Name (名称)

- OsSAL/AdDataPackager

- 名称：**先進データパッケージャー** (Advanced Data Packager)
  略称：**アドパック** (ADPACK)
  バージョン：**Beta5版** (2023/02/01)
  開発：OsSAL.org (オープンソース署名＆認証ラボ)

Abstract (概要)

アドパックはハッシュ値による検証機能を備えたZIP形式のパッケージファイルを、作成（パック）して、検証した上で解凍（アンパック）する為のツールです。

パッケージファイルの仕様はJIIMA発行の「測定機器データの長期保存技術ガイドブック第1.0版」のパッケージに準拠しています。ガイドブックは以下より取得できます。パッケージは汎用的仕様になっているので「測定機器データ」以外への利用も可能です。

- **JIIMA 測定機器データの長期保存技術ガイドブック第1.0版**

パック時にはパック対象の各ファイルのハッシュ値を、"META-INF/Manifest.xml"に列挙 して作成してからZIP圧縮します。アンパック時には、"META-INF/Manifest.xml"に列挙が されたハッシュ値と圧縮されている各ファイルのハッシュ値を比較して確認してからZIP 解凍します。"META-INF/Manifest.xml"はW3CのXML署名の仕様に従っています。

- W3C XML署名 / XML Signature Syntax and Processing Version 2.0

Functions (主な機能)

アドパックでは **パッケージ生成処理/Pack** と **パッケージ解凍処理/UnPack** の2つが提供されます。

* Pack Operations (パッケージ生成処理)

1. META-INF/Index.xmlの自動生成
   eng) Create META-INF/Index.xml

2. META-INF/Manifest.xmlの対象各ファイルのハッシュ値からの自動生成
   eng) Create META-INF/Manifest.xml with hash values

3. 対象ファイルと生成したMETA-INFファイルをまとめてZIPパッケージ化
   eng) Packing to ZIP file with META-INF/*

* UnPack Operations (パッケージ解凍処理)

1. META-INF/Manifest.xmlによる各ファイルのハッシュ値比較(改ざんチェック)
   eng) Check hash values include files by META-INF/Manifest.xml

2. 各ファイルのZIP解凍処理
   eng) Extract to all file

Configuration (ツール構成)

アドパックは**メインツール adpack** と **オプションツール adtool** の2つが提供されます。

* adpack (メインツール)

1. パッケージファイルの生成機能(Manifest.xmlとIndex.xmlの自動生成)
   eng) Create package file. (create Manifest.xml and Index.xml)
2. [オプション] プレ実行ファイル指定 (例)独自形式の Index.xml 生成
   eng) [opt] pre execute file. (ex) create origintal Index.xml file
3. [オプション] ポスト実行ファイル指定 (例)署名やタイムスタンプ取得等の実行
   eng) [opt] post execute file. (ex) add Timestamp.tst or Signature.xml file
4. パッケージファイルの解凍機能(Manifest.xmlによる各ファイルハッシュ値比較)
   eng) Extract package file with Manifest.xml hash values check
5. [オプション] 検証実行ファイル指定 (例)署名やタイムスタンプ等の検証の実行
   eng) [opt] verify execute file. (ex) verify Timestamp.tst or Signature.xml file

* adtool (オプションツール)

1. RFC 3161形式のタイムスタンプ取得 (例)URL指定してTimestamp.tstの生成
   eng) Get RFC 3161 TimeStamp Token. (ex) create Timestamp.tst by URL
2. RFC 3161形式のタイムスタンプ検証 (注)TSA証明書の検証は行いません
   eng) Verify RFC 3161 TimeStamp Token. (notice) not verify TSA-Cert chain.
3. 指定したファイルからハッシュ値の計算と表示
   eng) Get hash value from specify file.

DEMO (デモ：使い方の例)

STD:標準機能 (adpackのみ)

- A1) パッケージ生成(パック) : test_pack.bat
- A2) パッケージチェック : test_check.bat
- A3) パッケージ解凍(アンパック) : test_unpack.bat

OPT:オプション機能 (adtool呼び出しによるタイムスタンプ利用)

- B1) タイムスタンプ付きパッケージ生成(パック) : test_pack_timestamp.bat
- B2) タイムスタンプ付きパッケージ解凍(アンパック) : test_unpack_timestamp.bat

---

0) 前準備(フォルダ移動)

```
cmd> cd AdDataPackager/bin_win/Release
cmd>
```

A1) パッケージ生成(パック) : test_pack.bat

```
cmd> test_pack.bat

> adpack -pack -in test -out testOut.zip -force -list

# -in test         : input directory is "test".
# -out testOut.zip : output zip file is "test2.zip".
# -force           : overwrite output zip file.
# -list            : show all zipped file names.

---(execute adpack)------------------

Target from: test/

Create META-INF/Index.xml
Create META-INF/Manifest.xml

Pack to: "testOut.zip"
  aaa.txt
  data/input.pdf
  META-INF/Index.xml
  META-INF/Manifest.xml

---(end adpack)----------------------

# test_pack.bat done.

cmd>
```

A2) パッケージチェック : test_check.bat

```
cmd> test_check.bat

> adpack -unpack -in testOut.zip -info -check -list

# -in testOut.zip : input zip file is "testOut.zip".
# -info           : show Index.xml infos.
# -check          : only Manifest.xml hash check (no extract files).
# -list           : show all file names.

---(execute adpack)------------------

Package file:
  testOut.zip
Index info:
  Title: test
  Date: 2022-02-06T21:36:33+09:00
  User: miyachi
  Host: DINOSAUR
File list:
  aaa.txt (3 bytes)
  data/input.pdf (9438 bytes)
  META-INF/Index.xml (113 bytes)
  META-INF/Manifest.xml (670 bytes)

Check from: "testOut.zip"
  aaa.txt ...ok
  data/input.pdf ...ok
  META-INF/Index.xml ...ok

---(end adpack)----------------------

# test_check.bat done.
```

```
cmd>
```

A3) パッケージ解凍(アンパック) : test_unpack.bat

```
cmd> test_unpack.bat

> adpack -unpack -in testOut.zip -force -list

# -in testOut.zip : input zip file is "testOut.zip".
# -force          : overwrite output extract files.
# -list           : show all file names.

---(execute adpack)------------------

Check from: "testOut.zip"
  aaa.txt ...ok
  data/input.pdf ...ok
  META-INF/Index.xml ...ok

Extract to: "testOut/"
  aaa.txt
  data/input.pdf
  META-INF/Index.xml
  META-INF/Manifest.xml

---(end adpack)----------------------

# test_unpack.bat done.

cmd>
```

B1) タイムスタンプ付きパッケージ生成(パック) : test_pack_timestamp.bat

```
cmd> test_pack_timestamp.bat

> adpack -pack -in test -out testTst.zip -force -post sample\\tst_get.bat -list

# -in test                  : input directory is "test".
# -out testTst.zip          : output zip file is "testTst.zip".
# -post sample\\tst_get.bat : post execute for get timestamp.
# -force                    : overwrite output zip file.
# -list                     : show all zipped file names.

---(execute adpack)------------------

Target from: test/

Create META-INF/Index.xml
Create META-INF/Manifest.xml

Post Execute: 'sample\\tst_get.bat "test/" "test/META-INF/Manifest.xml"'
Output tst to: 'test/META-INF/Timestamp.tst'

Pack to: "testTst.zip"
  aaa.txt
  data/input.pdf
  META-INF/Index.xml
  META-INF/Manifest.xml
  META-INF/Timestamp.tst
```

```
---(end adpack)----------------------

# test_pack_timestamp.bat done.

cmd>
```

B2) タイムスタンプ付きパッケージ解凍(アンパック) : test_unpack_timestamp.bat

```
cmd> test_unpack_timestamp.bat

> adpack -unpack -in testTst.zip -force -verify sample\\tst_verify.bat -list

# -in testTst.zip               : input zip file is "testTst.zip".
# -force                        : overwrite output extract files.
# -verify sample\\tst_verify.bat : verify execute for timestamp.
# -list                         : show all file names.

---(execute adpack)-----------------

Check from: "testTst.zip"
  aaa.txt ...ok
  data/input.pdf ...ok
  META-INF/Index.xml ...ok

Extract to: "testTst/"
  aaa.txt
  data/input.pdf
  META-INF/Index.xml
  META-INF/Manifest.xml
  META-INF/Timestamp.tst

Verify Execute: 'sample\\tst_verify.bat "testTst/"'
Timestamp Token Check: valid.
Timestamp Date: 2022-02-06T21:42:45 (20220206124245Z)
Target Hash Value: SHA-2(256)
  0x52090BF106CDF6FE03271FAE61C3DE108115377B38E32FEA709D30DA77AEF3D4
TSA Certificate:
  Subject: CN=LE TSA 200020,OU=demo user,O=LangEdge,C=JP
  Issuer : CN=LangEdge CA Root 02,OU=demo,O=LangEdge,C=JP
  Serial : 0x200020
  Not Before: 2018-04-01T09:00:00 (20180401000000Z)
  Not After : 2026-04-01T09:00:00 (20260401000000Z)
  * NOTICE: TSA certificates chain is not check.

---(end adpack)----------------------

# test_unpack_timestamp.bat done.

cmd>
```

Requirement (依存ライブラリ)

	ZipLib	libxml2	OpenSSL
adpack	*USED	*USED	---
adtool	---	*USED	*USED

Usage (使用方法)

adpack (メインツール)

```
AdDataPackager : Advanced Data Packager (adpack) Beta2
Copyright (c) OsSAL.org. All Rights Reserved.

 * Create Pakcage (Zip) file.

  > adpk -pack [-options] -in <target-dir> ( -out <package-path> )

    -in  <target-dir>   : specify input target data directory. (require)
    -out <package-path> : specify output package zip file path. (optional)
                          [default package name is "<target-dir>.zip"]

  options: optional arguments.

    -hash <s256/s384/s512/sha1> : specify hash algorithm. [default=s256]
       s256: SHA256(SHA-2/32bytes). [default setting]
       s384: SHA384(SHA-2/48bytes).
       s512: SHA512(SHA-2/64bytes).
       sha1: SHA-1(20bytes). [not recommended]
    -force  : specify no confirm overwriting zip file.
    -silent : do not output operation informations.
    -list   : show file names being processed.
    -pre <pre-exec>   : specify pre execute file. (optional)
        execute "<pre-exec> <target-dir>" before manifest.
        [default create simple META-INF/Index.xml]
        note: create META-INF/Index.xml file.
    -post <post-exec> : specify post execute file. (optional)
        execute "<post-exec> <target-dir> <manifest-path>" after manifest.
        [default nothing to do.]
        note: create META-INF/Signature.xml or META-INF/Timestamp.tst file.

  example:

    ex1) packing 'testdata' directory.
      > adpk -pack -in testdata
    ex2) packing current directory with specify hash alrorithm.
      > adpk -pack -hash s512 -in .
    ex3) show packing file name and specify output package (zip) file.
      > adpk -pack -list -in testdata -out mypackage.zip
    ex4) specify pre-exec for add original 'META-INF/Index.xml'.
      > adpk -pack -in testdata -pre index_make.bat
    ex5) specify post-exec for add 'META-INF/Signature.xml'.
      > adpk -pack -in testdata -post add_sign.bat
    ex6) specify pre-exec and post-exec for add 'META-INF/Timestamp.tst'.
      > adpk -pack -in testdata -pre index_make.bat -post add_timestamp.bat

 * UnPakcage (UnZip) file.

  > adpk -unpack [-options] -in <package-path> ( -out <extract-dir> )

    -in  <package-path> : specify input package zip file path. (require)
    -out <extract-dir>  : specify output extract directory. (optional)

  options: optional arguments.

    -info    : show package information.
    -force   : specify no confirm overwriting output extract directory.
    -silent  : do not output operation informations.
    -list    : show file names being processed.
```

```
    -check   : check hash value of target file of 'Manifest.xml'. (no extract)
    -nocheck : extract (unzip) only. (no check)
    -verify <verify-exec> : specify verify execute file. (optional)
        execute "<verify-exec> <extract-dir>" after check
        [default nothing to do.]
        note: verify 'META-INF/Signature.xml' file.

  example:

    ex1) simple (extract and check).
      > adpk -unpack -in testdata.zip
    ex2) package check only and specify extract-dir.
      > adpk -unpack -check -in testdata.zip -out "C:\temp\"
    ex3) package extract and check and show file list.
      > adpk -unpack -list -in testdata.zip
    ex4) package check only and no extract (delete extract directory).
      > adpk -unpack -check -in testdata.zip
    ex5) extract and check and specify verify-exec.
      > adpk -unpack -in testdata.zip -verify verify.bat

 * Utility.

  > adpk -opt [-options] -in <file-path>

    -in  <file-path> : specify input file path. (require)

  options: optional arguments.

    -hash <s256/s384/s512/sha1> : show hash value. [default=s256]
       s256: SHA256(SHA-2/32bytes). [default setting]
       s384: SHA384(SHA-2/48bytes).
       s512: SHA512(SHA-2/64bytes).
       sha1: SHA-1(20bytes).
    -hex : show hash value by HEX string. [default=off(Base64)]

  example:

    ex1) show Base64 SHA256 hash value of target.file.
      > adpk -opt -hash -in tagert.file
    ex2) show HEX SHA512 hash value of target.file.
      > adpk -opt -hash s512 -hex -in tagert.file
```

adtool (オプションツール)

```
AdDataTool : Advanced Data Tool (adtool) Beta2
Copyright (c) OsSAL.org. All Rights Reserved.

 * Get Timestamp Token from RFC 3161 server.

  > adtool -get [-options] -target <hash-target> -out <tst-file>

    -target <hash-target> : specify input hash target file path. (require)
    -out <tst-file> : specify output timestamp token (tst) file path. (require)

  options: optional arguments.

    -hash <s256/s512/sha1> : specify hash algorithm. [default=s256]
       s256: SHA256(SHA-2/32bytes). [default setting]
       s512: SHA512(SHA-2/64bytes).
       sha1: SHA-1(20bytes). [not recommended]
    -force : specify no confirm overwriting tst file.
    -ts <url> [id] [passwd] : specify timestamp server url.
              (opt: [id] and [passwd] are use for BasicAuth)
```

```
  example:

    ex) get tmp.txt timestamp token from https://www.langedge.jp/tsa.
      > adtool -get -ts https://www.langedge.jp/tsa -target tmp.txt -out tmp.tst

 * Verify Timestamp Token.

  > adtool -verify [-options] -target <hash-target> -in <tst-file>

    -target <hash-target> : specify input hash target file path. (require)
    -in <tst-file> : specify input timestamp token (tst) file path. (require)

  options: optional arguments.

    -detail : show verify detail.

  example:

    ex) verify tmp.tst
      > adtool -verify -target tmp.txt -in tmp.tst
```

Author (作者)

- Author: Naoto Miyachi / OsSAL.org
- E-mail: miyachi@langedge.jp

License (ライセンス)

History (履歴)

Beta5: 2023/02/01

1. 多階層の空のディレクトリがPack出来ない場合がある問題の修正。
   eng) Fixed a problem where empty multi-level directories could not be packed.

Beta4: 2023/01/27

1. 空のディレクトリがPack出来ない問題の修正。
   eng) Fixed a problem where empty directories could not be packed.
2. 多階層のディレクトリがUnPack出来ない問題の修正。
   eng) Fixed a problem that multi-level directories cannot be UnPacked.

Beta3: 2022/12/26

1. SHIFT-JISの"ソ"・"表"・"十"等のバックスラッシュを含む文字のPack時の問題の修正。
   eng) Fixed a problem when packing file names with backslashes in multibyte.

Beta2: 2022/11/30

1. ファイルサイズがゼロのファイルへの対応修正。
   eng) Fixed a problem of size zero file.

2. マルチバイトファイル名のUnPack時の問題の修正。
   eng) Fixed a problem when unpacking multibyte file names.

3. Zipファイル内のファイル削除時にエラーにならない問題の修正。
   eng) Fixed a problem that an error does not occur when deleting files in Zip files.

Beta1: 2022/03/18

1. 最初のリリース
   eng) 1st release version.