CFMutableBag Reference

Core Foundation

2005-12-06

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

Simultaneously published in the United States and Canada.

Contents

CFMutableBag Reference

Derived From:	CFBag : CFType
Framework:	CoreFoundation/CoreFoundation.h
Companion guide	Collections Programming Topics for Core Foundation
Declared in	CFBag.h

Overview

CFMutableBag manages dynamic bags. The basic interface for managing bags is provided by CFBag. CFMutableBag adds functions to modify the contents of a bag.

You create a mutable bag object using either the `CFBagCreateMutable` (page 6) or `CFBagCreateMutableCopy` (page 7) function.

CFMutableBag provides several functions for adding and removing values from a bag. The `CFBagAddValue` (page 6) function adds a value to a bag and `CFBagRemoveValue` (page 8) removes values from a bag.

Functions by Task

Creating a Mutable Bag

`CFBagCreateMutable` (page 6)
Creates a new empty mutable bag.

`CFBagCreateMutableCopy` (page 7)
Creates a new mutable bag with the values from another bag.

Modifying a Mutable Bag

`CFBagAddValue` (page 6)
Adds a value to a mutable bag.

`CFBagRemoveAllValues` (page 8)
Removes all values from a mutable bag.

`CFBagRemoveValue` (page 8)
Removes a value from a mutable bag.

`CFBagReplaceValue` (page 9)
Replaces a value in a mutable bag.

`CFBagSetValue` (page 9)
Sets a value in a mutable bag.

Functions

CFBagAddValue

Adds a value to a mutable bag.

```
void CFBagAddValue (
   CFMutableBagRef theBag,
   const void *value
);
```

Parameters

theBag
The bag to which *value* is added.

value
A CFType object or a pointer value to add to *theBag* (or the value itself, if it fits into the size of a pointer).

Discussion

The *value* parameter is retained by *theBag* using the retain callback provided when *theBag* was created. If *value* is not of the type expected by the retain callback, the behavior is undefined. If *value* already exists in the collection, it is simply retained again—no memory is allocated for the added value. Use a CFSet object if you don't want duplicate values in your collection.

Availability

Available in CarbonLib v1.0 and later.
Available in Mac OS X v10.0 and later.

Declared In

`CFBag.h`

CFBagCreateMutable

Creates a new empty mutable bag.

```
CFMutableBagRef CFBagCreateMutable (
   CFAllocatorRef allocator,
   CFIndex capacity,
   const CFBagCallBacks *callBacks
);
```

Parameters

allocator
The allocator object to use to allocate memory for the new bag and its storage for values. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

capacity

The maximum number of values that can be contained by the new bag. The bag starts empty and can grow to this number of values (and it can have less). If this parameter is `0`, the bag's maximum capacity is not limited. This value must not be negative.

callBacks

A pointer to a `CFBagCallBacks` structure initialized with the callbacks to use to retain, release, describe, and compare values in the bag. A copy of the contents of the callbacks structure is made, so that a pointer to a structure on the stack can be passed in or can be reused for multiple collection creations. This parameter may be `NULL`, which is treated as if a valid structure of version `0` with all fields `NULL` had been passed in.

If any of the fields are not valid pointers to functions of the correct type, or this parameter is not a valid pointer to a `CFBagCallBacks` structure, the behavior is undefined. If any value put into the collection is not one understood by one of the callback functions, the behavior when that callback function is used is undefined.

If the collection contains only CFType objects, then pass `kCFTypeBagCallBacks` as this parameter to use the default callback functions.

Return Value

A new mutable bag, or `NULL` if there was a problem creating the object. Ownership follows the Create Rule.

Discussion

This function creates an new empty mutable bag to which you can add values using the `CFBagAddValue` (page 6) function. The `capacity` parameter specifies the maximum number of values that the CFBag object can contain. If it is `0`, then there is no limit to the number of values that can be added (aside from constraints such as available memory).

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Declared In

`CFBag.h`

CFBagCreateMutableCopy

Creates a new mutable bag with the values from another bag.

```
CFMutableBagRef CFBagCreateMutableCopy (
   CFAllocatorRef allocator,
   CFIndex capacity,
   CFBagRef theBag
);
```

Parameters

allocator

The allocator to use to allocate memory for the new bag and its storage for values. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

capacity

The maximum number of values that can be contained by the new bag. The bag starts with the same count as *theBag*, and can grow to this number of values (and it can have less). If this value is `0`, the bag's maximum capacity is not limited. This value must be greater than or equal to the count of *theBag*, and must not be negative.

theBag

The bag to copy. The pointer values from *theBag* are copied into the new bag. However, the values are also retained by the new bag. The count of the new bag is the same as the count of *theBag*. The new bag uses the same callbacks as *theBag*.

Return Value
A new mutable bag that contains the same values as *theBag*. Ownership follows the Create Rule.

Availability
Available in CarbonLib v1.0 and later.
Available in Mac OS X v10.0 and later.

Declared In
CFBag.h

CFBagRemoveAllValues

Removes all values from a mutable bag.

```
void CFBagRemoveAllValues (
   CFMutableBagRef theBag
);
```

Parameters

theBag

The bag from which all of the values are to be removed.

Availability
Available in CarbonLib v1.0 and later.
Available in Mac OS X v10.0 and later.

Declared In
CFBag.h

CFBagRemoveValue

Removes a value from a mutable bag.

```
void CFBagRemoveValue (
   CFMutableBagRef theBag,
   const void *value
);
```

Parameters

theBag

The bag from which *value* is to be removed.

value

The value to be removed from the collection.

Availability
Available in CarbonLib v1.0 and later.
Available in Mac OS X v10.0 and later.

Declared In
CFBag.h

CFBagReplaceValue

Replaces a value in a mutable bag.

```
void CFBagReplaceValue (
   CFMutableBagRef theBag,
   const void *value
);
```

Parameters

theBag

The bag from which *value* is to be replaced.

value

The value to be replaced in the collection. If this value does not already exist in the collection, the function does nothing. You may pass the value itself instead of a pointer if it is pointer-size or less. The equal callback provided when *theBag* was created is used to compare. If the equal callback was NULL, pointer equality (in C, ==) is used. If *value*, or any other value in *theBag*, is not understood by the equal callback, the behavior is undefined.

Discussion
Depending on the implementation of the equal callback specified when creating *theBag*, the object that is replaced by *value* may not have the same pointer equality.

Availability
Available in CarbonLib v1.0 and later.
Available in Mac OS X v10.0 and later.

Declared In
CFBag.h

CFBagSetValue

Sets a value in a mutable bag.

```
void CFBagSetValue (
   CFMutableBagRef theBag,
   const void *value
);
```

Parameters

theBag

The bag in which *value* is to be set.

value

The value to be set in the collection. If this value already exists in *theBag*, it is replaced. You may pass the value itself instead of a pointer to it if the value is pointer-size or less. If *theBag* is fixed-size and the value is beyond its capacity, the behavior is undefined.

Discussion
Depending on the implementation of the equal callback specified when creating *theBag*, the value that is replaced by *value* may not have the same pointer equality.

Availability
Available in CarbonLib v1.0 and later.
Available in Mac OS X v10.0 and later.

Declared In
`CFBag.h`

Data Types

CFMutableBagRef

A reference to a mutable bag object.

```
typedef struct __CFBag *CFMutableBagRef;
```

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

Declared In
`CFBag.h`

Document Revision History

This table describes the changes to *CFMutableBag Reference*.

Date	Notes
2005-12-06	Made minor changes to text to conform to reference consistency guidelines.
2005-08-11	Cosmetic changes to conform to documentation guidelines.
2003-08-01	Enhanced description of all the `kCFType*Callbacks` and added link to Carbon-Cocoa integration document.
2003-01-01	First version of this document.

Document Revision History

Index

C