Compliance Checklist

Follow the steps summarized in this section to ensure your objects are key-value coding compliant. See the previous sections for details.

Attribute and To-One Relationship Compliance

For each property that is an attribute or a to-one relationship:

- ✓ Implement a method named `<key>` or `is<Key>`, or create an instance variable `<key>` or `_<key>`. The compiler typically does this for you when it automatically synthesizes properties.

> NOTE
> Although property names frequently begin with a lowercase letter, the default implementation of the protocol also works with names that begin with an uppercase letter, such as `URL`.

- ✓ If the property is mutable, implement the `set<Key>:` method. The compiler typically does this for you when you allow it to automatically synthesize your properties.

> IMPORTANT
> If you override the default setter, be sure not to invoke any of the protocol's validation methods.

- ✓ If the property is a scalar, override the `setNilValueForKey:` method to gracefully handle the case where a `nil` value is assigned to the scalar property.

On This Page

Indexed To-Many Relationship Compliance

For each property that is an ordered, to-many relationship (such as an `NSArray` object):

- ✓ Implement a method named `<key>` that returns an array, or have an array instance variable named `<key>` or `_<key>`. The compiler typically does this for you when it automatically synthesizes properties.
- ✓ Alternatively, implement the method `countOf<Key>` and one or both of `objectIn<Key>AtIndex:` and `<key>AtIndexes:`.
- ✓ Optionally, implement `get<Key>:range:` to improve performance.

In addition, if the property is mutable:

- ✓ Implement one or both of the methods `insertObject:in<Key>AtIndex:` and `insert<Key>:atIndexes:`.
- ✓ Implement one or both of the methods `removeObjectFrom<Key>AtIndex:` and `remove<Key>AtIndexes:`.
- ✓ Optionally, implement `replaceObjectIn<Key>AtIndex:withObject:` or `replace<Key>AtIndexes:with<Key>:` to improve performance.

Unordered To-Many Relationship Compliance

For each property that is an unordered, to-many relationship (such as an `NSSet` object):

- ✓ Implement the `<key>` that returns a set, or have an `NSSet` instance variable named `<key>` or `_<key>`. The compiler typically does this for you when it automatically synthesizes properties.
- ✓ Alternatively, implement the methods `countOf<Key>`, `enumeratorOf<Key>`, and `memberOf<Key>:`.

In addition, if the property is mutable:

- ✓ Implement one or both of the methods `add<Key>Object:` and `add<Key>:`.
- ✓ Implement one or both of the methods `remove<Key>Object:` and `remove<Key>:`.

- ✓ Optionally, implement `intersect<Key>:` to improve performance.

Validation

Opt in to validation for properties that need it:

- ✓ Implement the `validate<Key>:error:` method, returning a boolean indicating the validity of the value, and a reference to an error object when appropriate.

On This Page