Byte-Order Utilities Reference

Core Foundation > Data Management

2006-05-23

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Byte-Order Utilities Reference

Framework:	CoreFoundation/CoreFoundation.h
Companion guide	Memory Management Programming Guide for Core Foundation
Declared in	CFByteOrder.h

Overview

When handling binary data transmitted or shared across platforms, you need be concerned with how each platform stores numerical values. A platform stores values either in big-endian or little-endian format. On big-endian machines, such as PowerPC machines, values are stored with the most-significant bytes first in memory; on little-endian machines, such as Pentium machines, values are stored with the least-significant bytes first. A multibyte value transmitted to a platform with a different format will be misinterpreted if it is not converted properly by one of the computers.

You identify the native format of the current platform using the `CFByteOrderGetCurrent` (page 5) function. Use functions such as `CFSwapInt32BigToHost` (page 11) and `CFConvertFloat32HostToSwapped` (page 6) to convert values between different byte order formats.

Functions

CFByteOrderGetCurrent

Returns the byte order of the current computer.

```
CFByteOrder CFByteOrderGetCurrent (
   void
);
```

Return Value
The byte order of the current computer. See "Byte Order Flags" (page 16) for the list of possible return values.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFConvertDoubleHostToSwapped

Converts a 64-bit double from the host's native byte order to a platform-independent format.

```
CFSwappedFloat64 CFConvertDoubleHostToSwapped (
   double arg
);
```

Parameters

arg

The real value to convert.

Return Value

A structure holding the real value in a canonical byte order.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

CFConvertDoubleSwappedToHost

Converts a 64-bit double from a platform-independent format to the host's native byte order.

```
double CFConvertDoubleSwappedToHost (
   CFSwappedFloat64 arg
);
```

Parameters

arg

A structure holding the real value to convert.

Return Value

The real value in the host's native format.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

CFConvertFloat32HostToSwapped

Converts a 32-bit float from the host's native byte order to a platform-independent format.

```
CFSwappedFloat32 CFConvertFloat32HostToSwapped (
   Float32 arg
);
```

Parameters

arg

The real value to convert.

Return Value

A structure holding the real value in a canonical byte order.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFConvertFloat32SwappedToHost

Converts a 32-bit float from a platform-independent format to the host's native byte order.

```
Float32 CFConvertFloat32SwappedToHost (
   CFSwappedFloat32 arg
);
```

Parameters

arg

A structure holding the real value to convert.

Return Value
The real value in the host's native format.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFConvertFloat64HostToSwapped

Converts a 64-bit float from the host's native byte order to a platform-independent format.

```
CFSwappedFloat64 CFConvertFloat64HostToSwapped (
   Float64 arg
);
```

Parameters

arg

The real value to convert.

Return Value
A structure holding the real value in a canonical byte order.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFConvertFloat64SwappedToHost

Converts a 64-bit float from a platform-independent format to the host's native byte order.

```
Float64 CFConvertFloat64SwappedToHost (
   CFSwappedFloat64 arg
);
```

Parameters

arg

A structure holding the real value to convert.

Return Value

The real value in the host's native format.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

CFConvertFloatHostToSwapped

Converts a 32-bit float from the host's native byte order to a platform-independent format.

```
CFSwappedFloat32 CFConvertFloatHostToSwapped (
   float arg
);
```

Parameters

arg

The real value to convert.

Return Value

A structure holding the real value in a canonical byte order.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

CFConvertFloatSwappedToHost

Converts a 32-bit float from a platform-independent format to the host's native byte order.

```
float CFConvertFloatSwappedToHost (
   CFSwappedFloat32 arg
);
```

Parameters

arg

A structure holding the real value to convert.

Return Value

The real value in the host's native format.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFSwapInt16

Swaps the bytes of a 16-bit integer.

```
uint16_t CFSwapInt16 (
   uint16_t arg
);
```

Parameters

`arg`

The integer whose bytes should be swapped.

Return Value
The integer with its bytes swapped.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFSwapInt16BigToHost

Converts a 16-bit integer from big-endian format to the host's native byte order.

```
uint16_t CFSwapInt16BigToHost (
   uint16_t arg
);
```

Parameters

`arg`

The integer whose bytes should be swapped.

Return Value
The integer with its bytes swapped. If the host is big-endian, this function returns *`arg`* unchanged.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFSwapInt16HostToBig

Converts a 16-bit integer from the host's native byte order to big-endian format.

```
uint16_t CFSwapInt16HostToBig (
   uint16_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value

The integer with its bytes swapped. If the host is big-endian, this function returns *arg* unchanged.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

CFSwapInt16HostToLittle

Converts a 16-bit integer from the host's native byte order to little-endian format.

```
uint16_t CFSwapInt16HostToLittle (
   uint16_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value

The integer with its bytes swapped. If the host is little-endian, this function returns *arg* unchanged.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

CFSwapInt16LittleToHost

Converts a 16-bit integer from little-endian format to the host's native byte order.

```
uint16_t CFSwapInt16LittleToHost (
   uint16_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value

The integer with its bytes swapped. If the host is little-endian, this function returns *arg* unchanged.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFSwapInt32

Swaps the bytes of a 32-bit integer.

```
uint32_t CFSwapInt32 (
   uint32_t arg
);
```

Parameters

`arg`

The integer whose bytes should be swapped.

Return Value
The integer with its bytes swapped.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFSwapInt32BigToHost

Converts a 32-bit integer from big-endian format to the host’s native byte order.

```
uint32_t CFSwapInt32BigToHost (
   uint32_t arg
);
```

Parameters

`arg`

The integer whose bytes should be swapped.

Return Value
The integer with its bytes swapped. If the host is big-endian, this function returns *`arg`* unchanged.

Availability
Available in Mac OS X v10.0 and later.

Related Sample Code
AudioQueueTools

TabsShowcase

Declared In
`CFByteOrder.h`

CFSwapInt32HostToBig

Converts a 32-bit integer from the host’s native byte order to big-endian format.

```
uint32_t CFSwapInt32HostToBig (
   uint32_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value

The integer with its bytes swapped. If the host is big-endian, this function returns *arg* unchanged.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

AudioQueueTools

HIFleetingControls

RecordAudioToFile

TabsShowcase

Declared In

`CFByteOrder.h`

CFSwapInt32HostToLittle

Converts a 32-bit integer from the host's native byte order to little-endian format.

```
uint32_t CFSwapInt32HostToLittle (
   uint32_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value

The integer with its bytes swapped. If the host is little-endian, this function returns *arg* unchanged.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

CFSwapInt32LittleToHost

Converts a 32-bit integer from little-endian format to the host's native byte order.

```
uint32_t CFSwapInt32LittleToHost (
   uint32_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value

The integer with its bytes swapped. If the host is little-endian, this function returns *arg* unchanged.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

CFSwapInt64

Swaps the bytes of a 64-bit integer.

```
uint64_t CFSwapInt64 (
   uint64_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value

The integer with its bytes swapped.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

CFSwapInt64BigToHost

Converts a 64-bit integer from big-endian format to the host's native byte order.

```
uint64_t CFSwapInt64BigToHost (
   uint64_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value

The integer with its bytes swapped. If the host is big-endian, this function returns *arg* unchanged.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFSwapInt64HostToBig

Converts a 64-bit integer from the host's native byte order to big-endian format.

```
uint64_t CFSwapInt64HostToBig (
   uint64_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value
The integer with its bytes swapped. If the host is big-endian, this function returns *arg* unchanged.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFSwapInt64HostToLittle

Converts a 64-bit integer from the host's native byte order to little-endian format.

```
uint64_t CFSwapInt64HostToLittle (
   uint64_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value
The integer with its bytes swapped. If the host is little-endian, this function returns *arg* unchanged.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

CFSwapInt64LittleToHost

Converts a 64-bit integer from little-endian format to the host's native byte order.

```
uint64_t CFSwapInt64LittleToHost (
   uint64_t arg
);
```

Parameters

arg

The integer whose bytes should be swapped.

Return Value

The integer with its bytes swapped. If the host is little-endian, this function returns *arg* unchanged.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

Data Types

CFSwappedFloat32

Structure holding a 32-bit float value in a platform-independent byte order.

```
struct CFSwappedFloat32 {
    uint32_t v;
};
typedef struct CFSwappedFloat32 CFSwappedFloat32;
```

Fields

`v`

A 32-bit float value stored with a platform-independent byte order.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`CFByteOrder.h`

CFSwappedFloat64

Structure holding a 64-bit float value in a platform-independent byte order.

```
struct CFSwappedFloat64 {
    uint64_t v;
};
typedef struct CFSwappedFloat64 CFSwappedFloat64;
```

Fields

`v`

A 64-bit float value stored with a platform-independent byte order.

Availability

Available in Mac OS X v10.0 and later.

Declared In
`CFByteOrder.h`

Constants

Byte Order Flags

Flags that identify byte order.

```
enum __CFByteOrder {
    CFByteOrderUnknown,
    CFByteOrderLittleEndian,
    CFByteOrderBigEndian
};
typedef enum __CFByteOrder CFByteOrder;
```

Constants

`CFByteOrderUnknown`

The byte order is unknown.

Available in Mac OS X v10.0 and later.

Declared in `CFByteOrder.h`.

`CFByteOrderLittleEndian`

Multi-byte values are stored with the least-significant bytes stored first. Pentium CPUs are little endian.

Available in Mac OS X v10.0 and later.

Declared in `CFByteOrder.h`.

`CFByteOrderBigEndian`

Multi-byte values are stored with the most-significant bytes stored first. PowerPC CPUs are big endian.

Available in Mac OS X v10.0 and later.

Declared in `CFByteOrder.h`.

Document Revision History

This table describes the changes to *Byte-Order Utilities Reference*.

Date	Notes
2006-05-23	Corrected typographical errors.
2005-08-11	Corrected various typographical errors.
2003-01-01	First version of this document.

Index