Documentation Language: Swift

Address Book / ABMultiValue

Class

ABMultiValue

An immutable representation of a property that might have multiple values.

macOS

```
class ABMultiValue
```

Overview

Each value in a multivalue list must be of the same type, and must have an associated predefined or user-defined label, and unique identifier. The labels, however, need not be unique. For example, you can have multiple Home phone numbers. Each multivalue object may have a primary identifier—used as 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 methods that manipulate the content of a multivalue list.

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

Topics

Accessing the primary identifier

```
func primaryIdentifier() -> String!
```

Returns the identifier for the primary value.

Accessing identifiers

```
func identifier(at: Int) -> String!
```

Returns the identifier for the given index.

```
func index(forIdentifier: String!) -> Int
```

Returns the index for the given identifier.

Accessing entries

```
func label(at: Int) -> String!
```

Returns the label for the given index.

```
func value(at: Int) -> Any!
```

Returns the value for the given index.

```
func value(forIdentifier: String!) -> Any!
```

Returns the value for the given identifier.

```
func label(forIdentifier: String!) -> Any!
```

Returns the label for the given identifier.

Querying the list

```
func count() -> Int
```

Returns the number of entries in a multivalue list.

```
func propertyType() -> ABPropertyType
```

Returns the type for the values in a multivalue list.

Relationships

Inherits From

`NSObject`

Inherited By

`ABMutableMultiValue`

Conforms To

`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`
`NSCopying`
`NSFastEnumeration`
`NSMutableCopying`
`NSObjectProtocol`

See Also

Data Types

`class ABPerson`

An object that encapsulates all information about a person in the Address Book database.

`class ABGroup`

An object that represents a group of records in the Address Book database.

`class ABMutableMultiValue`

A mutable representation of a property that might have multiple values.

`protocol ABImageClient`

Methods for responding to a request to load images associated with a contact.

`class ABRecord`

An abstract class that defines the common properties for all Address Book records.