
NSException Additions

Inherits From: NSObject
Declared In: EOControl/EOClassDescription.h

Class Description

Enterprise Objects Framework adds methods to the Foundation Framework's `NSException` class for handling validating errors and augmenting an exception's **userInfo** dictionary. The methods used for validation errors are **`validationExceptionWithFormat:`** and **`aggregateExceptionWithExceptions:`**. You use **`validationExceptionWithFormat:`** in an enterprise object's **`validateFor...`** or **`validateProperty:`** method, as described in the NSObject Additions class specification. The other method used for validation errors, **`aggregateExceptionWithExceptions:`**, is used internally by the Framework to group multiple validation exceptions together.

The method **`exceptionAddingEntriesToUserInfo:`** is used to augment an exception's **userInfo** dictionary.

Method Types

Creating a validation exception	+ <code>validationExceptionWithFormat:</code>
Collecting exceptions	+ <code>aggregateExceptionWithExceptions:</code>
Returning an exception with an augmented userInfo dictionary	– <code>exceptionAddingEntriesToUserInfo:</code>

Class Methods

`aggregateExceptionWithExceptions:`

+ (NSException *)**`aggregateExceptionWithExceptions:(NSArray *)exceptions`**

Returns the first `NSException` in *exceptions*, with the remaining exceptions collected in its **userInfo** dictionary as an `NSArray` under the key `EOAdditionalExceptionsKey`.

validationExceptionWithFormat:

+ (NSException *)**validationExceptionWithFormat:**(NSString *)*format*, ...

Returns an NSException whose name is EOValidationException and whose reason is an NSString created from *format* and subsequent arguments. For example:

```
[NSException validationExceptionWithFormat:@"invalid name \"%@\": entity names  
cannot be nil or empty", name];
```

Instance Methods

exceptionAddingEntriesToUserInfo:

– (NSException *)**exceptionAddingEntriesToUserInfo:**(NSDictionary *)*additions*

Returns an NSException whose **userInfo** dictionary has been augmented with the object and property information contained in *additions*. When exceptions are raised by certain validation methods such as **validateValue:forKey:**, this message is sent to the exception to create a duplicate exception with object and property information added to the new exception's **userInfo** dictionary. This information is stored in the **userInfo** dictionary under the keys EOValidatedObjectUserInfoKey and EOValidatedPropertyUserInfoKey, respectively. The exception that's returned by this method has the same class with the same name and reason as the original exception; the only difference is the augmented **userInfo** dictionary.