HIArchive Reference

Carbon > Human Interface Toolbox

2005-08-11

Contents

HIArchive Reference

Framework:	Carbon/Carbon.h
Declared in	HIArchive.h

Overview

HIArchive provides a convenient and standardized mechanism for flattening data objects so they can be stored in memory or on disk. Applications can use these archives whenever they need to package complex data. For example, you can use archives to:

- Store document data
- Transfer data using pasteboards, drag-and-drop, streams, or Apple events
- Store localization strings and user interface elements in the same package

HIArchive encodes archives in the binary property list format. You can convert archives to a text XML format using the `plutil` property list tool accessible from Terminal. HIArchive is comparable to (and uses the same underlying mechanism as) the Cocoa NSKeyedArchiver/Unarchiver classes.

For details about using HIArchive, see *HIArchive Programming Guide*.

HIArchive is available in Mac OS X version 10.4 and later.

Functions by Task

Storing Objects in an Archive

`HIArchiveCreateForEncoding` (page 8)
Creates an HIArchive object to store objects.

`HIArchiveEncodeBoolean` (page 10)
Stores a Boolean value in an archive.

`HIArchiveEncodeNumber` (page 11)
Stores a number in an archive.

`HIArchiveEncodeCFType` (page 10)
Stores a CFType object in an archive.

`HIArchiveCopyEncodedData` (page 7)
Compresses an archive for storage.

Retrieving Objects from an Archive

`HIArchiveCreateForDecoding` (page 7)
Creates an HIArchive object to retrieve objects.

`HIArchiveDecodeBoolean` (page 8)
Retrieves a Boolean value from an archive.

`HIArchiveDecodeNumber` (page 9)
Retrieves a number from an archive.

`HIArchiveCopyDecodedCFType` (page 6)
Retrieves a CFType object from an archive.

Miscellaneous Function

`HIArchiveGetTypeID` (page 12)
Obtains the CFType ID for HIArchive objects.

Functions

HIArchiveCopyDecodedCFType

Retrieves a CFType object from an archive.

```
OSStatus HIArchiveCopyDecodedCFType (
   HIArchiveRef inDecoder,
   CFStringRef inKey,
   CFTypeRef *outCFType
);
```

Parameters

inDecoder
The archive holding the CFType object to retrieve.

inKey
A Core Foundation string key identifying the CFType object to retrieve.

outCFType
On return, `outCFType` points to the retrieved CFType object.

Return Value
A result code.

Discussion
You also use this function for retrieving HIObjects and objects subclassed from HIObject.

Availability
Available in Mac OS X v10.4 and later.
Not available to 64-bit applications.

Declared In
`HIArchive.h`

HIArchiveCopyEncodedData

Compresses an archive for storage.

```
OSStatus HIArchiveCopyEncodedData (
   HIArchiveRef inEncoder,
   CFDataRef *outData
);
```

Parameters

inEncoder

The archive to compress.

outData

On return, `outData` points to the compressed archive.

Return Value

A result code.

Discussion

When you have finished adding data to an archive, calling `HIArchiveCopyEncodedData` compresses the data and returns it to you as a CFData object. You can use the returned data reference to store or transfer the data as you choose, for example writing it to a file or copying it to a pasteboard.

After compression, you can release the original HIArchive reference by calling `CFRelease`.

Availability

Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared In

`HIArchive.h`

HIArchiveCreateForDecoding

Creates an HIArchive object to retrieve objects.

```
OSStatus HIArchiveCreateForDecoding (
   CFDataRef inData,
   OptionBits inOptions,
   HIArchiveRef *outDecoder
);
```

Parameters

inData

A CFData reference pointing to archived data. This archive was originally written to a data stream using `HIArchiveCopyEncodedData` (page 7). This data reference does not have to be the one originally returned by `HIArchiveCopyEncodedData` (page 7), but it must contain a copy of the same data.

inOptions

Any decoding options. Currently the only option is `kHIArchiveDecodeSuperclassForUnregisteredObjects`.

outDecoder

On return, `outDecoder` points to the newly created HIArchive object.

Return Value
A result code.

Discussion
You use this function when you want to retrieve data from an existing HIArchive.

Availability
Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared In
`HIArchive.h`

HIArchiveCreateForEncoding

Creates an HIArchive object to store objects.

```
OSStatus HIArchiveCreateForEncoding (
   HIArchiveRef *outEncoder
);
```

Parameters

outEncoder

On return, `outEncoder` points to the newly created HIArchive object.

Return Value
A result code.

Discussion
Before you can archive any objects, you must create an HIArchive object in which to store them.

Availability
Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared In
`HIArchive.h`

HIArchiveDecodeBoolean

Retrieves a Boolean value from an archive.

```
OSStatus HIArchiveDecodeBoolean (
   HIArchiveRef inDecoder,
   CFStringRef inKey,
   Boolean *outBoolean
);
```

Parameters

inDecoder

The archive holding the Boolean value.

inKey

A Core Foundation string key identifying the Boolean to retrieve.

outBoolean
: On return, `outBoolean` points to the retrieved Boolean value.

Return Value
A result code.

Discussion
This function is a convenience wrapper that calls `HIArchiveCopyDecodedCFType` (page 6) to obtain a CFBoolean value.

Availability
Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared In
`HIArchive.h`

HIArchiveDecodeNumber

Retrieves a number from an archive.

```
OSStatus HIArchiveDecodeNumber (
   HIArchiveRef inDecoder,
   CFStringRef inKey,
   CFNumberType inNumberType,
   void *outNumberValue
);
```

Parameters

inDecoder
: The archive holding the number to retrieve.

inKey
: A Core Foundation string key identifying the number to retrieve.

inNumberType
: A CFNumber type identifying the type of number value to be retrieved. For example, `kCFNumberSInt32Type`. See CFNumber Reference in Core Foundation Reference Documentation for additional possible values.

outNumberValue
: Before calling, `outNumberValue` must point to a number variable of the type and size you specified in `inNumberType`. On return, `outNumberValue` points to the retrieved number.

Return Value
A result code.

Discussion
This function is a convenience wrapper that calls `HIArchiveCopyDecodedCFType` (page 6) to obtain a CFNumber value.

Availability
Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared In
`HIArchive.h`

HIArchiveEncodeBoolean

Stores a Boolean value in an archive.

```
OSStatus HIArchiveEncodeBoolean (
   HIArchiveRef inEncoder,
   CFStringRef inKey,
   Boolean inBoolean
);
```

Parameters

inEncoder
: The archive to store the Boolean value.

inKey
: A Core Foundation string key identifying the Boolean value.

inBoolean
: The Boolean value.

Return Value
A result code.

Discussion
This function is a convenience wrapper that calls `HIArchiveEncodeCFType` (page 10) with a CFBoolean value.

Availability
Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared In
`HIArchive.h`

HIArchiveEncodeCFType

Stores a CFType object in an archive.

```
OSStatus HIArchiveEncodeCFType (
   HIArchiveRef inEncoder,
   CFStringRef inKey,
   CFTypeRef inCFType
);
```

Parameters

inEncoder
: The archive to store the CFType object.

inKey
: A Core Foundation string key identifying the CFType object.

inCFType
: The CFType object to store in the archive.

Return Value
A result code.

Discussion
You can only encode base CFType objects that correspond to archivable NSFoundation objects. For example, type `CFStringRef` is supported, but type `HIShapeRef` is not.

You also use this function for storing HIObjects and objects subclassed from HIObject. Currently only the following HIObject subclass types support archiving:

- `HIObjectRef`
- `HIViewRef`
- `WindowRef`
- `ControlRef`
- `MenuRef`

Availability
Available in Mac OS X v10.4 and later.
Not available to 64-bit applications.

Declared In
`HIArchive.h`

HIArchiveEncodeNumber

Stores a number in an archive.

```
OSStatus HIArchiveEncodeNumber (
   HIArchiveRef inEncoder,
   CFStringRef inKey,
   CFNumberType inNumberType,
   const void *inNumberValue
);
```

Parameters

inEncoder
The archive to store the number.

inKey
A Core Foundation string key identifying the number.

inNumberType
A CFNumber type identifying the type of number value to be stored, for example, `kCFNumberSInt32Type`. See CFNumber Reference in Core Foundation Reference Documentation for additional possible values.

inNumberValue
A pointer to the number value.

Return Value
A result code.

Discussion
This function is a convenience wrapper that calls `HIArchiveEncodeCFType` (page 10) with a CFNumber value.

Availability
Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared In
`HIArchive.h`

HIArchiveGetTypeID

Obtains the CFType ID for HIArchive objects.

```
CFTypeID HIArchiveGetTypeID (
   void
);
```

Return Value
The Core Foundation type ID for the HIArchive object type.

Availability
Available in Mac OS X v10.4 and later.

Not available to 64-bit applications.

Declared In
`HIArchive.h`

Data Types

HIArchiveRef

Defines an uncompressed archive object.

```
typedef struct OpaqueHIArchiveRef* HIArchiveRef;
```

Discussion
The structure pointed to by this reference is opaque. `HIArchiveRef` is a CFType, and therefore responds to `CFRetain` and `CFRelease` calls.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`HIArchive.h`

Constants

Archive Decoding Option Constant

Defines options available when calling `HIArchiveCreateForDecoding` (page 7).

```
enum {
  kHIArchiveDecodeSuperclassForUnregisteredObjects = (1 <<  0)
};
```

Constants

`kHIArchiveDecodeSuperclassForUnregisteredObjects`

If the class of the HIObject you are attempting to decode is not a registered subclass, this option allows `HIArchiveCopyDecodedCFType` (page 6) to instantiate the object as its superclass, if it exists. For example, if your application has not yet registered `com.myCorp.mycustomView` before attempting to unarchive an instance of that HIView, HIArchive instantiates the data as class `com.apple.hiview`. Only data written to the superclass is decoded; any data unique to the unregistered subclass is ignored. Specifying this option also signals the HIObject to load its custom archive data so you can access it by calling `HIObjectCopyCustomArchiveData`.

This option can be useful when creating an archive editor that doesn't implement all the objects contained in a client archive.

Available in Mac OS X v10.4 and later.

Declared in `HIArchive.h`.

Result Codes

Result Code	Value	Description
`noErr`	0	No error. Available in Mac OS X v10.0 and later.
`hiArchiveTypeMismatchErr`	-6780	The encoding or decoding archive was passed into a noncorresponding function. (For example, an archive created for encoding was passed into a decoding function.) Available in Mac OS X v10.4 and later.
`hiArchiveKeyNotAvailableErr`	-6781	The requested key does not exist in the specified archive. Available in Mac OS X v10.4 and later.
`hiArchiveEncodingCompleteErr`	-6782	`HIArchiveCopyEncodedData` (page 7) was called on this archive, so no more data can be added. Available in Mac OS X v10.4 and later.
`hiArchiveHIObjectIgnoresArchivingErr`	-6783	The HIObject you wanted to encode does not support the HIArchive protocol. Available in Mac OS X v10.4 and later.

Document Revision History

This table describes the changes to *HIArchive Reference*.

Date	Notes
2005-08-11	Moved conceptual information into the HIArchive Programming Guide.
2005-04-29	New document that describes the HIArchive APIs.

Index