

XAD Plugin 1.0

The Multi-Format Unarchiver For Hollywood

Andreas Falkenhahn

Inhaltsverzeichnis

1	Allgemeine Informationen	1
1.1	Einführung	1
1.2	Lizenzvereinbarung	1
1.3	Voraussetzung	2
1.4	Installation	3
2	Über xad.hwp	5
2.1	Danksagungen	5
2.2	Häufig gestellte Fragen	5
2.3	Bekannte Probleme	5
2.4	Zukunft	5
2.5	Geschichte	6
3	Verwendung	7
3.1	Aktivieren des Plugins	7
3.2	Archive als Verzeichnisse	7
3.3	Dateien entpacken	8
3.4	Archiv Grundlagen	9
3.5	Dateien verknüpfen	9
4	Befehlsreferenz	11
4.1	xad.CloseArchive	11
4.2	xad.ExtractFile	11
4.3	xad.GetFileAtIndex	12
4.4	xad.GetFileAttributes	12
4.5	xad.GetObjectType	13
4.6	xad.LocateFile	14
4.7	xad.OpenArchive	15
4.8	xad.SetDefaultPassword	15
Anhang A	Lizenzen	17
A.1	xadmaster.library-Lizenz	17
A.2	LGPL-Lizenz	17
Index		21

1 Allgemeine Informationen

1.1 Einführung

Dieses Plugin ermöglicht es Hollywood-Skripten, Archive mit dem XAD-System zu öffnen. Das XAD-System ist ein Toolkit, das für die Handhabung verschiedener Datei- und Disk-Archivierungs-Formate entwickelt wurde. Es besteht aus einer Master-Bibliothek und einer Anzahl von XAD-Clients für die Handhabung verschiedener Datei- und Disk-Archivierungs-Formate. Standardmäßig unterstützt das XAD-System bereits viele gängige Archivierungs-Formate wie ZIP, LhA, LZX, CAB, bzip2, TAR und RAR, was es zu einem leistungsstarken Werkzeug für Leute macht, die viel mit komprimierten Daten zu tun haben.

Das XAD-System hat seinen Ursprung auf der Amiga-Plattform, für die es in den 1990er Jahren ursprünglich von Dirk Stoecker entwickelt wurde. Aus diesem Grund verwenden alle Versionen von `xad.hwp` für AmigaOS und kompatible weiterhin die vom Hostsystem bereitgestellte `xadmaster.library`. Sowohl auf AmigaOS 4 als auch auf MorphOS ist die `xadmaster.library` installiert. Für AmigaOS 3 und AROS ist `xadmaster.library` als Software von Drittanbietern verfügbar. Auf allen anderen Plattformen verwendet `xad.hwp` eine Open-Source-Version des XAD-Systems.

Das XAD-Plugin verwendet die neue Plugin-Schnittstelle von Hollywood 6.0 für Datei- und Verzeichnisadapter, mit denen Sie die vom XAD-System unterstützten Dateiarhive so durchlaufen können, als wären sie normale Verzeichnisse. Auf Dateien in Archiven, die vom XAD-System unterstützt werden, kann ebenfalls zugegriffen werden, als ob das Archiv ein normales Verzeichnis wäre. Es ist nicht notwendig, eine in einem Archiv gespeicherte Datei in eine temporäre Datei zu entpacken, bevor sie geöffnet werden kann. Die Adapter-Plugin-Schnittstelle von Hollywood 6.0 ermöglicht das direkte Lesen vom Archiv in den jeweiligen Datei-Handler.

Darüber hinaus bietet `xad.hwp` eine Reihe von Funktionen für den Umgang mit Archiven, die vom XAD-System unterstützt werden. Dies ist für Personen, die eine fein abgestimmte Kontrolle über das XAD-System benötigen. Mit diesen Funktionen können Sie bequem alle Dateien und Verzeichnisse in einem Archiv durchlaufen, Dateiattribute wie Datum, Schutzbits, Kompressionsdetails usw. lesen und entpacken Dateien in eine externe oder virtuelle Datei.

1.2 Lizenzvereinbarung

`xad.hwp` ist © Copyright 2014-2018 bei Andreas Falkenhahn (im folgenden "der Autor" genannt). Alle Rechte vorbehalten.

Das Programm wird zur Verfügung gestellt "wie es ist" und der Autor kann für keinerlei Schäden, welcher Natur sie auch immer sein mögen, verantwortlich gemacht werden. Sie benutzen dieses Programm völlig auf eigene Gefahr und eigenes Risiko. Der Autor gibt keinerlei Garantien in Verbindung mit der Benutzung dieses Programmes, nicht einmal die Garantie der Funktionstüchtigkeit.

`xad.hwp` für AmigaOS 3, AmigaOS 4, AROS und MorphOS kann frei weitergegeben werden solange die folgenden drei Bedingungen erfüllt sind:

1. Es dürfen keine Änderungen am Programm vorgenommen werden.

2. Das Programm darf nicht verkauft werden.
3. Wenn Sie das Programm auf einer Coverdisk veröffentlichen möchten, müssen Sie erst um Erlaubnis fragen.

xad.hwp für alle anderen Plattformen ist freie Software und darf weitergegeben und/oder verändert werden unter den Bedingungen der GNU Lesser General Public License wie sie von der Free Software Foundation publiziert wurde; entweder Version 2.1 der Lizenz oder (nach Ihrer Wahl) eine beliebige neuere Version. xad.hwp wurde entwickelt in der Hoffnung, dass es nützlich sein wird, aber ohne jegliche Garantie; nicht einmal die Garantie der Gebrauchstauglichkeit oder Eignung für einen bestimmten Zweck. Siehe [Abschnitt A.2 \[LGPL-Lizenz\]](#), [Seite 17](#), für Details.

Diese Software benutzt die xadmaster.library von Dirk Stoecker. Siehe [Abschnitt A.1 \[xadmaster.library-Lizenz\]](#), [Seite 17](#), für Details.

xad.hwp für alle Nicht-Amiga-Plattformen benutzt den XAD RAR-Client, welcher von Chris Young <chris@unsatisfactorysoftware.co.uk> und Stephan Matzke <stephan.matzke.adev@gmx.de> geschrieben wurde. Dieser Client basiert auf einer modifizierten Version des RAR_Extractor von Shay Green, der wiederum auf dem UnRAR-Quelltext von Alexander L. Roshal basiert.

Alle Warenzeichen sind Eigentum ihrer jeweiligen Firmen.

FÜR DIESES PROGRAMM GIBT ES KEINE GARANTIE, SOWEIT ES DIE ANZUWENDENDEN GESETZE ZULASSEN. SOFERN ANDERSWO NICHTS GEGENTEILIGES GESCHRIEBEN STEHT STELLEN DER AUTOR UND/ODER DRITTE DAS PROGRAMM "SO WIE ES IST" ZUR VERFÜGUNG, OHNE IRGEND-EINE GARANTIE, WEDER DIREKT NOCH INDIREKT. DIES BEINHÄLTET, IST ABER NICHT DARAUf BESCHRÄNKT, VERKÄUFLICHKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN VERWENDUNGSZWECK. DAS VOLLSTÄNDIGE RISIKO DER QUALITÄT UND AUSFÜHRBARKEIT DES PROGRAMMS LIEGT BEIM ANWENDER. SOLLTE SICH DAS PROGRAMM ALS DEFECT HERAUSSTELLEN, LIEGEN ALLE KOSTEN FÜR SERVICE, INSTANDSETZUNG ODER NACHBESSERUNG BEIM ANWENDER.

KEIN COPYRIGHT-INHABER ODER DRITTER, DER DAS PROGRAMM WIE OBEN ERLAUBT WEITERVERKAUFT, KANN FÜR SCHÄDEN IRGENDWELCHER ART HAFTBAR GEMACHT WERDEN (DIES BEINHÄLTET, IST ABER NICHT BESCHRÄNKT AUF, DATENVERLUST INFOLGE UNFÄHIGKEIT DES PROGRAMMS, MIT ANDEREN PROGRAMMEN ZUSAMMENZUARBEITEN), SELBST WENN EIN SOLCHER INHABER ODER DRITTER AUF DIE MÖGLICHKEIT EINES SOLCHEN SCHADENS HINGEWIESEN WURDE, AUSSER ES BESTEHT EINE SCHRIFTLICHE EINWILLIGUNG ODER WIRD VOM GESETZ VERLANGT.

1.3 Voraussetzung

- minimum: Hollywood 6.0 oder besser
- Unicode: Für die Unterstützung von Unicode wird Hollywood 7.0 oder höher empfohlen

1.4 Installation

Die Installation von `xad.hwp` ist unkompliziert und einfach: Kopieren Sie einfach die Datei `xad.hwp` für die Plattform Ihrer Wahl in Hollywoods Plugin-Verzeichnis. Auf allen Systemen außer auf AmigaOS und kompatiblen Computern müssen Plugins in einem Verzeichnis mit dem Namen `Plugins` gespeichert werden, das sich im selben Verzeichnis wie das Hauptprogramm von Hollywood befindet. Auf AmigaOS und kompatiblen Systemen müssen Plugins stattdessen in `LIBS:Hollywood` installiert werden. Unter Mac OS X muss sich das `Plugins`-Verzeichnis im Verzeichnis `Resources` des Programmpakets befinden, d.h. im Verzeichnis `HollywoodInterpreter.app/Contents/Resources`. Beachten Sie, dass `HollywoodInterpreter.app` im `Hollywood.app`-Programmpaket selbst gespeichert ist, nämlich in `Hollywood.app/Contents/Resources`.

Unter Windows sollten Sie auch die Datei `xad.chm` in das `Docs`-Verzeichnis Ihrer Hollywood-Installation kopieren. Dann können Sie die Online-Hilfe aufrufen, indem Sie F1 drücken, wenn sich der Cursor in der Hollywood IDE über einem `xad.hwp`-Befehl befindet.

Kopieren Sie unter Linux und Mac OS das Verzeichnis `xap`, das sich im Verzeichnis `Docs` des Distributionsarchivs von `xad.hwp` befindet, in das Verzeichnis `Docs` Ihrer Hollywood-Installation. Beachten Sie, dass sich das `Docs`-Verzeichnis unter Mac OS im `Hollywood.app`-Programmpaket befindet, d.h. in `Hollywood.app/Contents/Resources/Docs`.

2 Über xad.hwp

2.1 Danksagungen

Xad.hwp wurde von Andreas Falkenhahn geschrieben. Die Arbeit an diesem Projekt wurde Anfang 2014 als Demonstration der leistungsfähigen neuen Datei- und Verzeichnisadapter-Plugin-Schnittstelle von Hollywood 6.0 gestartet, die es Plugins ermöglichen, sich in Hollywoods Datei- und Verzeichnishandler einzuklinken. Xad.hwp nutzt diese Funktion, indem Hollywood denkt, dass die vom XAD-System unterstützten Archive nur Verzeichnisse sind, so dass es möglich ist, sie mithilfe der normalen Verzeichnisfunktionen von Hollywood zu durchlaufen oder Dateien in Archiven zu öffnen, ohne sie zuerst zu entpacken.

Ein besonderer Dank geht an Helmut Haake und Dominic Widmer für die Übersetzung des Handbuchs ins Deutsche. Fehler oder Verbesserungsvorschläge bzgl. des deutschen Hollywood-Handbuchs bitte an das Übersetzungsteam richten, welches unter handbuch@gmx.ch erreicht werden kann.

Wenn Sie mich kontaktieren möchten, können Sie entweder eine E-Mail an andreas@airsoftsoftwair.de senden oder das Kontaktformular auf <http://www.hollywood-mal.com> verwenden.

2.2 Häufig gestellte Fragen

In diesem Abschnitt werden einige häufig gestellte Fragen behandelt. Bitte lesen Sie sie zuerst, bevor Sie in der Mailingliste oder im Forum nachfragen, da Ihr Problem hier möglicherweise behandelt wurde.

F: Wo kann ich um Hilfe bitten?

A: Es gibt ein lebhaftes Forum auf <http://forums.hollywood-mal.com> und wir haben auch eine Mailingliste, auf die Sie unter airsoft_hollywood@yahoogroups.com zugreifen können. Besuchen Sie <http://www.hollywood-mal.com> für Informationen darüber, wie Sie der Mailingliste beitreten können.

F: Ich habe einen Fehler gefunden.

A: Bitte poste darüber in den entsprechenden Bereichen des Forums oder der Mailingliste.

2.3 Bekannte Probleme

Hier ist eine Liste von Dingen, die xad.hwp noch nicht unterstützt oder die in irgendeiner Weise verwirrend sein könnten:

- tbd

2.4 Zukunft

Hier sind einige Dinge, die auf meiner Liste stehen:

- tbd

Zögern Sie nicht, mich zu kontaktieren, wenn in xad.hwp eine bestimmte Funktion fehlt, die für Ihr Projekt wichtig ist.

2.5 Geschichte

In der auf englisch verfasste Datei `history.txt` finden Sie ein vollständiges Änderungsprotokoll von `xad.hwp`.

3 Verwendung

3.1 Aktivieren des Plugins

Es gibt zwei Möglichkeiten, dieses Plugin zu verwenden: Sie können das Plugin global aktivieren, indem Sie beim Befehl `@REQUIRE` den Tag `InstallAdapter` auf `True` setzen. Geben Sie dazu einfach den erwähnten Präprozessor-Befehl im Skript ein:

```
@REQUIRE "xad", {InstallAdapter = True}
```

Wenn Sie das Plugin auf diese Weise aktivieren, wird es global verfügbar und alle folgenden Befehle, die sich mit Dateien befassen, werden das Öffnen von Dateien aus XAD-Archivquellen unterstützen. Zum Beispiel könnten Sie dann ein Bild aus einem Archiv laden:

```
LoadBrush(1, "test.rar/testpicture.jpg")
```

Wenn Sie nur sehr wenige Dateien aus XAD-Archivquellen öffnen müssen, können Sie das Plugin auch nicht global aktivieren, indem Sie den Tag `InstallAdapter` beim Befehl `@REQUIRE` weglassen und einfach den Tag `Adapter` verwenden, welcher von den meisten Hollywood-Befehlen angeboten wird, um die Datei mit dem Plugin `xad.hwp` zu öffnen. Hier ist ein Beispiel:

```
LoadBrush(1, "test.rar/testpicture.jpg", {Adapter = "xad"})
```

Mit dem Tag `Adapter` wird `LoadBrush()` angewiesen, die erwähnte Datei mit dem angegebenen Adapter zu öffnen, der in unserem Fall `"xad"` ist. Mit dem Tag `Adapter` können Sie also dieses Plugin verwenden, ohne zuvor einen globalen Dateiadapter installiert zu haben. Dasselbe gilt für Hollywood-Befehle, die mit Verzeichnissen arbeiten. Sobald das XAD-Plugin aktiviert wurde, ist Folgendes möglich:

```
OpenDirectory(1, "test.rar")
```

Sie könnten dann alle Dateien und Verzeichnisse in `test.rar` durchlaufen. Wenn Sie keinen globalen Adapter für `xad.hwp` aktiviert haben, verwenden Sie einfach den `Adapter` wie oben, z.B.

```
OpenDirectory(1, "test.rar", {Adapter = "xad"})
```

Im nächsten Kapitel erfahren Sie mehr darüber, wie die vom XAD-System unterstützten Archive als Verzeichnisse behandelt werden.

3.2 Archive als Verzeichnisse

Wenn der Tag `InstallAdapter` auf `True` gesetzt wird, greift das XAD-Plugin in den Verzeichnishandler ein, damit Hollywood glaubt, dass die vom XAD-System unterstützten Archive normale Verzeichnisse sind. Dies ermöglicht Ihnen alle Dateien und Verzeichnisse innerhalb eines Archivs mit normalen Befehlen aus Hollywoods DOS-Bibliothek zu durchlaufen.

Zum Beispiel wenn das Archiv `test.rar` heisst, können Sie fürs Durchlaufen den folgenden Code verwenden:

```
OpenDirectory(1, "test.rar")
Local e = NextDirectoryEntry(1)
While e <> Nil
    DebugPrint(e.name)
```

```

    e = NextDirectoryEntry(1)
Wend
CloseDirectory(1)

```

Wenn Sie nicht aus dem Stammverzeichnis innerhalb von `test.rar` anfangen möchten, können Sie auch bequem von einem Unterverzeichnis starten, indem Sie einfach so tun, als wäre `test.rar` ein Verzeichnis. Z.B. um auf ein Unterverzeichnis namens `files` innerhalb von `test.rar` zuzugreifen, gehen Sie einfach wie folgt vor:

```
OpenDirectory(1, "test.rar/files")
```

Schließlich ist es auch möglich, alle Dateien und Verzeichnisse innerhalb eines Archivs rekursiv zu durchlaufen. Hier ist eine Funktion, die das tut:

```

Function p_DumpArchive(d$, idt)
    Local id = OpenDirectory(1, d$)
    Local e = NextDirectoryEntry(id)
    While e <> Nil
        If e.Type = #DOSTYPE_DIRECTORY
            DebugPrint(RepeatStr(" ", idt) .. "+", e.name)
            p_DumpArchive(FullPath(d$, e.name), idt + 4)
        Else
            DebugPrint(RepeatStr(" ", idt) .. " ", e.name, e.size, e.time)
        EndIf
        e = NextDirectoryEntry(id)
    Wend
    CloseDirectory(id)
EndFunction

```

Um den Inhalt eines Archivs auszugeben, rufen Sie einfach folgende Funktion auf:

```
p_DumpArchive("test.rar", 0)
```

Es wird dann einen schönen Baum des Archivinhalts ausgeben.

3.3 Dateien entpacken

Da sich `xad.hwp` in den Dateihandler von Hollywood einklinken kann, ist das entpacken von Dateien nur eine Frage der Verwendung des Hollywood-Befehls `CopyFile()`. Um beispielsweise eine Datei namens `testpicture.jpg` aus `test.rar` zu entpacken, verwenden Sie einfach die folgende Zeile:

```
CopyFile("test.rar/testpicture.jpg", "outputdir")
```

Da `CopyFile()` auch ganze Verzeichnisse inklusive aller Unterverzeichnisse kopieren kann und `xad.hwp` sich auch in Hollywoods Verzeichnishandler einhakt, ist es sogar möglich, ein ganzes Archiv mit `CopyFile()` zu entpacken. Das geht so:

```
CopyFile("test.rar", "outputdir")
```

Dies wird jedoch ziemlich langsam sein, da `xad.hwp` das RAR-Archiv für jede einzelne zu extahierende Datei öffnet und schließt, was natürlich ein Performance-Killer ist. Aus diesem Grund bietet `xad.hwp` auch einen speziellen Befehl zum entpacken von Dateien für die Feinsteuerung von Archiven. Siehe [Abschnitt 4.2 \[xad.ExtractFile\]](#), Seite 11, für Details.

3.4 Archiv Grundlagen

Archive, die vom XAD-System unterstützt werden, sind nur eine Sammlung von Dateien, die in Indizes von 0 bis zur Anzahl der Einträge im Archiv minus 1 gespeichert werden. Es ist nicht notwendig, Verzeichnisse als einzelne Einträge zu speichern. Stattdessen können sie auch als Teil eines Dateinamens gespeichert werden. Z.B. wenn eine Datei als `a/b/c/test.txt` im Archiv gespeichert ist, werden die Verzeichnisse `a`, `b` und `c` implizit als existierend deklariert, obwohl sie nicht ihre individuellen Einträge im Archiv haben, sondern nur als existierenden Teil einer Datei. Sie müssen darauf vorbereitet sein, wenn Sie Archive manuell mit den Befehlen von diesem Plugin analysieren.

3.5 Dateien verknüpfen

Beachten Sie, dass alle in den Präprozessor-Befehlen deklarierten Dateien automatisch mit Ihrem Applet oder Ihrer ausführbaren Datei verknüpft werden, wenn sich Hollywood im Kompilierungsmodus befindet. Wenn Sie also etwas wie das Folgende tun, wird nicht nur `testpicture.jpg`, sondern das gesamte RAR-Archiv `test.rar` mit Ihrem Applet oder Ihrer ausführbaren Datei verknüpft:

```
@BRUSH 1, "test.rar/testpicture.jpg"
```

Wenn Sie das nicht möchten, können Sie den optionalen Tag `Link` auf `False` setzen. Wenn der Tag `Link` auf `False` festgelegt ist, verknüpft Hollywood die angegebene Datei nicht mit Ihrem Applet oder Ihrer ausführbaren Datei. Das bedeutet jedoch, dass Sie `test.rar` mit Ihrem Applet oder Ihrer ausführbaren Datei verteilen müssen, damit die Daten daraus geladen werden können. So deaktivieren Sie die Verknüpfung:

```
@BRUSH 1, "test.rar/testpicture.jpg", {Link = False}
```

Wenn Sie so vorgehen, wird Hollywood die Datei niemals mit Ihrem Applet oder Ihrer ausführbaren Datei verknüpfen. Stattdessen wird die angegebene Datei immer von extern geladen.

4 Befehlsreferenz

4.1 xad.CloseArchive

BEZEICHNUNG

xad.CloseArchive – schließt das XAD-Archiv

ÜBERSICHT

xad.CloseArchive(id)

BESCHREIBUNG

Dieser Befehl schließt das in `id` angegebene xad-Archiv.

EINGABEN

`id` Identifikator des zu schließenden XAD-Archivs

4.2 xad.ExtractFile

BEZEICHNUNG

xad.ExtractFile – entpackt die Datei aus dem xad-Archiv

ÜBERSICHT

xad.ExtractFile(id, idx, dst\$[, table])

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um die Datei im Index `idx` innerhalb des durch `id` angegebenen xad-Archivs in die externe Datei zu entpacken, die in `dst$` angegeben wird. Wenn `dst$` bereits existiert, wird es überschrieben. Mit dem optionalen Tabellenargument `table` können Sie weitere Optionen für die Operation angeben.

Die folgenden Tags werden derzeit im optionalen Argument `table` erkannt:

Password:

Wenn die Datei, die Sie entpacken möchten, durch ein Passwort geschützt ist, müssen Sie dieses Passwort hier angeben. Wenn Sie diesen Tag nicht angeben, wird das mit `xad.SetDefaultPassword()` festgelegte Standardpasswort verwendet.

Callback:

Mit diesem Tag können Sie eine Funktion übergeben, die von Zeit zu Zeit aufgerufen wird. Dies kann nützlich sein, wenn Sie eine Statusleiste oder etwas anzeigen möchten, während die xad-Datei entpackt wird. Die Funktion erhält eine Tabelle als einziges Argument. In der Tabelle werden folgende Felder initialisiert:

Action: Enthält die Zeichenkette "ExtractFile".

ID: Enthält die ID des XAD-Archivs.

Progress:

Enthält einen Wert zwischen 0 und 100, der angibt, wie viel Prozent der Arbeit bereits erledigt wurde.

UserData:

Enthält den Wert, den Sie im Argument `UserData` übergeben haben (siehe unten).

Sie können auch Benutzerdaten, die an Ihrer Callback-Funktion weitergeleitet werden sollen, mit dem Tag unten verwenden.

UserData:

Dieser Tag kann auf beliebige Daten festgelegt werden, die an die Callback-Funktion übergeben werden sollen, den Sie im Tag `Callback` übergeben haben. Wenn Sie diesen ohne den Tag `Callback` angeben, wird der Tag `UserData` einfach ignoriert.

EINGABEN

<code>id</code>	ID des zu verwendenden XAD-Archivs
<code>idx</code>	Index der zu entpackenden Datei
<code>dst\$</code>	gewünschte Zielfile
<code>table</code>	optional: Tabelle mit weiteren Parametern

4.3 `xad.GetFilesAtIndex`

BEZEICHNUNG

`xad.GetFilesAtIndex` – gibt den Namen der Datei beim Index zurück

ÜBERSICHT

```
name$ = xad.GetFilesAtIndex(id, idx)
```

BESCHREIBUNG

Dieser Befehl gibt den Namen der Datei beim Index `idx` in dem in `id` angegebene `xad`-Archiv zurück.

Um die Anzahl der Dateien in einem XAD-Archiv herauszufinden, können Sie `#XADATTRNUMENTRIES` mit dem Hollywood-Befehl `GetAttribute()` abfragen. Siehe [Abschnitt 4.5 \[`xad.GetObjectType`\], Seite 13](#), für Details.

EINGABEN

<code>id</code>	Identifikator des zu verwendenden XAD-Archivs
<code>idx</code>	Index (im Bereich von 0 bis Anzahl der Einträge minus 1)

RÜCKGABEWERTE

`name$` Name des Eintrags beim Index `idx`

4.4 `xad.GetFilesAttributes`

BEZEICHNUNG

`xad.GetFilesAttributes` – ruft Dateiattribute ab

ÜBERSICHT

```
t = xad.GetFilesAttributes(id, idx)
```

BESCHREIBUNG

Dieser Befehl gibt Attribute der Datei an dem von `idx` angegebenen Index innerhalb des durch `id` angegebenen `xad`-Archivs zurück. `xad.GetFilesAttributes()` gibt eine Tabelle mit folgenden Informationen über die Datei zurück:

- Type:** Dies ist `#DOSTYPE_FILE`, wenn der Eintrag eine Datei ist oder `#DOSTYPE_DIRECTORY`, wenn der Eintrag ein Verzeichnis ist.
- Name:** Dieses Feld enthält eine Zeichenkette mit dem vollständigen Pfad zu dieser Datei oder diesem Verzeichnis.
- Size:** Die Größe der Datei in Bytes oder 0 für Verzeichnisse.
- CompressedSize:**
Die komprimierte Größe der Datei in Byte oder 0 für Verzeichnisse.
- Encryption:**
Dies wird auf `True` gesetzt, wenn die Datei mit Verschlüsselung gespeichert wurde, andernfalls auf `False`.
- Flags:** Dieses Feld erhält eine Kombination von Schutzbits der Datei oder des Verzeichnisses. Für Einzelheiten zu möglichen Werten dieses Feldes siehe die Dokumentation des Befehls `GetFilesAttributes()` in Ihrem Hollywood Handbuch.
- Time:** Der Datumsstempel für die Datei. Dies wird im Standard-Hollywood-Datumsformat sein: `dd-mmm-yyyy hh:mm:ss`

EINGABEN

- `id` ID des zu verwendenden XAD-Archivs
- `idx` Index der abzufragenden Datei

RÜCKGABEWERTE

- `t` Tabelle mit Dateiattributen

4.5 xad.GetObjectType

BEZEICHNUNG

`xad.GetObjectType` – gibt den `xad`-Archivobjekttyp zurück

ÜBERSICHT

```
type = xad.GetObjectType()
```

BESCHREIBUNG

Dieser Befehl gibt den Objekttyp zurück, der von `xad`-Archiven verwendet wird, die mit dem Befehl `xad.OpenArchive()` geöffnet wurden. Sie können diesen Objekttyp dann mit Befehlen aus der Objektbibliothek von Hollywood wie `GetAttribute()`, `SetObjectData()`, `GetObjectData()` usw. verwenden.

Insbesondere kann der Befehl `GetAttribute()` von Hollywood verwendet werden, um bestimmte Eigenschaften von xad-Archiven, die mit `xad.OpenArchive()` geöffnet wurden, abzufragen. Die folgenden Attribute werden derzeit von `GetAttribute()` für xad-Archive unterstützt:

#XADATTRNUMENTRIES:

Gibt die Anzahl der Einträge im xad-Archiv zurück.

#XADATTRTYPE:

Gibt eine Zeichenkette zurück, die den Archivierungsnamen beschreibt, der vom aktuellen xad-Archiv verwendet wird, z.B. "RAR" oder "LhA".

EINGABEN

keine

RÜCKGABEWERTE

`type` interner xad-Archivtyp zur Verwendung mit Hollywoods Objektbibliothek

BEISPIEL

```
xad.OpenArchive(1, "test.rar")
XAD_ARCHIVE = xad.GetObjectType()
numentries = GetAttribute(XAD_ARCHIVE, 1, #XADATTRNUMENTRIES)
type$ = GetAttribute(XAD_ARCHIVE, 1, #XADATTRTYPE)
```

Der obige Code öffnet `test.rar` und fragt die Anzahl der Einträge im Archiv und dessen Typ über `GetAttribute()` ab.

4.6 xad.LocateFile

BEZEICHNUNG

`xad.LocateFile` – sucht nach einer Datei im XAD-Archiv

ÜBERSICHT

```
idx = xad.LocateFile(id, name$[, table])
```

BESCHREIBUNG

Dieser Befehl sucht nach der Datei `name$` innerhalb des in `id` angegebenen xad-Archivs und gibt ihren Index zurück, wenn sie gefunden wurde, andernfalls -1.

Mit dem optionalen Argument `table` können weitere Optionen angegeben werden. Folgende Tabellentags werden derzeit erkannt:

NoCase: Wenn dieser Tag auf `True` gesetzt ist, unterscheidet `xad.LocateFile()` nicht zwischen Groß- und Kleinbuchstaben. Dies macht die Suche langsamer. Der Standardwert ist `False`.

NoDir: Wenn dieser Tag auf `True` gesetzt ist, wird `xad.LocateFile()` nur nach Dateinamen suchen. So wird dieser Befehl auch fündig, wenn sich die Datei in einem Unterverzeichnis im Archiv befindet. Der Standardwert ist `False`.

EINGABEN

`id` Identifikator des zu verwendenden XAD-Archivs

`name$` Name der zu suchenden Datei
`table` optional: Tabellenargument mit weiteren Optionen (siehe oben)

RÜCKGABEWERTE

`idx` Index der Datei im XAD-Archiv oder -1, wenn sie nicht gefunden wurde

4.7 xad.OpenArchive**BEZEICHNUNG**

`xad.OpenArchive` – öffnet ein XAD-Archiv

ÜBERSICHT

`[id] = xad.OpenArchive(id, filename$)`

BESCHREIBUNG

Dieser Befehl versucht, das mit `filename$` angegebene xad-Archiv zu öffnen und weist ihr die ID in `id` zu. Wenn Sie `Nil` in `id` übergeben, wählt `xad.OpenArchive()` automatisch eine freie ID aus und gibt sie zurück. Wenn die Datei nicht existiert, schlägt dieser Befehl fehl.

Obwohl `xad.hwp` automatisch alle geöffneten xad-Archive beim Beenden schließt, wird dringend empfohlen, dass Sie ein geöffnetes xad-Archiv schließen, wenn Sie damit fertig sind. Verwenden Sie dafür den Befehl `xad.CloseArchive()`, da sonst Ressourcen verschwendet werden.

Beachten Sie, dass `xad.OpenArchive()` ein Standard-Hollywood-Objekt erstellt, das auch mit Befehlen aus Hollywoods Objektbibliothek wie `GetAttribute()`, `SetObjectData()`, `GetObjectData()` usw. verwendet werden kann. Siehe [Abschnitt 4.5 \[xad.GetObjectType\]](#), [Seite 13](#), für Details.

EINGABEN

`id` Identifikator der Datei oder `Nil` für die automatische ID-Auswahl
`filename$`
 Name der zu öffnenden Datei

RÜCKGABEWERTE

`id` optional: ID der Datei; wird nur zurückgegeben, wenn Sie `Nil` als Argument in `id` übergeben (siehe oben)

4.8 xad.SetDefaultPassword**BEZEICHNUNG**

`xad.SetDefaultPassword` – legt das Standardpasswort fest

ÜBERSICHT

`xad.SetDefaultPassword(pwd$)`

BESCHREIBUNG

Dieser Befehl kann verwendet werden, um ein Standardpasswort festzulegen, das zum Entschlüsseln von Dateien verwendet wird, wenn kein anderes Passwort angegeben wird. Wenn Sie in `pwd$` eine leere Zeichenkette übergeben, wird das Standardpasswort nicht gesetzt.

Das Standardpasswort wird von `xad.ExtractFile()` verwendet, wenn kein anderes Passwort explizit angegeben ist. Es wird auch von normalen Hollywood-Befehlen verwendet, wenn der globale Adapter für `xad.hwp` aktiviert wurde, indem der Tag `InstallAdapter` auf `True` gesetzt wird.

EINGABEN

`pwd$` neues Standardpasswort oder leere Zeichenfolge, um das Standardpasswort zu deaktivieren

Anhang A Lizenzen

A.1 xadmaster.library-Lizenz

The xadmaster.library is Shareware with very special conditions. There are no usage restrictions (means this is no Crippleware!), and xadmaster.library use can be implemented into programs without any restrictions. Only the end users have to pay shareware fee, if using xadmaster.library. There are 3 forms of payment to get a registered user:

- Send 20 EUR or US \$20 to my snail mail address or use any other more secure transfer system.
- Make detailed bug reports about 3 currently unknown bugs. I decide whether a bug report is counted or not. Mainly this depends on two facts: was the bug already reported / does the description help me to find the bug
- You make an own external client and send me the source and the client for inclusion in next release. Some more are good as well!

For programmers using xadmaster.library in own applications (f.e. in virus scanners):

- It would be fine, if you would tell me suggestions or bugs, as implementing xadmaster.library use may show you some problems.
- You must state above conditions in your distribution, at a place, where users can read it.
- And please do not forget: Most time you are users as well, so think about above conditions.

NOTE: This conditions only cover the xadmaster.library! External clients may have different legal state. For example LZK and Zoom both are Freeware. So expect clients to have any legal state. You need to check this for every client. Normally the version string in the client should tell you what type it is.

A.2 LGPL-Lizenz

GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

- a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
- b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

- a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

- a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
- b) Accompany the Combined Work with a copy of the GNU GPL and this license document.
- c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.

d) Do one of the following:

0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.

1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.

e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

Index

xad.CloseArchive	11	xad.GetObjectType	13
xad.ExtractFile	11	xad.LocateFile	14
xad.GetFileAtIndex	12	xad.OpenArchive	15
xad.GetFileAttributes	12	xad.SetDefaultPassword	15

