

---

# Core Animation Function Reference

[Graphics & Imaging > Quartz](#)



2007-07-24



Apple Inc.  
© 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

Apple, the Apple logo, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

iPhone is a trademark of Apple Inc.

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 Animation Function Reference 5

---

Overview	5
Functions by Task	5
Timing Functions	5
Transform Functions	5
Functions	6
CACurrentMediaTime	6
CATransform3DConcat	6
CATransform3DEqualToTransform	6
CATransform3DGetAffineTransform	7
CATransform3DInvert	7
CATransform3DIsAffine	7
CATransform3DIsIdentity	7
CATransform3DMakeAffineTransform	8
CATransform3DMakeRotation	8
CATransform3DMakeScale	8
CATransform3DMakeTranslation	9
CATransform3DRotate	9
CATransform3DScale	9
CATransform3DTranslate	9

---

## Document Revision History 11

---

## Index 13

---



# Core Animation Function Reference

---

<b>Framework:</b>	QuartzCore/QuartzCore.h
<b>Declared in</b>	CABase.h CATransform3D.h

## Overview

## Functions by Task

### Timing Functions

[CACurrentMediaTime](#) (page 6)  
Returns the current absolute time, in seconds.

### Transform Functions

[CATransform3DIsIdentity](#) (page 7)  
Returns a Boolean value that indicates whether the transform is the identity transform.

[CATransform3DEqualToTransform](#) (page 6)  
Returns a Boolean value that indicates whether the two transforms are exactly equal.

[CATransform3DMakeTranslation](#) (page 9)  
Returns a transform that translates by '(tx, ty, tz)'.  $t' = [1\ 0\ 0\ 0; 0\ 1\ 0\ 0; 0\ 0\ 1\ 0; tx\ ty\ tz\ 1]$ .

[CATransform3DMakeScale](#) (page 8)  
Returns a transform that scales by '(sx, sy, sz)'.  $t' = [sx\ 0\ 0\ 0; 0\ sy\ 0\ 0; 0\ 0\ sz\ 0; 0\ 0\ 0\ 1]$ .

[CATransform3DMakeRotation](#) (page 8)  
Returns a transform that rotates by 'angle' radians about the vector '(x, y, z)'. If the vector has length zero the identity transform is returned.

[CATransform3DTranslate](#) (page 9)  
Translate 't' by '(tx, ty, tz)' and return the result:  $*t' = \text{translate}(tx, ty, tz) * t$ .

[CATransform3DScale](#) (page 9)  
Scale 't' by '(sx, sy, sz)' and return the result:  $*t' = \text{scale}(sx, sy, sz) * t$ .

[CATransform3DRotate](#) (page 9)  
Rotate 't' by 'angle' radians about the vector '(x, y, z)' and return the result. If the vector has zero length the behavior is undefined:  $t' = \text{rotation}(\text{angle}, x, y, z) * t$ .

[CATransform3DConcat](#) (page 6)

Concatenate 'b' to 'a' and return the result:  $t' = a * b$ .

[CATransform3DInvert](#) (page 7)

Invert 't' and return the result. Returns the original matrix if 't' has no inverse.

[CATransform3DMakeAffineTransform](#) (page 8)

Return a transform with the same effect as affine transform 'm'.

[CATransform3DIsAffine](#) (page 7)

Returns true if 't' can be exactly represented by an affine transform.

[CATransform3DGetAffineTransform](#) (page 7)

Returns the affine transform represented by 't'. If 't' can not be exactly represented as an affine transform the returned value is undefined.

## Functions

### **CACurrentMediaTime**

Returns the current absolute time, in seconds.

```
CTimeInterval CACurrentMediaTime (void);
```

#### **Return Value**

A `CFTimeInterval` derived by calling `mach_absolute_time()` and converting the result to seconds.

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`CABase.h`

### **CATransform3DConcat**

Concatenate 'b' to 'a' and return the result:  $t' = a * b$ .

```
CATransform3D CATransform3DConcat (CATransform3D a, CATransform3D b);
```

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`CATransform3D.h`

### **CATransform3DEqualToTransform**

Returns a Boolean value that indicates whether the two transforms are exactly equal.

```
bool CATransform3DEqualToTransform (CATransform3D a, CATransform3D b);
```

**Return Value**

YES if *a* and *b* are exactly equal, otherwise NO.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CATransform3D.h

### CATransform3DGetAffineTransform

Returns the affine transform represented by 't'. If 't' can not be exactly represented as an affine transform the returned value is undefined.

```
CGAffineTransform CATransform3DGetAffineTransform (CATransform3D t);
```

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CATransform3D.h

### CATransform3DInvert

Invert 't' and return the result. Returns the original matrix if 't' has no inverse.

```
CATransform3D CATransform3DInvert (CATransform3D t);
```

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CATransform3D.h

### CATransform3DIsAffine

Returns true if 't' can be exactly represented by an affine transform.

```
bool CATransform3DIsAffine (CATransform3D t);
```

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CATransform3D.h

### CATransform3DIsIdentity

Returns a Boolean value that indicates whether the transform is the identity transform.

```
bool CATransform3DIsIdentity (CATransform3D t);
```

**Return Value**

YES if *t* is the identity transform, otherwise NO.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CATransform3D.h

### CATransform3DMakeAffineTransform

Return a transform with the same effect as affine transform 'm'.

```
CATransform3D CATransform3DMakeAffineTransform (CGAffineTransform m)
```

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CATransform3D.h

### CATransform3DMakeRotation

Returns a transform that rotates by 'angle' radians about the vector '(x, y, z)'. If the vector has length zero the identity transform is returned.

```
CATransform3D CATransform3DMakeRotation (CGFloat angle, CGFloat x, CGFloat y, CGFloat z);
```

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CATransform3D.h

### CATransform3DMakeScale

Returns a transform that scales by '(sx, sy, sz)': \* t' = [sx 0 0 0; 0 sy 0 0; 0 0 sz 0; 0 0 0 1].

```
CATransform3D CATransform3DMakeScale (CGFloat sx, CGFloat sy, CGFloat sz);
```

**Availability**

Available in Mac OS X v10.5 and later.

**Related Sample Code**

CALayerEssentials

Core Animation QuickTime Layer

**Declared In**

CATransform3D.h



### **CATransform3DMakeTranslation**

Returns a transform that translates by '(tx, ty, tz)'.  $t' = [1\ 0\ 0\ 0; 0\ 1\ 0\ 0; 0\ 0\ 1\ 0; tx\ ty\ tz\ 1]$ .

`CATransform3D CATransform3DMakeTranslation (CGFloat tx, CGFloat ty, CGFloat tz)`

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`CATransform3D.h`

### **CATransform3DRotate**

Rotate 't' by 'angle' radians about the vector '(x, y, z)' and return the result. If the vector has zero length the behavior is undefined:  $t' = \text{rotation}(\text{angle}, x, y, z) * t$ .

`CATransform3D CATransform3DRotate (CATransform3D t, CGFloat angle, CGFloat x, CGFloat y, CGFloat z)`

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`CATransform3D.h`

### **CATransform3DScale**

Scale 't' by '(sx, sy, sz)' and return the result:  $* t' = \text{scale}(sx, sy, sz) * t$ .

`CATransform3D CATransform3DScale (CATransform3D t, CGFloat sx, CGFloat sy, CGFloat sz)`

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`CATransform3D.h`

### **CATransform3DTranslate**

Translate 't' by '(tx, ty, tz)' and return the result:  $* t' = \text{translate}(tx, ty, tz) * t$ .

`CATransform3D CATransform3DTranslate (CATransform3D t, CGFloat tx, CGFloat ty, CGFloat tz);`

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Declared In**

`CATransform3D.h`



# Document Revision History

---

This table describes the changes to *Core Animation Function Reference*.

Date	Notes
2007-07-24	New document that describes C functions that provide supporting functionality for the classes and protocols in Core Animation.

**REVISION HISTORY**

Document Revision History

# Index

---

## C

---

- CACurrentMediaTime **function 6**
- CATransform3DConcat **function 6**
- CATransform3DEqualToTransform **function 6**
- CATransform3DGetAffineTransform **function 7**
- CATransform3DInvert **function 7**
- CATransform3DIsAffine **function 7**
- CATransform3DIsIdentity **function 7**
- CATransform3DMakeAffineTransform **function 8**
- CATransform3DMakeRotation **function 8**
- CATransform3DMakeScale **function 8**
- CATransform3DMakeTranslation **function 9**
- CATransform3DRotate **function 9**
- CATransform3DScale **function 9**
- CATransform3DTranslate **function 9**