CATransaction Class Reference

Graphics & Imaging > Quartz

2007-07-24

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

Simultaneously published in the United States and Canada.

Contents

CATransaction Class Reference 5

Overview 5
Tasks 5
  Creating and Committing Transactions 5
  Getting and Setting Transaction Properties 6
Class Methods 6
  begin 6
  commit 6
  flush 6
  setValue:forKey: 7
  valueForKey: 7
Constants 8
  Transaction properties 8

Document Revision History 9

Index 11

CATransaction Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/QuartzCore.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	CATransaction.h
Companion guides	Core Animation Programming Guide Core Animation Cookbook

Overview

`CATransaction` is the Core Animation mechanism for batching multiple layer-tree operations into atomic updates to the render tree. Every modification to a layer tree must be part of a transaction Nested transactions are supported.

Core Animation supports two types of transactions: *implicit* transactions and *explicit* transactions. Implicit transactions are created automatically when the layer tree is modified by a thread without an active transaction and are committed automatically when the thread's run-loop next iterates. Explicit transactions occur when the the application sends the CATransaction class a `begin` (page 6) message before modifying the layer tree, and a `commit` (page 6) message afterwards.

In some circumstances (for example, if there is no run-loop, or the run-loop is blocked) it may be necessary to use explicit transactions to get timely render tree updates.

Tasks

Creating and Committing Transactions

`+ begin` (page 6)
Begin a new transaction for the current thread.

`+ commit` (page 6)
Commit all changes made during the current transaction.

`+ flush` (page 6)
Flushes any extant implicit transaction.

Getting and Setting Transaction Properties

`+ valueForKey:` (page 7)
Returns the arbitrary keyed-data specified by the given key.

`+ setValue:forKey:` (page 7)
Sets the arbitrary keyed-data for the specified key.

Class Methods

begin

Begin a new transaction for the current thread.

```
+ (void)begin
```

Discussion
The transaction is nested within the thread's current transaction, if there is one.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`CATransaction.h`

commit

Commit all changes made during the current transaction.

```
+ (void)commit
```

Special Considerations

Raises an exception if no current transaction exists.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`CATransaction.h`

flush

Flushes any extant implicit transaction.

```
+ (void)flush
```

Discussion
Delays the commit until any nested explicit transactions have completed.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`CATransaction.h`

setValue:forKey:

Sets the arbitrary keyed-data for the specified key.

```
+ (void)setValue:(id)anObject
    forKey:(NSString *)key
```

Parameters

anObject

The value for the key identified by *key*.

key

The name of one of the receiver's properties.

Discussion
Nested transactions have nested data scope; setting a key always sets it in the innermost scope.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`CATransaction.h`

valueForKey:

Returns the arbitrary keyed-data specified by the given key.

```
+ (id)valueForKey:(NSString *)key
```

Parameters

key

The name of one of the receiver's properties.

Return Value
The value for the data specified by the key.

Discussion
Nested transactions have nested data scope. Requesting a value for a key first searches the innermost scope, then the enclosing transactions.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`CATransaction.h`

Constants

Transaction properties

These constants define the property keys used by `valueForKey:` (page 7) and `setValue:forKey:` (page 7).

```
NSString * const kCATransactionAnimationDuration;
NSString * const kCATransactionDisableActions;
```

Constants

`kCATransactionAnimationDuration`

Default duration, in seconds, for animations added to layers. The value for this key must be an instance of `NSNumber`.

Available in Mac OS X v10.5 and later.

Declared in `CATransaction.h`.

`kCATransactionDisableActions`

If `YES`, implicit actions for property changes are suppressed. The value for this key must be an instance of `NSNumber`.

Available in Mac OS X v10.5 and later.

Declared in `CATransaction.h`.

Declared In

`CATransaction.h`

Document Revision History

This table describes the changes to *CATransaction Class Reference*.

Date	Notes
2007-07-24	New document that describes the class that provides nested transaction support for Core Animation.

Document Revision History

Index

B

`begin` class method 6

C

`commit` class method 6

F

`flush` class method 6

K

`kCATransactionAnimationDuration` constant 8
`kCATransactionDisableActions` constant 8

S

`setValue:forKey:` class method 7

T

Transaction properties 8

V

`valueForKey:` class method 7