

# Validation

by Craig Federighi, EO Development Team

## Overview

This example presents a simple, generic scheme to allow validation constraints to be put where they belong: as methods in your enterprise objects. The implementation consists of two parts. First, the `KeyValueValidation` protocol passes dictionaries of proposed values to your object and allows it to return a dictionary of error messages. The default implementation allows your object to implement methods like the following:

```
- (NSString *)validateSalary:(NSNumber *)proposed
{
    if ([proposed doubleValue] < 1000.0)
        return @"Salaries must be greater than $1,000";

    return nil; // no error
}
```

The second part of the implementation is a reusable controller delegate that will call the validation routines at the important points: when the user makes a change to a property,

and before an object is inserted, updated, or deleted in the data source.

## The Example Folder

The example is composed of the following:

Validation.subproj-	A application-independent validation scheme.
KeyValueValidation.h	- A protocol for asking objects to validate properties.
ValidatingDelegate	- An EOController delegate that calls the object to perform validation during willSaveEdits, willInsertObject:inDataSource, etc.
EOAcceptsNullFor...	- An (unrelated) example of how objects can tell EOF to turn database NULLs into zeros (or any other values) when they are assigned to scalar numeric properties (int, double, etc).
AppController	- Subclass of ValidatingDelegate that adds a check box to turn on and off immediate validation on change.
Employee	- Example EO class that specifies validation constraints.
Department	- Another EO class that participates in validating an

Employee (no employee can make more than ten times as much as others in the department).

## Running the Example

Try making changes to the city, address, and salary fields in the application. If "validate changes immediately" is checked certain constraints will be checked as soon as you try to leave a field. All constraints are checked before saving to the data source.

### Validation Rules:

The employee class in this example encodes the following constraints:

- Salaries must be greater than \$1000
- A city must be entered (non-blank)
- Addresses may not be longer than 40 characters

Also, employees pass themselves on to their departments for department-specific validation. This check is performed only when the object is about to be inserted or updated in the data source. In this case department adds the constraint:

- No employee may make more than ten times as much as anyone else in the

department.

## **Known Limitations**

This validation support just offers an idea of a useful approach to performing validation under EOF; it is not a complete solution. Current limitations include:

- The default implementation of the KeyValueValidation protocol won't call methods that take C numeric arguments like `setSalary:(double)salary`.
- When performing "immediate" upon the user hitting tab or return, the edit focus is not automatically restored to the field where the error occurred.

## **Possible Enhancements**

- Some simple constraints might best be expressed in the EOModel file (perhaps as additions to the user dictionary) rather than in the EO class. The ValidatingDelegate could be extended to check entries against constraints specified on their corresponding attributes in the model.
- The validation protocol does not provide a way for validators to automatically coerce or correct a proposed value; this might be useful in some circumstances.

## Tips for moving from a Sybase to an Oracle database

This example works for Sybase currently. However, you can easily make it work for Oracle. When moving from a Sybase to an Oracle based example, you'll need to create a new **.eomodel** file from the new server. However, if you keep the internal names of the database as well as the external names identical, you won't have to redo all the connections inside Interface Builder. Naturally, you should also test your new Oracle application to make sure that it behaves exactly like the version on Sybase.

For example:

Prepare a People\_Oracle.eomodel

## Change History

16/Feb/95 Created for EOF 1.1 release (Craig Federighi)