFib * )
```

---

## FUNCTION

XarExamine() is the xpkarchive.library version of XpkExamine(). With this function it is possible to get additional informations about files in the archive. You cannot use this function on archives which are no xpk-archives.

## INPUT

```

xarlock - xarlock obtained by
          XarGetLock()
          ,
          XarNextLock()
          ,
          XarPrevLock()
          or
          XarFindFileA()
          xfib      - pointer to an xpk-fib structure

```

## RESULT

```
err - Error code
```

## SEE ALSO

```

XarGetLock()
,
XarNextLock()
,
XarPrevLock()
,
XarFindFileA()

```

## 1.6 xpkarchive.library/XarExtractFileA

## NAME

```
XarExtractFileA
```

## SYNOPSIS

```
err = XarExtractFileA( tags )
D0                                     A0
```

```
LONG XarExtractFileA(struct TagItem *)
```

## FUNCTION

This function is used to extract files from an archive and decompress them. Just like XarAddFile() it uses the XPK tags for uncompression plus some additional tags. These tags are passed on to XpkUnPack(). Therefore you must not supply XPK\_In - tags.

NOTE: a4 is preserved, therefore hooks called by this function need not use \_\_saveds. See xpk/--progress-- for further information.

## TAGS

```

supported tags:
  XAR_Error
  XAR_ArcLock

```



```
XAR_Archive
XAR_FileIndex
XAR_FileName
XAR_Generation
```

all XPK-Tags, that are supported by XpkUnpack(),  
without input Tags

#### INPUT

tags - Pointer to an array of struct TagItem. See xpkarchive/tags

#### RESULT

err - Error code

#### SEE ALSO

XarAddFile(), xpkarchive/tags

## 1.7 xpkarchive.library/XarFindFileA

#### NAME

XarFindFileA

#### SYNOPSIS

```
arclock = XarFindFileA( tags )
D0          A0
```

```
XarLock *XarFindFileA(struct TagItem *)
```

#### FUNCTION

Find the arclock to a given filename and generation number or  
fileindex number.

#### TAGS

```
supported tags:
  XAR_ArcLock
  XAR_Archive
  XAR_FileIndex
  XAR_FileName
  XAR_Generation
```

#### INPUT

tags - Pointer to an array of struct TagItem. See xpkarchive/tags

#### RESULT

arclock - the arclock of the file or NULL

#### SEE ALSO

```
XarNextLock()
,
XarPrevLock()
,xpkarchive/tags
```

## 1.8 xpkarchive.library/XarGetFileData

NAME  
XarGetFileData

### SYNOPSIS

```
fdata = XarGetFileData( arclock )
D0          A0

struct FileData *XarGetFileData(XarLock *)
```

### FUNCTION

With this function you can get information about the file stored in the archive. NOTE: the size of the compressed file is NOT stored in fdata. You can get the size of the compressed file using the function

```
XarGetFileSize()
.
```

### INPUT

arclock - the arclock of an file in the archive. You can find the arclock of an file using  
XarFindFileA()  
or XarGetLock()/

```
XarNextLock()/XarPrevLock()
RESULT
```

fdata - a pointer to struct FileData:

```
struct FileData {
    UBYTE CHKSum;    // Checksum cannot be modified
    UBYTE Method;   // Compression Method: XPK_COMPRESSION
    UBYTE Version;  // Version of then xpkmaster.library
    UBYTE Generation; // Generation
    USHORT SysID;   // System ID for Amiga: "AM"
    ULONG Filesize; // Size of the uncompressed File
    struct Time Time; // Date of the File
    USHORT CRC;     // user-definable CRC-Checksum
    ULONG Protection; // FileProtection Flags
};
```

### SEE ALSO

```
XarFindFileA()
,
XarGetLock()
,
XarNextLock()
,
XarPrevLock()
, xpkarchive.h,
XarGetFileSize()
```

## 1.9 xpkarchive.library/XarGetFileIndex

```

                                NAME
XarGetFileIndex

SYNOPSIS
index = XarGetFileIndex( arclock )
      D0                                A0

      ULONG XarGetFileIndex(XarLock *)

FUNCTION
      Evaluates the position of the file in the archive.

INPUT
      arclock - the arclock of an file in the archive. You can find the
                arclock of an file using
                XarFindFileA()
                or XarGetLock()/
                XarNextLock()/XarPrevLock()
      RESULT
      index - the index of the file in the archive

SEE ALSO
      XarFindFileA()
      ,
      XarGetLock()
      ,
      XarNextLock()
      ,
      XarPrevLock()

```

## 1.10 xpkarchive.library/XarGetFileName

```

                                NAME
XarGetFileName

SYNOPSIS
filename = XarGetFileName( arclock )
      D0                                A0

      UBYTE *XarGetFileName(XarLock *)

FUNCTION
      Determines the filename of the file, that is identified by arclock.

INPUT
      arclock - the arclock of an file in the archive. You can find the
                arclock of an file using XarGetLock()/XarNextLock()/
                XarPrevLock()

```

---

RESULT  
filename - the filename  
SEE ALSO

XarGetLock()  
,  
XarNextLock()  
,  
XarPrevLock()

## 1.11 xpkarchive.library/XarGetFileNote

NAME  
XarGetFileNote

SYNOPSIS  
filename = XarGetFileNote( arclock )  
D0 A0

UBYTE \*XarGetFileNote(XarLock \*)

FUNCTION  
Determines the filename of the file, that is identified by arclock.

INPUT  
arclock - the arclock of an file in the archive. You can find the  
arclock of an file using XarGetLock()/XarNextLock()/

XarPrevLock()  
RESULT  
filename - the filename of the file  
SEE ALSO

XarGetLock()  
,  
XarNextLock()  
,  
XarPrevLock()

## 1.12 xpkarchive.library/XarGetFileSize

NAME  
XarGetFileSize  
SYNOPSIS  
filesize = XarGetFileSize( arclock )  
D0 A0

ULONG XarGetFileSize(XarLock \*)

FUNCTION

---

Determines the compressed size of the file, that is identified by arclock.

#### INPUT

arclock - the arclock of an file in the archive. You can find the arclock of an file using XarGetLock()/XarNextLock()/

XarPrevLock()

RESULT

filesize - the compressed size of the file

#### SEE ALSO

XarGetLock()

,

XarNextLock()

,

XarPrevLock()

## 1.13 xpkarchive.library/XarGetLock

NAME

XarGetLock

#### SYNOPSIS

arclock = XarGetLock( arc )

D0

A0

XarLock \* XarGetLock( XarHandle \*)

#### FUNCTION

This function gets the first entry of the contents list of the archive. The arclock returned by this function can be used to get further information of the file or to get the next or previous entry in the file using

XarNextLock()

or

XarPrevLock()

.

#### INPUT

arc - The archandle returned by

XarOpenArchiveA()

RESULT

arclock - the identifier of a file in the archive, or NULL if the archive is empty.

#### SEE ALSO

XarNextLock()

,

XarPrevLock()

,

XarOpenArchiveA()

## 1.14 xpkarchive.library/XarGetType

NAME

XarGetType

SYNOPSIS

```
type = XarGetType( arc )
D0                                A0
```

```
UBYTE * XarGetType( XarHandle * )
```

FUNCTION

The type of an archive gives a hint, with which packer it was created. The archive that was created with xpkarchive.library have the type "XPKArchive". With the function XarGetType() it is possible to determine this type of the archive.

NOTE: Archives of an other type can NOT be modified. No files can be extraced from these archives. This behaviour may change in the future when there will be xpksublibs, that can handle other types of archives, e.g. archives that were created with lha or zoo.

INPUT

arc - The archandle returned by  
XarOpenArchiveA()  
RESULT

type - The type of the archive as a null-terminated string.

SEE ALSO

XarOpenArchiveA()

## 1.15 xpkarchive.library/XarIsXpkArchive

NAME

XarIsXpkArchive

SYNOPSIS

```
bool = XarIsXpkArchive( arc )
D0                                A0
```

```
UBYTE XarGetType( XarHandle * )
```

FUNCTION

Test if an archive is an xpk-archive.

NOTE: Archives that are no xpkarchives can NOT be modified. No files can be extraced from these archives. This behaviour may change in the future when there will be xpksublibs, that can handle other types of archives, e.g. archives that were created with lha or zoo.

INPUT  
arc - The xarhandle returned by  
XarOpenArchiveA()  
RESULT  
bool - 0 if this is no xpk-archive

SEE ALSO

XarOpenArchiveA()

## 1.16 xpkarchive.library/XarModifyFileDataA

NAME

XarModifyFileDataA()

SYNOPSIS

```
err = XarModifyFileDataA( tags )  
D0                                A0
```

```
LONG XarModifyFileDataA(struct TagItem *)
```

FUNCTION

This function is used to modify entries in the FileData structure, that is bound to every.

NOTE: it is not possible to change some fields in the FileData structure:

FileSize,CHKSum

TAGS

supported tags:

- XAR\_Error
- XAR\_ArcLock
- XAR\_Archive
- XAR\_FileIndex
- XAR\_FileName
- XAR\_Generation
- XAR\_TimeDay
- XAR\_TimeMonth
- XAR\_TimeYear
- XAR\_TimeHour
- XAR\_TimeMin
- XAR\_TimeSec
- XAR\_DateStamp
- XAR\_CRC
- XAR\_Protection
- XAR\_NewGeneration

INPUT

tags - Pointer to an array of struct TagItem. See xpkarchive/tags

RESULT

err - Error code

---

SEE ALSO

```
XarGetFileData()
, xpkarchive/tags
```

## 1.17 xpkarchive.library/XarNextGenerationA

NAME

```
XarNextGenerationA()
```

SYNOPSIS

```
err = XarNextGenerationA( tags )
D0                                A0
```

```
LONG XarNextGenerationA(struct TagItem *)
```

FUNCTION

Unlike most other archivers xpkarchive supports filegenerations. This feature allows to have more than one file with the same name in one archive. In most cases this will be different versions of the same file. It is thought that the newest version has the lowest versionnumber. This function is used to increase the generationnumber of this file and all files with the same name and a higher generationnumber. This function is automatically called by the functions XarAddFile(), XarRenameFileA() and XarCopyFileA() to prevent the duplicate occurrence of files with the same filename and generationnumber. If you want to set the generationnumber of a single file use the function XarModifyFileData().

TAGS

```
supported tags:
  XAR_Error
  XAR_ArcLock
  XAR_Archive
  XAR_FileIndex
  XAR_FileName
  XAR_Generation
```

INPUT

```
tags    - Pointer to an array of struct TagItem. See xpkarchive/tags
```

RESULT

```
err - Error code
```

SEE ALSO

```
XarAddFile(), XarCopyFile(),
  XarRenameFileA()
, XarModifyFileData(),
--generations--
```



## 1.18 xpkarchive.library/XarNextLock

NAME  
XarNextLock()

### SYNOPSIS

```
arclock = XarNextLock( arclock )
D0                      A0
```

```
XarLock * XarNextLock ( XarLock * )
```

### FUNCTION

Get the next entry of the contents list.

### INPUT

arclock - The arclock of a file

### RESULT

arclock - The arclock of the next file in the archive or NULL, if there is no file left.

### SEE ALSO

```
XarPrevLock()
,
XarGetLock()
```

## 1.19 xpkarchive.library/XarOpenArchiveA

NAME  
XarOpenArchiveA

### SYNOPSIS

```
arc = XarOpenArchiveA( tags )
D0                      A0
```

```
XarHandle *XarOpenArchiveA(struct TagItem *)
```

### FUNCTION

Opens an archive from disk and scans through the archive for building the table of contents of the archive. You can access this list using

```
XarGetLock()
to get the first entry and XarNextLock()/
```

```
XarPrevLock()
```

to get the next/previous entry. XarOpenArchiveA() will even recognize every CDAF-Archive e.g. archives that were generated with shrink. You cannot add to or extract files from these archives,

but you can examine the contents of these archives with  
 XarGetLock()

.

#### INPUT

tags - XAR\_ArchiveName: The filename of the archive

#### XAR\_ArchiveMode:

There are three modes to open an archive:

##### XAR\_ModeOldArchive:

Only existing archives will be opened. If it does not exist XarOpenArchiveA will fail. This is the default mode.

##### XAR\_ModeNewArchive:

Existing files will be deleted and a new, empty archive will be created.

##### XAR\_ModeAppend:

Creates an archive if none exists and opens the archive if one exists.

- See also xpkarchive/tags

#### RESULT

XarHandle \* if successful NULL else

#### SEE ALSO

XarCloseArchive()

,

XarGetLock()

## 1.20 xpkarchive.library/XarPrevLock

#### NAME

XarPrevLock()

#### SYNOPSIS

```
arclock = XarPrevLock( arclock )
           D0             A0
```

```
XarLock * XarPrevLock ( XarLock * )
```

#### FUNCTION

Get the previous entry of the contents list.

#### INPUT

arclock - The arclock of a file

#### RESULT

arclock - The arclock of the previous file in the archive or NULL, if there is no file left.

#### SEE ALSO

```
XarNextLock()  
,  
XarGetLock()
```

## 1.21 xpkarchive.library/XarRenameFileA

```
NAME  
XarRenameFileA()
```

### SYNOPSIS

```
err = XarRenameFileA( tags )  
D0          A0  
  
LONG XarRenameFileA(struct TagItem *)
```

### FUNCTION

In archives of the type "XPKArchive" it is possible to rename files. Using this function you can set a new filename and generationnumber. Internally it calls XarNextGeneration() to prevent duplicate filenames with the same generationnumber by default. You can suppress this by using the tag XAR\_AutoNextGen.

### TAGS

```
supported tags:  
XAR_ArcLock  
XAR_Archive  
XAR_FileIndex  
XAR_FileName  
XAR_Generation  
XAR_Error  
XAR_NewName  
XAR_NewGeneration  
XAR_AutoNextGen
```

### INPUT

```
tags - Pointer to an array of struct TagItem. See xpkarchive/tags
```

### RESULT

```
err - Error code
```

### SEE ALSO

```
XarNextGeneration(), XarModifyFileData(),  
XarGetType()
```

## 1.22 xpkarchive.library/XarSetFileNoteA

```
NAME  
XarSetFileNoteA()
```

### SYNOPSIS

---

```
err = XarSetFileNoteA( tags )
D0                               A0
```

```
LONG XarSetFileNoteA(struct TagItem *)
```

#### FUNCTION

In archives of the type "XPKArchive" it is possible to set filenotes to files which are already in the archive. For removing filenotes supply a pointer to a string of length 0 or a NULL-pointer.

#### TAGS

supported tags:

```
XAR_Error
XAR_ArcLock
XAR_Archive
XAR_FileIndex
XAR_FileName
XAR_Generation
XAR_FileNote
```

#### INPUT

tags - Pointer to an array of struct TagItem. See xpkarchive/tags

#### RESULT

err - Error code

#### SEE ALSO

XarGetType()

## 1.23 xpkarchive.library/--tags--

Tags for XarOpenArchiveA()

```
XAR_ArchiveName (UBYTE *)
Name of the file of the archive.
```

```
XAR_ArchiveMode (ULONG)
```

There are three modes to open an archive:

```
XAR_ModeOldArchive:
```

Only existing archives will be opened. If it does not exist

```
XarOpenArchiveA()
```

will fail. This is the default mode.

```
XAR_ModeNewArchive:
```

Existing files will be deleted and a new, empty archive will be created.

```
XAR_ModeAppen:
```

Creates an archive if none exists and opens the archive if one exists.

Tags for all functions requiring tags except XarOpenArchiveA():

XAR\_ArcLock (XarLock \*)  
Arclock of the file.

XAR\_Archive (XarHandle \*)  
Archandle of the archive.

XAR\_FileIndex (ULONG)  
Index of the file in the archive. This tag is only valid if there is also XAR\_Archive given.

XAR\_FileName (UBYTE \*)  
Filename of the file in the archive. This tag is only valid if there is also XAR\_Archive given. The default generation is 0 unless you change the generation with the tag:

XAR\_Generation (UBYTE)  
Generationnumber of the file in the archive.

For identifying file in the archive use one of the following:

- XAR\_ArcLock
- or - XAR\_Archive + XAR\_FileIndex
- or - XAR\_Archive + XAR\_FileName [ + XAR\_Generation (default: 0)]

NOTE: the combination of XAR\_Archive and XAR\_FileName is ambiguous when there is more than one file with the same filename and generationnumber in the archive. In this case it identifies the first file in the archive that matches the filename and generationnumber.

Additional Tags for XarAddFileA():

XAR\_Archive, XAR\_FileName, XAR\_Generation,

XAR\_FileNote

XAR\_CRC

XAR\_Protection

XAR\_AutoNextGen

XAR\_TimeDay (UBYTE)

XAR\_TimeMonth (UBYTE)

XAR\_TimeYear (UBYTE)

XAR\_TimeHour (UBYTE)

XAR\_TimeMin (UBYTE)

XAR\_TimeSec (UBYTE)

XAR\_DateStamp (struct DateStamp \*)

These tags are used to specify the time for XarAddFile() as day, month, year, hour, minute and second or using a datestamp. The default behaviour is to use the date and time of the file, or in the case of XPK\_InBuf or XPK\_InHook the actual time.

---

This time can be altered using the above tags.

These tags are also used to modify the FileData using XarModifyFileData().

XAR\_CurrentTime (BOOLEAN)

Use the current time instead of the date and time of the file.

Additional Tags for XarRenameFileA()

XAR\_NewName (UBYTE \*)

The new filename for the file.

XAR\_NewGeneration (UBYTE)

The new generationnumber for the file.

XAR\_AutoNextGen (ULONG)

Call XarNextGeneration if 1, which is the default value.

If you supply 0 then XarNextGeneration is not called. This may result in duplicate files with the same generationnumber in the archive.

IMPORTANT: You must not use other values than 0 or 1, since it might be that there are other values with other functionality in the future.

Additional tags for XarSetFileNoteA()

XAR\_FileNote (UBYTE \*)

Filenote for the file. In the archive the filenotes have no limited size, but remember that it is not possible to extract the complete filenote, if it is longer than 80 bytes.

Additional tags for XarCopyFileA()

XAR\_DestArchive (XarHandle \*)

ArchHandle of the destination archive.

XAR\_AutoNextGen (ULONG)

(see above)

XAR\_NewName (UBYTE \*)

New name for the file in the dest. archive,  
default: use old name

XAR\_NewGeneration (UBYTE)

New generationnumber for the file in the dest. archive,  
default: use old generationnumber

XAR\_FileNote (UBYTE \*)

---

New filenote, default: use old filenote

Additional tags for

XarModifyFileDataA()  
and XarAddFileA()

XAR\_CRC (USHORT)

CRC checksums of the file in the archive. CRC checksums are not supported by the xpkarchive.library yet. With this option you can add your own CRC Checksum to the file. The default value used by xpkarchive.library is 0x00.

XAR\_Protection (ULONG)

Protection of the file.

XAR\_AutoNextGen (ULONG)

(see above)

Additional tags for XarOpenArchiveA()

XAR\_ArchiveMode (ULONG)

Mode of opening the archive. There are three modes to open the archive:

XAR\_ModeOldArchive:

Only existing archives will be opened. If it does not exist

XarOpenArchiveA()

will fail. This is the default mode.

XAR\_ModeNewArchive:

Existing files will be deleted and a new, empty archive will be created.

XAR\_ModeAppend:

Creates an archive if none exists and opens the archive if one exists.

Global tags for all functions

XAR\_Error (LONG \*)

Pointer to an error variable. The error code is stored in this variable on failure. Error codes <0 are error codes of the xpkmaster.library (see xpk.h), error codes >0 are from the xpkarchive.library.

IMPORTANT NOTE:

Only error-codes >=XARERROR\_LEVEL are real errors. The other codes are just informational or warnings. See also --Error--

Input / Output tags:

The normal xpk-tags from xpkmaster.library are used as input/output

---

```
tags. Be aware that you must not supply output tags to
XarAddFileA()
    and no input tags to
XarExtractFileA()
.
```

## 1.24 xpkarchive.library/--error--

Some notes about the error codes:

The xpkarchive.library has three levels of error codes:

- Informational codes: These are no errors at all.

```
XARINF_NEW_ARCHIVE:      A new archive has been created.
XARINF_NO_READWRITE:    The archivefile could not be opened as
                        MODE_READWRITE
XARINF_ARCHIVE_CORRUPT: There are more or less harmless errors
                        in the archive.
```

- Warning codes (code>=XARWARNING\_LEVEL): They give you a warning that there is something wrong with the archive.

```
XARWARN_ARCHIVE_CORRUPT: There is a serious error in the
                        archive. Therefore it is not possible to add any files to
                        the archive any more. If you try it you will get the error
                        code XARERR_ARCHIVE_CORRUPT. You can still try to extract
                        files from the archive.
```

For the other codes read xpkarchive.h

- Error codes (code>=XARERROR\_LEVEL): Real error codes

## 1.25 xpkarchive.library/--generations--

Unlike most formats of other archivers the file-format that is used by the xpkarchive.library supports having different versions of one file with the same filename in one archive. Therefore this file-format supports generations of files, which identify different versions of the file. The lower the generationnumber the newer is the file. Since it is always possible to modify the generationnumber by the programmer it may occur that there are files with the same filename and generation. It is the job of the programmer to prevent duplicate generationnumbers of one file except for the special generationnumber XAR\_FileDeleted. Deleted files are not removed from the archive. The only way to get rid of these files is to generate a new archive and copy the files, which are not deleted or which have a generationnumber lower than a certain level to this archive and delete the old archive.

## 1.26 xpkarchive.library/--BUGS--



The xpkarchive.library does not support the tags TAG\_MORE and TAG\_SKIP yet.

---