ABMultiValue Reference for C

Carbon > Apple Applications

2003-08-20

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

Simultaneously published in the United States and Canada.

Contents

ABMultiValue Reference for C

Derived From:	CFType
Framework:	AddressBook/ABAddressBookC.h
Companion guide	Address Book Programming Guide for Mac OS X
Declared in	ABAddressBookC.h

Overview

The ABMultiValue and ABMutableMultiValue opaque types are used to represent properties that might have multiple values. Each value in a multi-value list must be the same type, and has an associated pre-defined or user-defined label, and unique identifier. The labels, however, need not be unique. For example, you can have multiple "Home" phone numbers. Each multi-value object may have a primary identifier—used to lookup a default value when a label is not provided. For example, a person record may have multiple addresses with the labels "Home" and "Work", where "Work" is designated as the primary value. Instances of this class are immutable, see ABMutableMultiValue for functions that manipulate the content of a multi-value list.

You can access values using a numeric index (similar to an array). Use the `ABMultiValueCopyIdentifierAtIndex` (page 5) function to get an identifier, the `ABMultiValueCopyLabelAtIndex` (page 6) function to get a label, and the `ABMultiValueCopyValueAtIndex` (page 7) function to get a value. However, a numeric index is temporary since a multi-value list may change. Each value or entry in a multi-value list has a unique identifier which can be used to save a reference to a specific value—the identifier is guaranteed never to change.

Use the `ABMultiValueCopyPrimaryIdentifier` (page 7) function to get the primary identifier (the identifier associated with the primary value).

The ABMultiValue opaque type is "toll-free bridged" with its Objective-C counterpart. This means that the `ABMultiValueRef` type is interchangeable in function or method calls with instances of the ABMultiValue class.

Functions

ABMultiValueCopyIdentifierAtIndex

Returns the identifier at the given index.

```
CFStringRef ABMultiValueCopyIdentifierAtIndex (
   ABMultiValueRef multiValue,
   CFIndex index
);
```

Parameters

multiValue

The multi-value list that you wish to access.

index

The index of the identifier you wish to obtain. If this parameter is out of bounds, this function raises an exception.

Return Value

The identifier at *index* in *multiValue*. You are responsible for releasing this object.

Discussion

Each value in a multi-value list must be the same type, and has an associated pre-defined or user-defined label, and unique identifier. Use the `ABMultiValueCopyLabelAtIndex` (page 6) function to get a label, and the `ABMultiValueCopyValueAtIndex` (page 7)function to get a value.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`ABAddressBookC.h`

ABMultiValueCopyLabelAtIndex

Returns the label for the given index.

```
CFStringRef ABMultiValueCopyLabelAtIndex (
   ABMultiValueRef multiValue,
   CFIndex index
);
```

Parameters

multiValue

The multi-value list that you wish to access.

index

The index of the identifier you wish to obtain. If this parameter is out of bounds, this function raises an exception.

Return Value

The label at *index* in *multiValue*. You are responsible for releasing this object.

Discussion

Each value in a multi-value list must be the same type, and has an associated pre-defined or user-defined label, and unique identifier. Use the `ABMultiValueCopyIdentifierAtIndex` (page 5) function to get a identifier, and the `ABMultiValueCopyValueAtIndex` (page 7)function to get a value.

Availability

Available in Mac OS X v10.2 and later.

Related Sample Code

AddressBookCarbon

Declared In
ABAddressBookC.h

ABMultiValueCopyPrimaryIdentifier

Returns the identifier for the primary value.

```
CFStringRef ABMultiValueCopyPrimaryIdentifier (
   ABMultiValueRef multiValue
);
```

Parameters

multiValue
The multi-value list that you wish to access.

Return Value
The unique identifier for the primary value. You are responsible for releasing this object.

Discussion
Use the ABMultiValueCopyIdentifierAtIndex (page 5) function to get index for the returned identifier, and the ABMultiValueCopyValueAtIndex (page 7)function to get its value.

Availability
Available in Mac OS X v10.2 and later.

Declared In
ABAddressBookC.h

ABMultiValueCopyValueAtIndex

Returns the value for the given index.

```
CFTypeRef ABMultiValueCopyValueAtIndex (
   ABMultiValueRef multiValue,
   CFIndex index
);
```

Parameters

multiValue
The multi-value list that you wish to access.

index
The index of the identifier you wish to obtain. If this parameter is out of bounds, this function raises an exception.

Return Value
The value at *index* in *multiValue*. You are responsible for releasing this object.

Discussion
Each value in a multi-value list must be the same type, and has an associated pre-defined or user-defined label, and unique identifier. Use the ABMultiValueCopyIdentifierAtIndex (page 5) function to get a identifier, and the ABMultiValueCopyLabelAtIndex (page 6)function to get a label.

Availability
Available in Mac OS X v10.2 and later.

Related Sample Code
AddressBookCarbon

Declared In
ABAddressBookC.h

ABMultiValueCount

Returns the number of entries in a multi-value list.

```
CFIndex ABMultiValueCount (
   ABMultiValueRef multiValue
);
```

Parameters

multiValue

The multi-value list that you wish to access.

Return Value
The number of entries in *multiValue*.

Availability
Available in Mac OS X v10.2 and later.

Related Sample Code
AddressBookCarbon

Declared In
ABAddressBookC.h

ABMultiValueCreate

Returns a new ABMultiValue object.

```
ABMultiValueRef ABMultiValueCreate (
   void
);
```

Return Value
A new ABMultiValue object. You are responsible for releasing this object.

Availability
Available in Mac OS X v10.2 and later.

Declared In
ABAddressBookC.h

ABMultiValueCreateCopy

Returns a copy of a multi-value object.

```
ABMultiValueRef ABMultiValueCreateCopy (
   ABMultiValueRef multiValue
);
```

Parameters

multiValue

The multi-value object you wish to copy. You are responsible for releasing this object.

Return Value
A copy of *multiValue*.

Availability
Available in Mac OS X 10.2 and later.

Declared In
ABAddressBookC.h

ABMultiValueCreateMutableCopy

Returns a mutable copy of a multi-value object.

```
ABMutableMultiValueRef ABMultiValueCreateMutableCopy (
   ABMultiValueRef multiValue
);
```

Parameters

multiValue

The multi-value object you wish to copy.

Return Value
A mutable copy of *multiValue*. You are responsible for releasing this object.

Availability
Available in Mac OS X v10.2 and later.

Declared In
ABAddressBookC.h

ABMultiValueIndexForIdentifier

Returns the index for the given identifier.

```
CFIndex ABMultiValueIndexForIdentifier (
   ABMultiValueRef multiValue,
   CFStringRef identifier
);
```

Parameters

multiValue

The multi-value list that you wish to access.

identifier

The identifier whose index you wish to obtain.

Return Value
The index of *identifier*.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

ABMultiValuePropertyType

Returns the type for the values in a multi-value list.

```
ABPropertyType ABMultiValuePropertyType (
   ABMultiValueRef multiValue
);
```

Parameters

`multiValue`

The multi-value list whose property type you wish to obtain.

Return Value
The property type of *`multiValue`*. If the list is empty or its values are of different types, returns `kABErrorInProperty`.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

Data Types

ABMultiValueRef

A reference to an ABMultiValue or ABMutableMultiValue object.

```
typedef const struct __ABMultiValue *ABMultiValueRef;
```

Availability
Available in Mac OS X v10.2 and later.

Declared In
`ABAddressBookC.h`

Document Revision History

This table describes the changes to *ABMultiValue Reference for C*.

Date	Notes
2003-08-20	Revised for Mac OS X v10.3.
2003-03-01	First version of *ABMultiValue Reference for C*.

Document Revision History

Index

A