
Core Endian Reference

[Carbon](#) > [Data Management](#)



2007-05-29



Apple Inc.
© 2005, 2007 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

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

.Mac is a registered service mark of Apple Inc.

Apple, the Apple logo, Carbon, Mac, Mac OS, Macintosh, QuickTime, and Xcode are trademarks of Apple Inc., registered in the United States and other countries.

PowerPC and the PowerPC logo are trademarks of International Business Machines Corporation, used under license therefrom.

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

Core Endian Reference 5

Overview	5
Functions by Task	5
Working With Flippers	5
Changing the Endian Format	6
Converting from Big-Endian to Native Format	6
Converting from Native Format to Big-Endian Format	6
Converting from Little-Endian Format to Native Format	7
Converting from Native Format to Little-Endian Format	7
Converting from Big-Endian to Little-Endian Format	7
Converting From Little-Endian to Big-Endian Format	8
Functions	8
CoreEndianFlipData	8
CoreEndianGetFlipper	9
CoreEndianInstallFlipper	10
Endian16_Swap	11
Endian32_Swap	11
Endian64_Swap	11
EndianS16_BtoL	12
EndianS16_BtoN	12
EndianS16_LtoB	13
EndianS16_LtoN	13
EndianS16_NtoB	13
EndianS16_NtoL	14
EndianS32_BtoL	14
EndianS32_BtoN	15
EndianS32_LtoB	15
EndianS32_LtoN	16
EndianS32_NtoB	16
EndianS32_NtoL	16
EndianS64_BtoL	17
EndianS64_BtoN	17
EndianS64_LtoB	18
EndianS64_LtoN	18
EndianS64_NtoB	18
EndianS64_NtoL	19
EndianU16_BtoL	19
EndianU16_BtoN	20
EndianU16_LtoB	20
EndianU16_LtoN	20
EndianU16_NtoB	21

- EndianU16_NtoL 21
- EndianU32_BtoL 22
- EndianU32_BtoN 22
- EndianU32_LtoB 22
- EndianU32_LtoN 23
- EndianU32_NtoB 23
- EndianU32_NtoL 24
- EndianU64_BtoL 24
- EndianU64_BtoN 24
- EndianU64_LtoB 25
- EndianU64_LtoN 25
- EndianU64_NtoB 26
- EndianU64_NtoL 26
- Callbacks 27
 - CoreEndianFlipProc 27
- Data Types 28
 - BigEndianLong 28
 - BigEndianUnsignedLong 29
 - BigEndianShort 29
 - BigEndianUnsignedShort 29
 - BigEndianFixed 30
 - BigEndianUnsignedFixed 30
 - BigEndianOSType 31
- Constants 31
 - Domain Types 31

Document Revision History 33

Index 35

Core Endian Reference

Framework:	CoreServices/CoreServices.h
Declared in	Endian.h

Overview

Core Endian Reference provides routines for converting data between big endian and little endian format. These routines are useful for developers who write code that must compile for multiple architectures, including:

- Macintosh developers who want to produce a universal binary.
- QuickTime developers who want their code to run in Windows as well as in Mac OS X.

Both QuickTime and Macintosh developers can use this API to handle reading or writing data to a file or network packet.

Macintosh developers can use this API to create and install callbacks that are invoked by Mac OS X automatically when your application:

- reads and writes custom resource data
- sends or receives custom Apple events
- reads and writes custom pasteboard data

The functions in this API are designed to do nothing when the target runtime is already in the desired format.

For more information see:

- [Universal Binary Programming Guidelines](#)
- [QuickTime API Reference](#)

Functions by Task

Working With Flippers

[CoreEndianInstallFlipper](#) (page 10)

Installs a flipper callback for the specified data type.

[CoreEndianGetFlipper](#) (page 9)

Obtains the flipper callback that is installed for the specified data type.

[CoreEndianFlipData](#) (page 8)

Calls the flipper callback associated with the specified data type.

Changing the Endian Format

[Endian16_Swap](#) (page 11)

Changes the endian format of an unsigned 16-bit integer.

[Endian32_Swap](#) (page 11)

Changes the endian format of an unsigned 32-bit integer.

[Endian64_Swap](#) (page 11)

Changes the endian format of an unsigned 64-bit integer.

Converting from Big-Endian to Native Format

[EndianS16_BtoN](#) (page 12)

Converts a signed 16-bit big-endian value to the equivalent value in the computer's native format.

[EndianS32_BtoN](#) (page 15)

Converts a signed 32-bit big-endian value to the equivalent value in the computer's native format.

[EndianS64_BtoN](#) (page 17)

Converts a signed 64-bit big-endian value to the equivalent value in the computer's native format.

[EndianU16_BtoN](#) (page 20)

Converts an unsigned 16-bit big-endian value to the equivalent value in the computer's native format.

[EndianU32_BtoN](#) (page 22)

Converts an unsigned 32-bit big-endian value to the equivalent value in the computer's native format.

[EndianU64_BtoN](#) (page 24)

Converts an unsigned 64-bit big-endian value to the equivalent value in the computer's native format.

Converting from Native Format to Big-Endian Format

[EndianS16_NtoB](#) (page 13)

Converts a signed 16-bit value in the computer's native format to the equivalent big-endian value.

[EndianS32_NtoB](#) (page 16)

Converts a signed 32-bit value in the computer's native format to the equivalent big-endian value.

[EndianS64_NtoB](#) (page 18)

Converts a signed 64-bit value in the computer's native format to the equivalent big-endian value.

[EndianU16_NtoB](#) (page 21)

Converts an unsigned 16-bit value in the computer's native format to the equivalent big-endian value.

[EndianU32_NtoB](#) (page 23)

Converts an unsigned 32-bit value in the computer's native format to the equivalent big-endian value.

[EndianU64_NtoB](#) (page 26)

Converts an unsigned 64-bit value in the computer's native format to the equivalent big-endian value.

Converting from Little-Endian Format to Native Format

[EndianS16_LtoN](#) (page 13)

Converts a signed 16-bit little-endian value to the equivalent value in the computer's native format.

[EndianS32_LtoN](#) (page 16)

Converts a signed 32-bit little-endian value to the equivalent value in the computer's native format.

[EndianS64_LtoN](#) (page 18)

Converts a signed 64-bit little-endian value to the equivalent value in the computer's native format.

[EndianU16_LtoN](#) (page 20)

Converts an unsigned 16-bit little-endian value to the equivalent value in the computer's native format.

[EndianU32_LtoN](#) (page 23)

Converts an unsigned 32-bit little-endian value to the equivalent value in the computer's native format.

[EndianU64_LtoN](#) (page 25)

Converts an unsigned 64-bit little-endian value to the equivalent value in the computer's native format.

Converting from Native Format to Little-Endian Format

[EndianS16_NtoL](#) (page 14)

Converts a signed 16-bit value in the computer's native format to the equivalent little-endian value.

[EndianS32_NtoL](#) (page 16)

Converts a signed 32-bit value in the computer's native format to the equivalent little-endian value.

[EndianS64_NtoL](#) (page 19)

Converts a signed 64-bit value in the computer's native format to the equivalent little-endian value.

[EndianU16_NtoL](#) (page 21)

Converts an unsigned 16-bit value in the computer's native format to the equivalent little-endian value.

[EndianU32_NtoL](#) (page 24)

Converts an unsigned 32-bit value in the computer's native format to the equivalent little-endian value.

[EndianU64_NtoL](#) (page 26)

Converts an unsigned 64-bit value in the computer's native format to the equivalent little-endian value.

Converting from Big-Endian to Little-Endian Format

[EndianS16_BtoL](#) (page 12)

Converts a signed 16-bit big-endian value to the equivalent little-endian value.

[EndianS32_BtoL](#) (page 14)

Converts a signed 32-bit big-endian value to the equivalent little-endian value.

[EndianS64_BtoL](#) (page 17)

Converts a signed 64-bit big-endian value to the equivalent little-endian value.

[EndianU16_BtoL](#) (page 19)

Converts an unsigned 16-bit big-endian value to the equivalent little-endian value.

[EndianU32_BtoL](#) (page 22)

Converts an unsigned 32-bit big-endian value to the equivalent little-endian value.

[EndianU64_BtoL](#) (page 24)

Converts an unsigned 64-bit big-endian value to the equivalent little-endian value.

Converting From Little-Endian to Big-Endian Format

[EndianS16_LtoB](#) (page 13)

Converts a signed 16-bit little-endian value to the equivalent big-endian value.

[EndianS32_LtoB](#) (page 15)

Converts a signed 32-bit little-endian value to the equivalent big-endian value.

[EndianS64_LtoB](#) (page 18)

Converts a signed 64-bit little-endian value to the equivalent big-endian value.

[EndianU16_LtoB](#) (page 20)

Converts an unsigned 16-bit little-endian value to the equivalent big-endian value.

[EndianU32_LtoB](#) (page 22)

Converts an unsigned 32-bit little-endian value to the equivalent big-endian value.

[EndianU64_LtoB](#) (page 25)

Converts an unsigned 64-bit little-endian value to the equivalent big-endian value.

Functions

CoreEndianFlipData

Calls the flipper callback associated with the specified data type.

```
OSStatus CoreEndianFlipData (
    OSType dataDomain,
    OSType dataType,
    SInt16 id,
    void *data,
    ByteCount dataLen,
    Boolean currentlyNative
);
```

Parameters

dataDomain

An `OSType` value that specifies the domain of the flipper callback you want to invoke. Pass [kCoreEndianResourceManagerDomain](#) (page 32) if your callback applies to resource data. Pass [kCoreEndianAppleEventManagerDomain](#) (page 32) if your callback applies to Apple event data. See “[Domain Types](#)” (page 31) for more information.

dataType

An `OSType` value that specifies the type of data that needs to be byte-swapped. This is the four character code of the resource type or Apple event. This never needs to be byte-swapped even though GDB and Xcode display the resource in byte-swapped order.

id

The resource ID of the data type. The Resource Manager byte-swaps this for you so you can compare the resource ID against constants in your code. If the data is not a resource, pass 0.

data

A pointer to the first byte of the data to be byte swapped.

dataLen

The length of the data (in bytes) to be byte swapped.

currentlyNative

A Boolean value that indicates the direction to byte swap. Pass `true` when the data specified by the `data` parameter uses the byte ordering of the currently executing code. On a PowerPC system, `true` specifies that the data is in big-endian format. On an x86 system, `true` specifies that the data is in little-endian format.

Return Value

A result code. Returns `noErr` if the data is byte swapped and `handlerNotFound` if the data is not byte swapped. Note that data is only byte swapped if it needs to be byte swapped.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Endian.h`

CoreEndianGetFlipper

Obtains the flipper callback that is installed for the specified data type.

```
OSStatus CoreEndianGetFlipper (
    OSType dataDomain,
    OSType dataType,
    CoreEndianFlipProc *proc,
    void **refcon
);
```

Parameters*dataDomain*

An `OSType` value that specifies the domain of the flipper callback you want to obtain. Pass `kCoreEndianResourceManagerDomain` (page 32) to obtain a callback that applies to resource data. Pass `kCoreEndianAppleEventManagerDomain` (page 32) to obtain a callback that applies to Apple event data. See “Domain Types” (page 31) for more information.

dataType

An `OSType` value that specifies the type of data associated with the flipper callback you want to obtain. This is the four character code of the resource type or Apple event. This never needs to be byte-swapped even though GDB and Xcode display the resource in byte-swapped order.

proc

On output, points to the flipper callback that is installed for the data type specified by the `dataType` parameter.

refCon

On output, points to a 32-bit value that references callback-specific data.

Return Value

A result code. Returns `noErr` if the flipper callback is found.

Discussion

You can call the function `CoreEndianGetFlipper` to determine whether a flipper for a given data type is available.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Endian.h`

CoreEndianInstallFlipper

Installs a flipper callback for the specified data type.

```
OSStatus CoreEndianInstallFlipper (
    OSType dataDomain,
    OSType dataType,
    CoreEndianFlipProc proc,
    void *refcon
);
```

Parameters

dataDomain

An `OSType` value that specifies the domain to which the flipper callback applies. Pass `kCoreEndianResourceManagerDomain` (page 32) if your callback applies to resource data. Pass `kCoreEndianAppleEventManagerDomain` (page 32) if your callback applies to Apple event data. See “Domain Types” (page 31) for more information.

dataType

An `OSType` value that specifies the type of data for which you want your flipper callback installed. This is the four character code of the resource type or Apple event.

proc

A pointer to your flipper callback. The flipper callback is installed into a per-process table that is searched before the system table.

refCon

A 32-bit value containing or referring to data needed by the callback.

Return Value

A result code. Returns `noErr` if your flipper callback is installed.

Discussion

You should install the callback by calling the function `CoreEndianInstallFlipper` when your application calls its initialization routine or when you open your resource file.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`Endian.h`

Endian16_Swap

Changes the endian format of an unsigned 16-bit integer.

```
UInt16 Endian16_Swap (  
    UInt16 value  
);
```

Parameters

value

An unsigned 16-bit integer input.

Return Value

The unsigned 16-bit integer result.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 5 and later for Windows.

Declared In

Endian.h

Endian32_Swap

Changes the endian format of an unsigned 32-bit integer.

```
UInt32 Endian32_Swap (  
    UInt32 value  
);
```

Parameters

value

An unsigned 32-bit integer input.

Return Value

The unsigned 32-bit integer result.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 5 and later for Windows.

Declared In

Endian.h

Endian64_Swap

Changes the endian format of an unsigned 64-bit integer.

```
static UInt64 Endian64_Swap (  
    UInt64 value  
);
```

Parameters

value

An unsigned 64-bit integer input.

Return Value

The unsigned 64-bit integer result.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 5 and later for Windows.

Declared In

Endian.h

EndianS16_BtoL

Converts a signed 16-bit big-endian value to the equivalent little-endian value.

```
SInt16 EndianS16_BtoL (  
    SInt16  value  
);
```

Parameters

value

A signed 16-bit big-endian value.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS16_BtoN

Converts a signed 16-bit big-endian value to the equivalent value in the computer's native format.

```
SInt16 EndianS16_BtoN (  
    SInt16  value  
);
```

Parameters

value

A signed 16-bit big-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Related Sample Code

QTCarbonShell

Declared In

Endian.h

EndianS16_LtoB

Converts a signed 16-bit little-endian value to the equivalent big-endian value.

```
SInt16 EndianS16_LtoB (  
    SInt16    value  
);
```

Parameters

value

A signed 16-bit little-endian value.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Declared In

Endian.h

EndianS16_LtoN

Converts a signed 16-bit little-endian value to the equivalent value in the computer's native format.

```
SInt16 EndianS16_LtoN (  
    SInt16    value  
);
```

Parameters

value

A signed 16-bit little-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS16_NtoB

Converts a signed 16-bit value in the computer's native format to the equivalent big-endian value.

```
SInt16 EndianS16_NtoB (  
    SInt16    value  
);
```

Parameters

value

A signed 16-bit value in the computer's native format.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS16_NtoL

Converts a signed 16-bit value in the computer's native format to the equivalent little-endian value.

```
SInt16 EndianS16_NtoL (  
    SInt16    value  
);
```

Parameters

value

A signed 16-bit value in the computer's native format.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS32_BtoL

Converts a signed 32-bit big-endian value to the equivalent little-endian value.

```
SInt32 EndianS32_BtoL (  
    SInt32    value  
);
```

Parameters

value

A signed 32-bit big-endian value.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS32_BtoN

Converts a signed 32-bit big-endian value to the equivalent value in the computer's native format.

```
SInt32 EndianS32_BtoN (  
    SInt32 value  
);
```

Parameters

value

A signed 32-bit big-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS32_LtoB

Converts a signed 32-bit little-endian value to the equivalent big-endian value.

```
SInt32 EndianS32_LtoB (  
    SInt32 value  
);
```

Parameters

value

A signed 32-bit little-endian value.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS32_LtoN

Converts a signed 32-bit little-endian value to the equivalent value in the computer's native format.

```
SInt32 EndianS32_LtoN (
    SInt32    value
);
```

Parameters

value

A signed 32-bit little-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS32_NtoB

Converts a signed 32-bit value in the computer's native format to the equivalent big-endian value.

```
SInt32 EndianS32_NtoB (
    SInt32    value
);
```

Parameters

value

A signed 32-bit value in the computer's native format.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS32_NtoL

Converts a signed 32-bit value in the computer's native format to the equivalent little-endian value.

```
SInt32 EndianS32_NtoL (
    SInt32    value
);
```

Parameters

value

A signed 32-bit value in the computer's native format.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS64_BtoL

Converts a signed 64-bit big-endian value to the equivalent little-endian value.

```
SInt64 EndianS64_BtoL (  
    SInt64  value  
);
```

Parameters

value

A signed 64-bit big-endian value.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS64_BtoN

Converts a signed 64-bit big-endian value to the equivalent value in the computer's native format.

```
SInt64 EndianS64_BtoN (  
    SInt64  value  
);
```

Parameters

value

A signed 64-bit big-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS64_LtoB

Converts a signed 64-bit little-endian value to the equivalent big-endian value.

```
SInt64 EndianS64_LtoB (
    SInt64    value
);
```

Parameters

value

A signed 64-bit little-endian value.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS64_LtoN

Converts a signed 64-bit little-endian value to the equivalent value in the computer's native format.

```
SInt64 EndianS64_LtoN (
    SInt64    value
);
```

Parameters

value

A signed 64-bit little-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS64_NtoB

Converts a signed 64-bit value in the computer's native format to the equivalent big-endian value.

```
SInt64 EndianS64_NtoB (
    SInt64    value
);
```

Parameters

value

A signed 64-bit value in the computer's native format.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianS64_NtoL

Converts a signed 64-bit value in the computer's native format to the equivalent little-endian value.

```
SInt64 EndianS64_NtoL (  
    SInt64 value  
);
```

Parameters

value

A signed 64-bit value in the computer's native format.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU16_BtoL

Converts an unsigned 16-bit big-endian value to the equivalent little-endian value.

```
UInt16 EndianU16_BtoL (  
    UInt16 value  
);
```

Parameters

value

An unsigned 16-bit big-endian value.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU16_BtoN

Converts an unsigned 16-bit big-endian value to the equivalent value in the computer's native format.

```
UInt16 EndianU16_BtoN (
    UInt16    value
);
```

Parameters

value

An unsigned 16-bit big-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU16_LtoB

Converts an unsigned 16-bit little-endian value to the equivalent big-endian value.

```
UInt16 EndianU16_LtoB (
    UInt16    value
);
```

Parameters

value

An unsigned 16-bit little-endian value.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU16_LtoN

Converts an unsigned 16-bit little-endian value to the equivalent value in the computer's native format.

```
UInt16 EndianU16_LtoN (
    UInt16    value
);
```

Parameters

value

An unsigned 16-bit little-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU16_NtoB

Converts an unsigned 16-bit value in the computer's native format to the equivalent big-endian value.

```
UInt16 EndianU16_NtoB (  
    UInt16    value  
);
```

Parameters

value

An unsigned 16-bit value in the computer's native format.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU16_NtoL

Converts an unsigned 16-bit value in the computer's native format to the equivalent little-endian value.

```
UInt16 EndianU16_NtoL (  
    UInt16    value  
);
```

Parameters

value

An unsigned 16-bit value in the computer's native format.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU32_BtoL

Converts an unsigned 32-bit big-endian value to the equivalent little-endian value.

```
UInt32 EndianU32_BtoL (
    UInt32    value
);
```

Parameters

value

An unsigned 32-bit big-endian value.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU32_BtoN

Converts an unsigned 32-bit big-endian value to the equivalent value in the computer's native format.

```
UInt32 EndianU32_BtoN (
    UInt32    value
);
```

Parameters

value

An unsigned 32-bit big-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU32_LtoB

Converts an unsigned 32-bit little-endian value to the equivalent big-endian value.

```
UInt32 EndianU32_LtoB (
    UInt32    value
);
```

Parameters

value

An unsigned 32-bit little-endian value.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU32_LtoN

Converts an unsigned 32-bit little-endian value to the equivalent value in the computer's native format.

```
UInt32 EndianU32_LtoN (  
    UInt32    value  
);
```

Parameters

value

An unsigned 32-bit little-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU32_NtoB

Converts an unsigned 32-bit value in the computer's native format to the equivalent big-endian value.

```
UInt32 EndianU32_NtoB (  
    UInt32    value  
);
```

Parameters

value

An unsigned 32-bit value in the computer's native format.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Related Sample Code

QTMetaData

Declared In

Endian.h

EndianU32_NtoL

Converts an unsigned 32-bit value in the computer's native format to the equivalent little-endian value.

```
UInt32 EndianU32_NtoL (
    UInt32    value
);
```

Parameters*value*

An unsigned 32-bit value in the computer's native format.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU64_BtoL

Converts an unsigned 64-bit big-endian value to the equivalent little-endian value.

```
UInt64 EndianU64_BtoL (
    UInt64    value
);
```

Parameters*value*

An unsigned 64-bit big-endian value.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU64_BtoN

Converts an unsigned 64-bit big-endian value to the equivalent value in the computer's native format.


```
UInt64 EndianU64_BtoN (  
    UInt64    value  
);
```

Parameters

value

An unsigned 64-bit big-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU64_LtoB

Converts an unsigned 64-bit little-endian value to the equivalent big-endian value.

```
UInt64 EndianU64_LtoB (  
    UInt64    value  
);
```

Parameters

value

An unsigned 64-bit little-endian value.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU64_LtoN

Converts an unsigned 64-bit little-endian value to the equivalent value in the computer's native format.

```
UInt64 EndianU64_LtoN (  
    UInt64    value  
);
```

Parameters

value

An unsigned 64-bit little-endian value.

Return Value

The equivalent value in the computer's native format.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU64_NtoB

Converts an unsigned 64-bit value in the computer's native format to the equivalent big-endian value.

```
UInt64 EndianU64_NtoB (  
    UInt64    value  
);
```

Parameters

value

An unsigned 64-bit value in the computer's native format.

Return Value

The equivalent big-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

EndianU64_NtoL

Converts an unsigned 64-bit value in the computer's native format to the equivalent little-endian value.

```
UInt64 EndianU64_NtoL (  
    UInt64    value  
);
```

Parameters

value

An unsigned 64-bit value in the computer's native format.

Return Value

The equivalent little-endian value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

Callbacks

CoreEndianFlipProc

Defines a pointer to a callback function that byte-swaps your custom data.

```
typedef CALLBACK_API (OSStatus, CoreEndianFlipProc)
    (OSType dataDomain,
     OSType dataType,
     short id,
     void *dataPtr,
     UInt32 dataSize,
     Boolean currentlyNative,
     void *refcon
    );
```

You would declare your flipper callback function as follows if you were to name it `MyCoreEndianFlipProc`:

```
OSStatus MyCoreEndianFlipProc (
    OSType dataDomain,
    OSType dataType,
    short id,
    void *dataPtr,
    UInt32 dataSize,
    Boolean currentlyNative,
    void *refcon
);
```

Parameters

dataDomain

An `OSType` value that specifies the domain to which the flipper callback applies. The value `kCoreEndianResourceManagerDomain` signifies the domain is resource data. The value `kCoreEndianAppleEventManagerDomain` signifies the domain is Apple event data. See “[Domain Types](#)” (page 31) for more information on the values that can be passed to your callback.

dataType

The type of data to be byte swapped by the callback. This is the four character code of the resource type or Apple event.

id

The resource id of the data type. The value 0 signifies the data is not a resource.

dataPtr

On input, points to the data to be flipped. On output, points to the byte-swapped data.

dataSize

The size of the data pointed to the by the `dataPtr` parameter.

currentlyNative

A Boolean value that indicates the direction to byte swap. The value `true` specifies the data pointed to by the `dataPtr` parameter uses the byte ordering of the currently executing code. On a PowerPC system, `true` specifies that the data is in big-endian format. On an x86 system, `true` specifies that the data is in little-endian format.

refcon

A 32-bit value that contains or refers to data needed by the callback.

Return Value

A result code that indicates whether the byte swapping is successful. Your callback should return `noErr` if the resource is byte swapped without error, `handlerNotFound` if you chose not to byte swap the data, and the appropriate result code to indicate an error condition if the data is bad. The result code you return is propagated through the appropriate manager (Resource Manager (`ResError`) or Apple Event Manager) to the caller.

Discussion

You should write each flipper callback so it traverses the data structure that contains the data and performs the following tasks:

- Byte swaps all Resource Manager counts and lengths so that array indexes are associated with the appropriate value
- Byte swaps all integers and longs so that when you read them into variables of a compatible type the values can be operated on correctly (such as numerical, offset, and shift operations)

A flipper callback must be bidirectional because it can be called by the Resource Manager or Apple Event Manager when you read data as well as when you write data. The system ensures that your flipper callback is invoked at the appropriate times.

Your flipper callback is not invoked on a microprocessor that uses big-endian byte ordering. It is called with `currentlyNative` set to `false` when data is read (or received) and `true` when the data is set to be written (or sent).

Availability

Available in Mac OS X 10.3 and later.

Declared In

`Endian.h`

Data Types

BigEndianLong

Protects a big-endian long value from being changed by little-endian code.

```
// Little-endian host
struct BigEndianLong {
    long bigEndianValue;
};
typedef struct BigEndianLong BigEndianLong;
// Big-endian host
typedef long BigEndianLong;
```

Fields

`bigEndianValue`

A long value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

BigEndianUnsignedLong

Protects a big-endian unsigned long value from being changed by little-endian code.

```
// Little-endian host
struct BigEndianUnsignedLong {
    unsigned long    bigEndianValue;
};
typedef struct BigEndianUnsignedLong BigEndianUnsignedLong;
// Big-endian host
typedef unsigned long    BigEndianUnsignedLong;
```

Fields

bigEndianValue

An unsigned long value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

BigEndianShort

Protects a big-endian short value from being changed by little-endian code.

```
// Little-endian host
struct BigEndianShort {
    short    bigEndianValue;
};
typedef struct BigEndianShort    BigEndianShort;
// Big-endian host
typedef short    BigEndianShort;
```

Fields

bigEndianValue

A short value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

BigEndianUnsignedShort

Protects a big-endian unsigned short value from being changed by little-endian code.

```
// Little-endian host
struct BigEndianUnsignedShort {
    unsigned short    bigEndianValue;
};
typedef struct BigEndianUnsignedShort    BigEndianUnsignedShort;
// Big-endian host
typedef unsigned short    BigEndianUnsignedShort;
```

Fields

bigEndianValue

An unsigned short value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

BigEndianFixed

Protects a big-endian Fixed value from being changed by little-endian code.

```
// Little-endian host
struct BigEndianFixed {
    Fixed    bigEndianValue;
};
typedef struct BigEndianFixed    BigEndianFixed;
// Big-endian host
typedef Fixed    BigEndianFixed;
```

Fields

bigEndianValue

A fixed value.

Availability

Available in Mac OS X v10.0 and later.

Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

BigEndianUnsignedFixed

Protects a big-endian unsigned Fixed value from being changed by little-endian code.

```
// Little-endian host
struct BigEndianUnsignedFixed {
    UnsignedFixed    bigEndianValue;
};
typedef struct BigEndianUnsignedFixed  BigEndianUnsignedFixed;
// Big-endian host
typedef UnsignedFixed BigEndianUnsignedFixed;
```

Fields

bigEndianValue
An unsigned fixed value.

Availability

Available in Mac OS X v10.0 and later.
Available in QuickTime 4 and later for Windows.

Declared In

Endian.h

BigEndianOSType

Protects a big-endian OSType value from being changed by little-endian code.

```
// Little-endian host
struct BigEndianOSType {
    OSType    bigEndianValue;
};
typedef struct BigEndianOSType  BigEndianOSType;
// Big-endian host
typedef OSType BigEndianOSType;
```

Fields

bigEndianValue
An OSType value.

Availability

Available in Mac OS X v10.0 and later.
Available in QuickTime 4 and later for Windows.

Constants

Domain Types

Specify the domain to which a flipper callback should be applied.

```
enum {  
    kCoreEndianResourceManagerDomain = 'rsrc',  
    kCoreEndianAppleEventManagerDomain = 'aevt'  
};
```

Constants

`kCoreEndianResourceManagerDomain`

Specifies that the domain is limited to the resources for a specific application.

Available in Mac OS X v10.4 and later.

Declared in `Endian.h`.

`kCoreEndianAppleEventManagerDomain`

Specifies that the domain is limited to Apple events.

Available in Mac OS X v10.4 and later.

Declared in `Endian.h`.

Discussion

The data types have specific meanings within their domain, although some data types can be registered with the same callback in several domains.

Availability

Available in Mac OS X 10.3 and later.

Document Revision History

This table describes the changes to *Core Endian Reference*.

Date	Notes
2007-05-29	Changed framework entry.
2005-08-11	Updated to consolidate all functions and data types provided in Endian.h into one reference document.
2005-07-07	New document that describes the C API for creating and installing callbacks to byte swap custom data.

REVISION HISTORY

Document Revision History

Index

B

BigEndianFixed **structure** [30](#)
BigEndianLong **structure** [28](#)
BigEndianOSType **structure** [31](#)
BigEndianShort **structure** [29](#)
BigEndianUnsignedFixed **structure** [30](#)
BigEndianUnsignedLong **structure** [29](#)
BigEndianUnsignedShort **structure** [29](#)

C

CoreEndianFlipData **function** [8](#)
CoreEndianFlipProc **callback** [27](#)
CoreEndianGetFlipper **function** [9](#)
CoreEndianInstallFlipper **function** [10](#)

D

Domain Types [31](#)

E

Endian16_Swap **macro** [11](#)
Endian32_Swap **macro** [11](#)
Endian64_Swap **macro** [11](#)
EndianS16_BtoL **macro** [12](#)
EndianS16_BtoN **macro** [12](#)
EndianS16_LtoB **macro** [13](#)
EndianS16_LtoN **macro** [13](#)
EndianS16_NtoB **macro** [13](#)
EndianS16_NtoL **macro** [14](#)
EndianS32_BtoL **macro** [14](#)
EndianS32_BtoN **macro** [15](#)
EndianS32_LtoB **macro** [15](#)
EndianS32_LtoN **macro** [16](#)

EndianS32_NtoB **macro** [16](#)
EndianS32_NtoL **macro** [16](#)
EndianS64_BtoL **macro** [17](#)
EndianS64_BtoN **macro** [17](#)
EndianS64_LtoB **macro** [18](#)
EndianS64_LtoN **macro** [18](#)
EndianS64_NtoB **macro** [18](#)
EndianS64_NtoL **macro** [19](#)
EndianU16_BtoL **macro** [19](#)
EndianU16_BtoN **macro** [20](#)
EndianU16_LtoB **macro** [20](#)
EndianU16_LtoN **macro** [20](#)
EndianU16_NtoB **macro** [21](#)
EndianU16_NtoL **macro** [21](#)
EndianU32_BtoL **macro** [22](#)
EndianU32_BtoN **macro** [22](#)
EndianU32_LtoB **macro** [22](#)
EndianU32_LtoN **macro** [23](#)
EndianU32_NtoB **macro** [23](#)
EndianU32_NtoL **macro** [24](#)
EndianU64_BtoL **macro** [24](#)
EndianU64_BtoN **macro** [24](#)
EndianU64_LtoB **macro** [25](#)
EndianU64_LtoN **macro** [25](#)
EndianU64_NtoB **macro** [26](#)
EndianU64_NtoL **macro** [26](#)

K

kCoreEndianAppleEventManagerDomain **constant** [32](#)
kCoreEndianResourceManagerDomain **constant** [32](#)