

JFace Databinding

Frank Gerhardt
Michael Scharf
Boris Bokowski

Data Binding

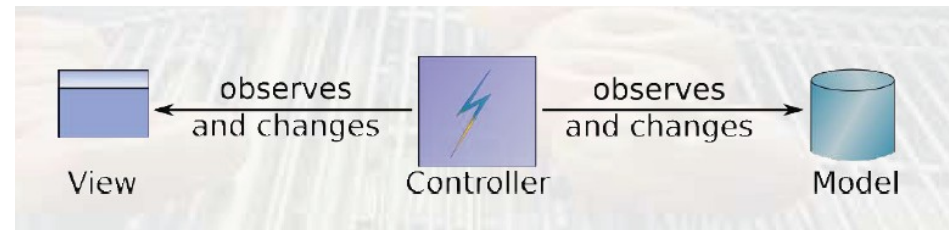
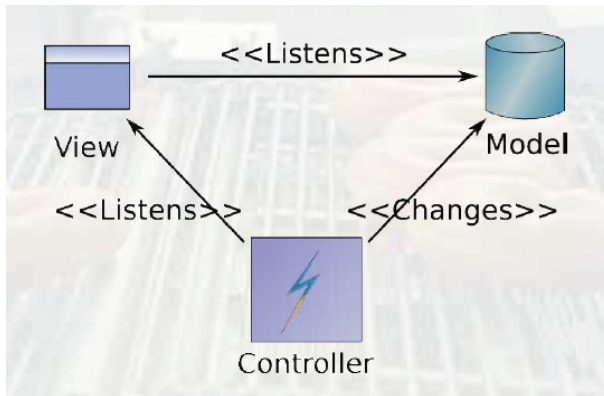
Vision: Get rid of listeners in UI code.

Why?

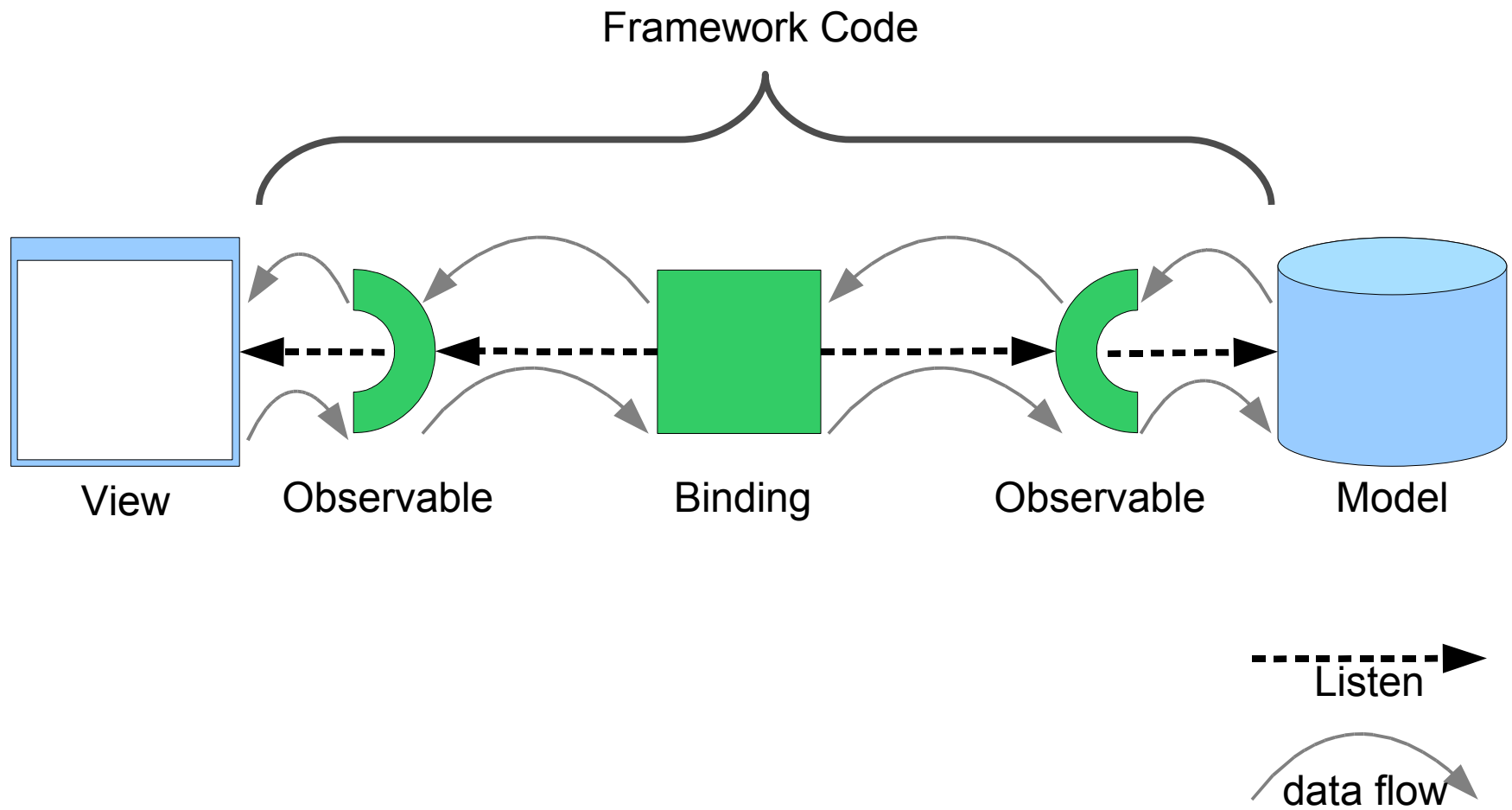
- Hard to write, hard to maintain
- For every aspect:
 - Copy initial state into widget
 - Hook listener (to widget, to model)
 - Write code that sync's state incrementally
 - Validation, conversion typically not separated
 - threading

Triangle -> Straight Line

- From model-view-controller (MVC) triangle to more **independence**

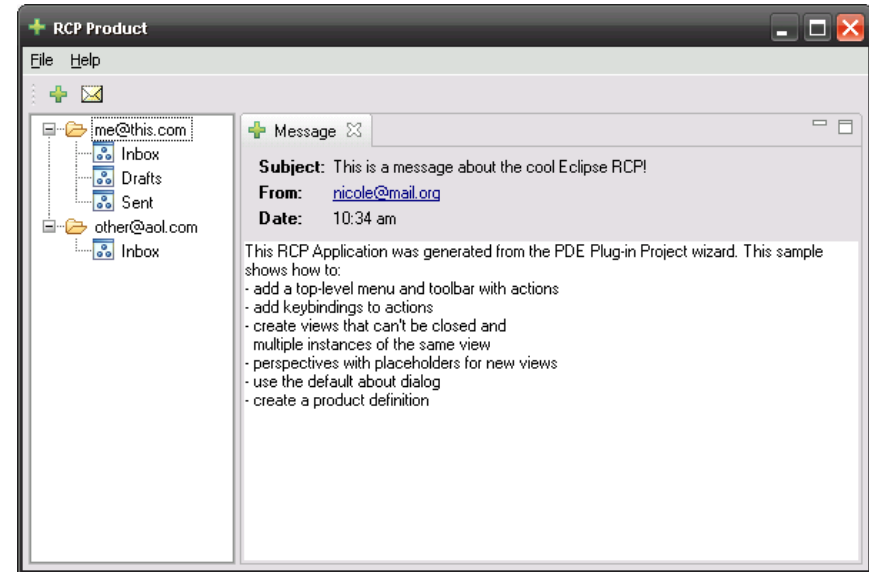


Concepts



Running Example: RCP Mail

- RCP Mail is the base of our data binding example
- We added some GUI elements to show various bindings



RCP Mail enhanced

1. Introduce model

2. Add a wizard dialog

4. Validation errors collected in title area

6a. Updating the table based on selection in tree

6b. Binding of table viewer



Create Mail Server

Create Server Connection

Invalid character for value [23,dfe3e] at position [3] character [.]

Hostname: localhost

Username:

Password:

Port: 23,dfe3e

Finish Cancel

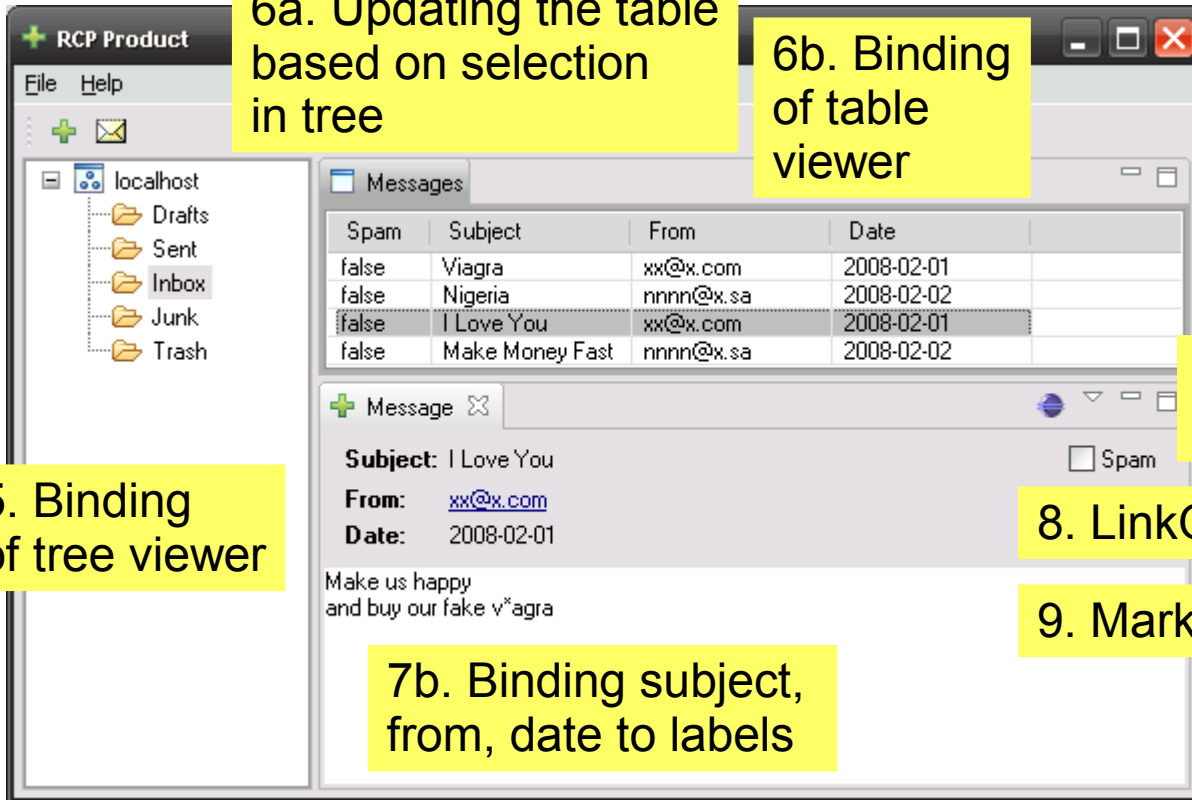
3. Binding text field to string and int property

7a. Master-detail from message list to message

8. LinkObservableValue

9. Mark As Spam (Action)

7b. Binding subject, from, date to labels



RCP Product

File Help

- localhost
 - Drafts
 - Sent
 - Inbox
 - Junk
 - Trash

Messages

| Spam | Subject | From | Date |
|-------|-----------------|-----------|------------|
| false | Viagra | xx@x.com | 2008-02-01 |
| false | Nigeria | nynn@x.sa | 2008-02-02 |
| false | I Love You | xx@x.com | 2008-02-01 |
| false | Make Money Fast | nynn@x.sa | 2008-02-02 |

Message

Subject: I Love You

From: xx@x.com

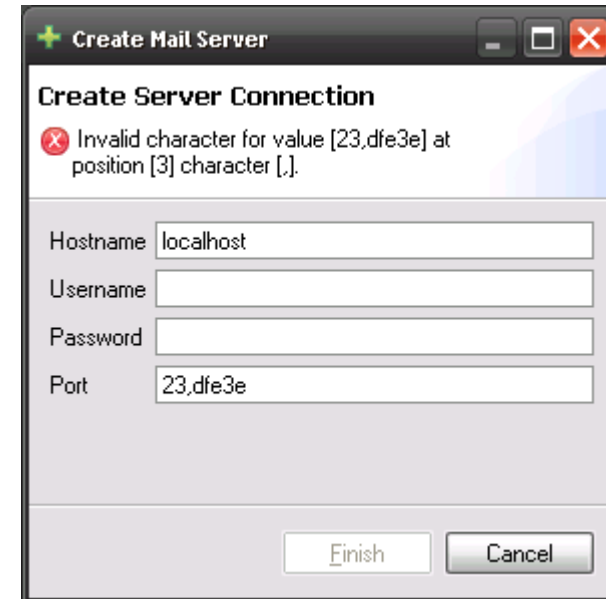
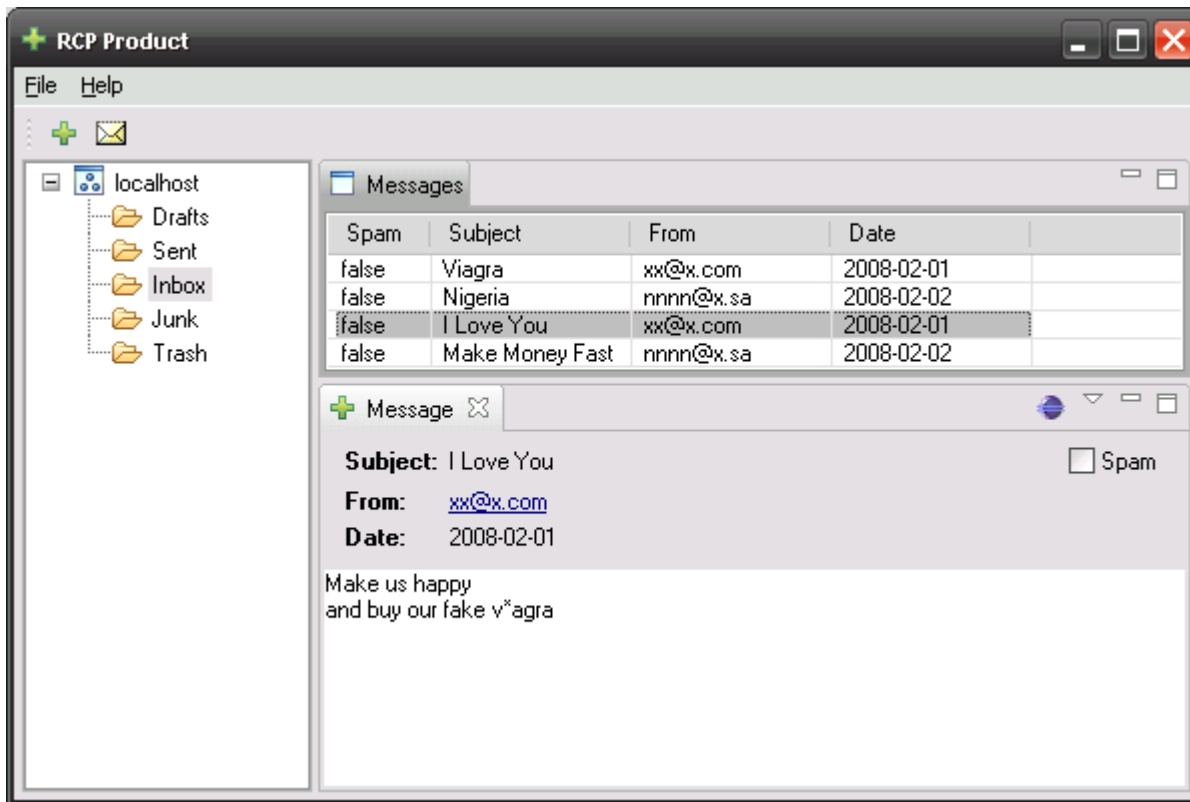
Date: 2008-02-01

Make us happy
and buy our fake v*agra

5. Binding of tree viewer

RCP Mail enhanced

1. Introduce model



PropertyChange Support



- No other requirements

```
import java.beans.PropertyChangeListener;  
import java.beans.PropertyChangeSupport;
```

```
private final PropertyChangeSupport propertyChangeSupport = new PropertyChangeSupport(this);
```

```
public void addPropertyChangeListener(PropertyChangeListener listener) {  
    propertyChangeSupport.addPropertyChangeListener(listener);  
}
```

```
public void addPropertyChangeListener(String propertyName, PropertyChangeListener listener) {  
    propertyChangeSupport.addPropertyChangeListener(propertyName, listener);  
}
```

```
public void removePropertyChangeListener(PropertyChangeListener listener) {  
    propertyChangeSupport.removePropertyChangeListener(listener);  
}
```

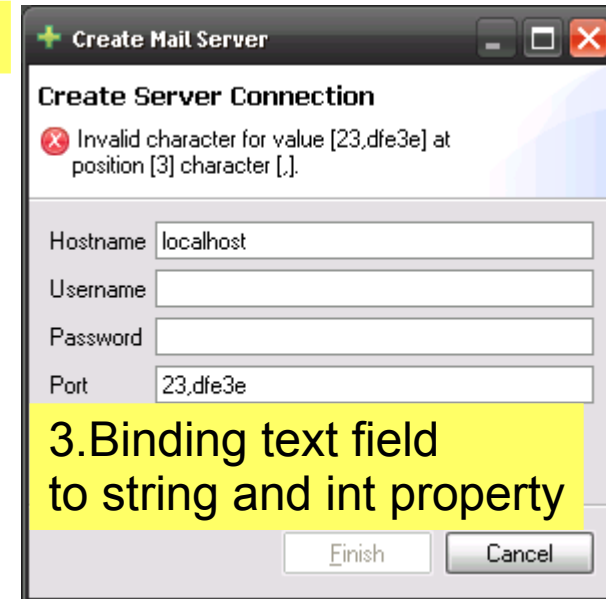
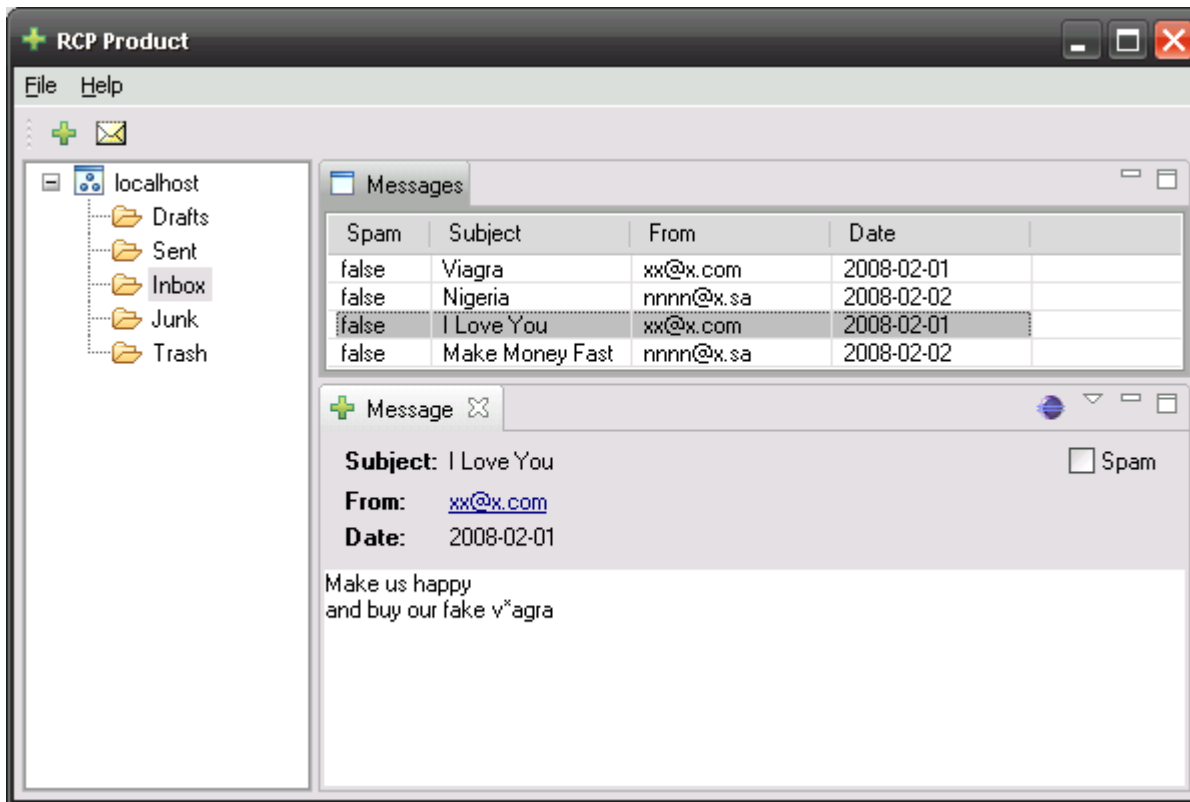
```
public void removePropertyChangeListener(String propertyName, PropertyChangeListener listener) {  
    propertyChangeSupport.removePropertyChangeListener(propertyName, listener);  
}
```

```
protected void firePropertyChange(String propertyName, Object oldValue, Object newValue) {  
    propertyChangeSupport.firePropertyChange(propertyName, oldValue, newValue);  
}
```

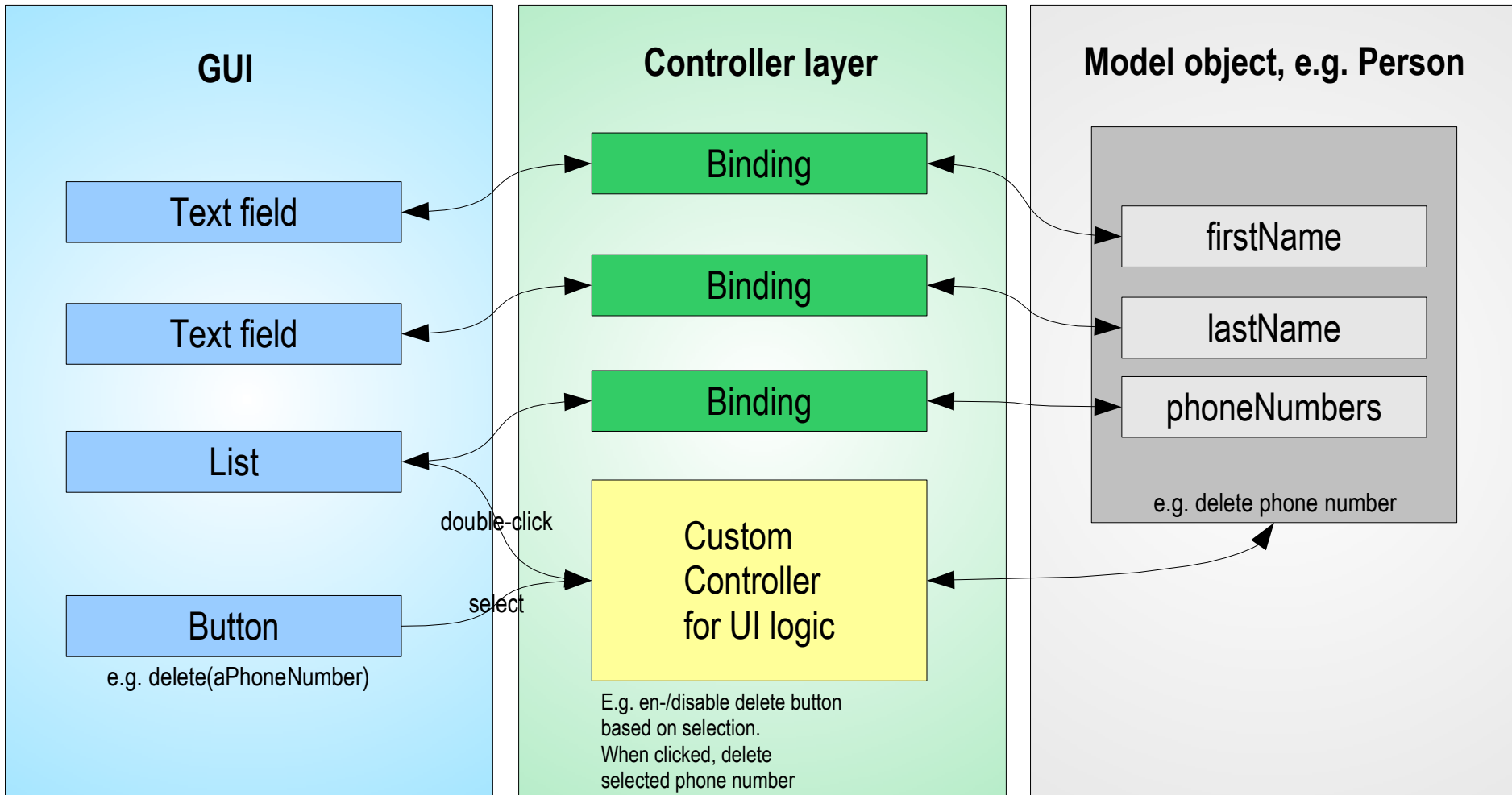

RCP Mail enhanced

1. Introduce model

2. Add a wizard dialog

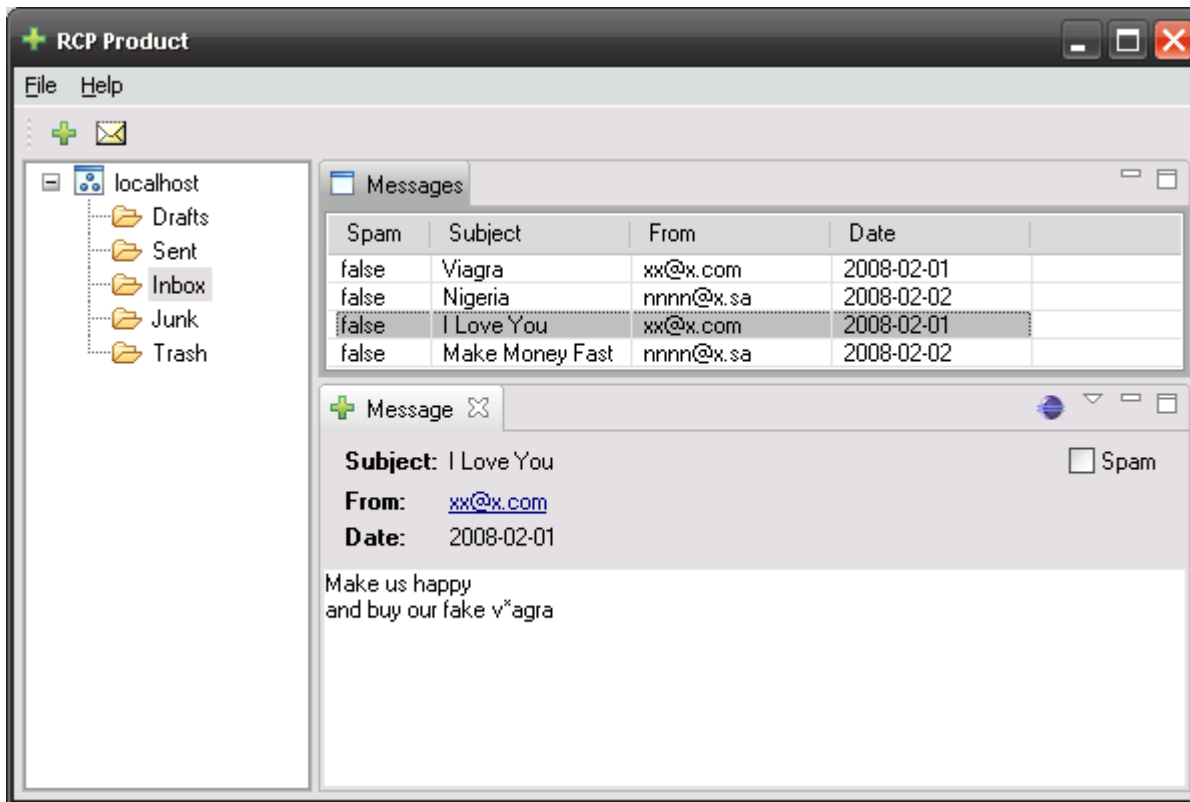


Architecture with DB



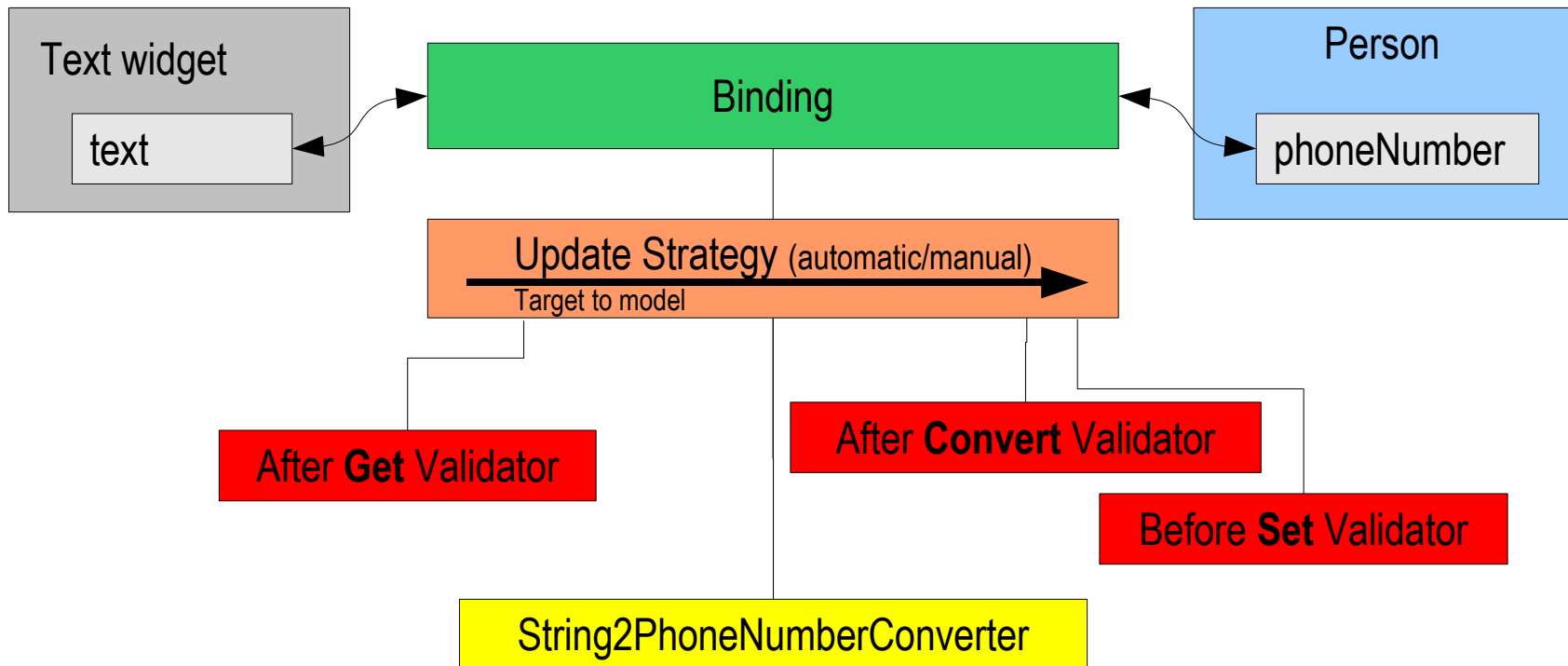
RCP Mail enhanced

4. Validation errors collected in title area

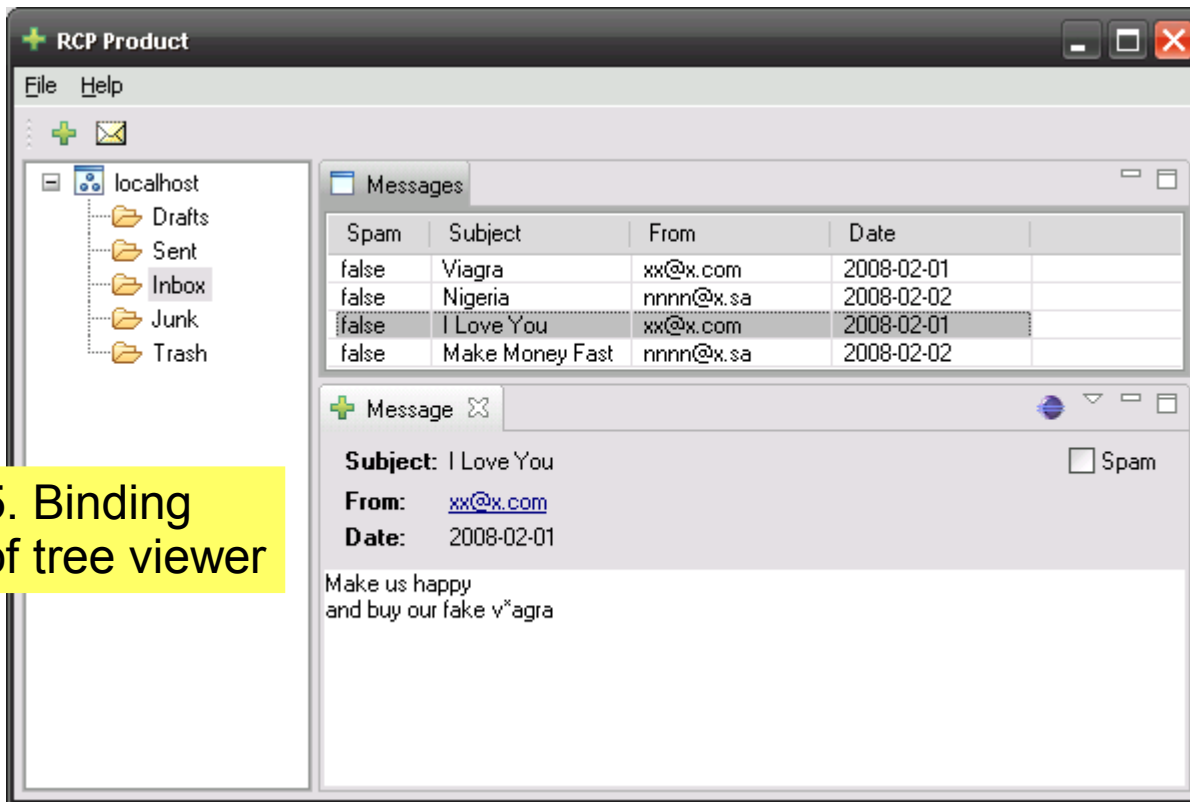


Converters & Validators

The **Binding** can be configured:
model to target **update strategy**
target to model **update strategy**
each with: 1 **Converter** and
3 **Validators**



RCP Mail enhanced



5. Binding of tree viewer

Note: non-API in 3.3

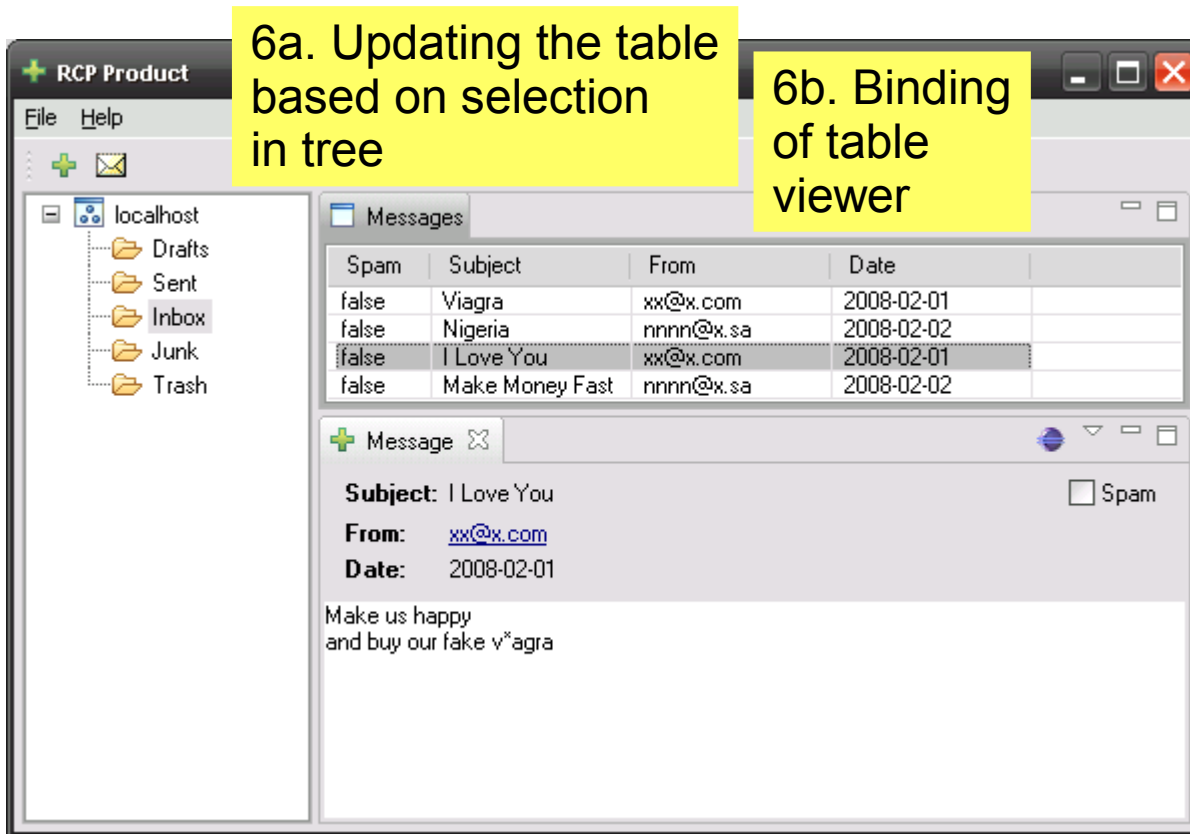


- TreeViewer support will be in 3.4
- Internal/provisional as of 3.3

RCP Mail enhanced

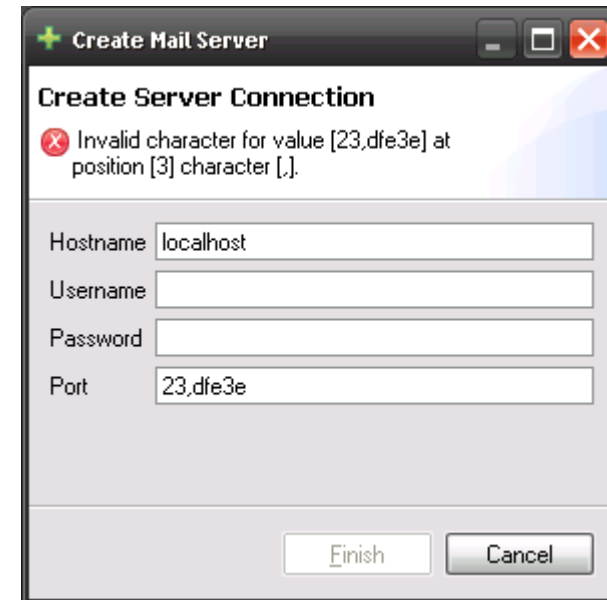
6a. Updating the table based on selection in tree

6b. Binding of table viewer



| Spam | Subject | From | Date |
|-------|-----------------|-----------|------------|
| false | Viagra | xx@x.com | 2008-02-01 |
| false | Nigeria | nnnn@x.sa | 2008-02-02 |
| false | I Love You | xx@x.com | 2008-02-01 |
| false | Make Money Fast | nnnn@x.sa | 2008-02-02 |

Message preview:
Subject: I Love You
From: xx@x.com
Date: 2008-02-01
Make us happy
and buy our fake v*agra



Create Mail Server

Create Server Connection

Invalid character for value [23,dfe3e] at position [3] character [.]

Hostname: localhost

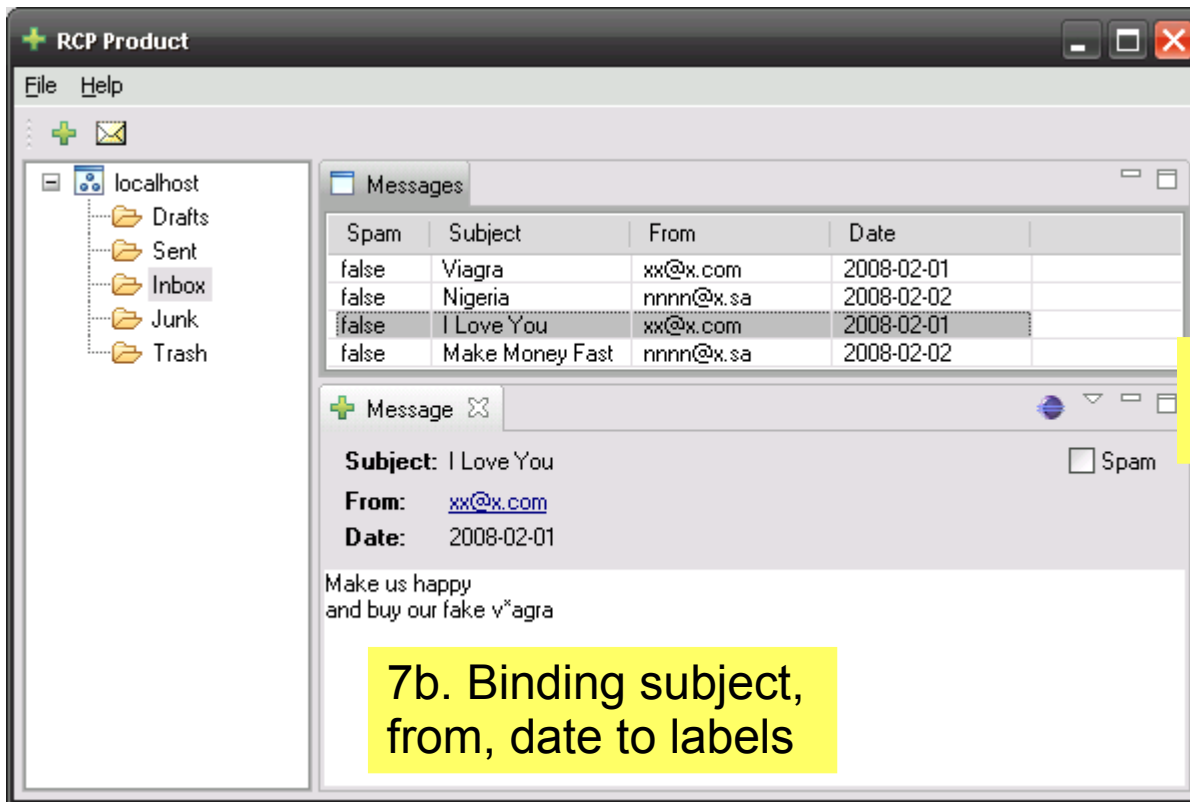
Username:

Password:

Port: 23,dfe3e

Finish Cancel

RCP Mail enhanced

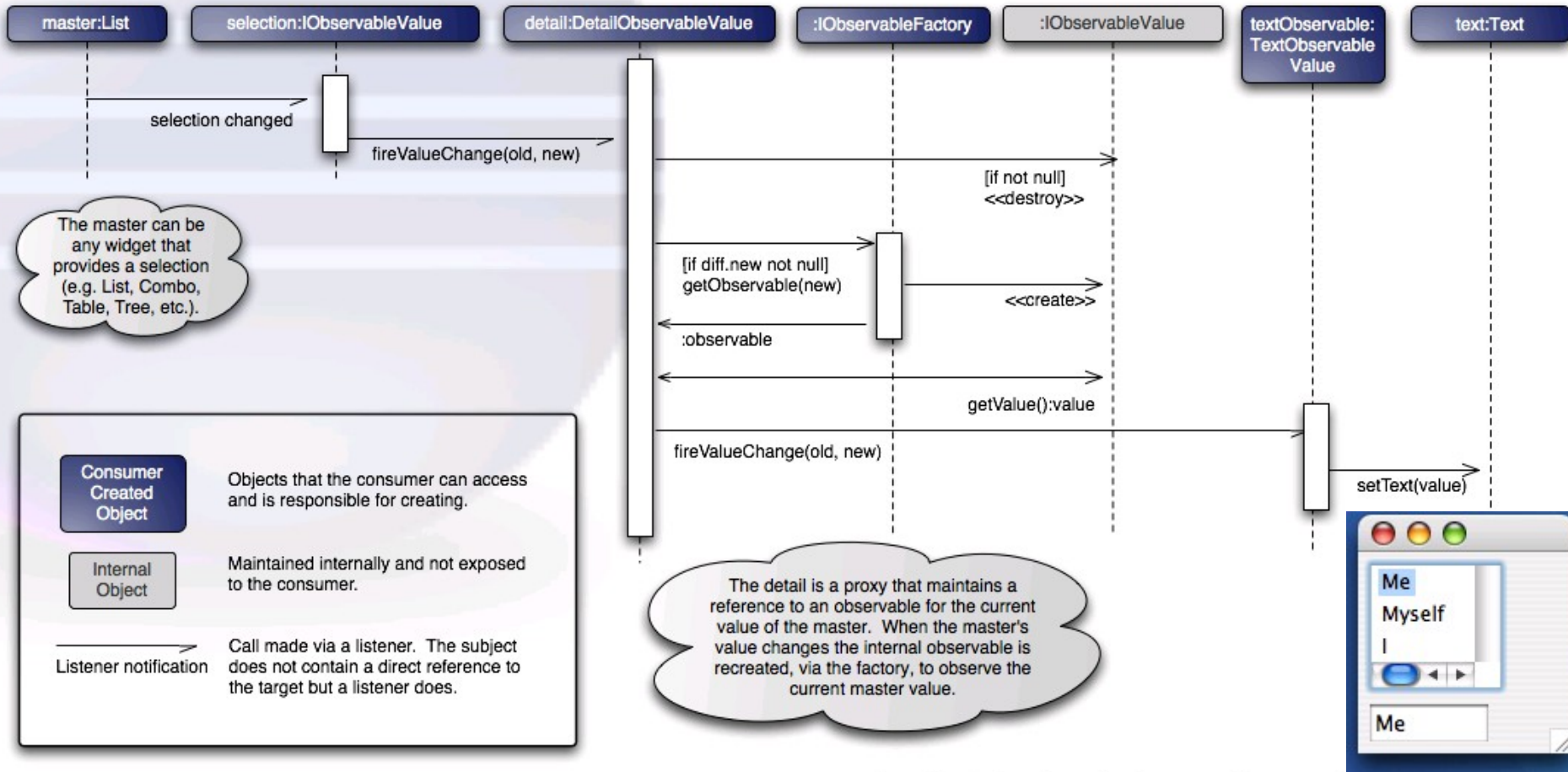


7a. Master-detail from message list to message

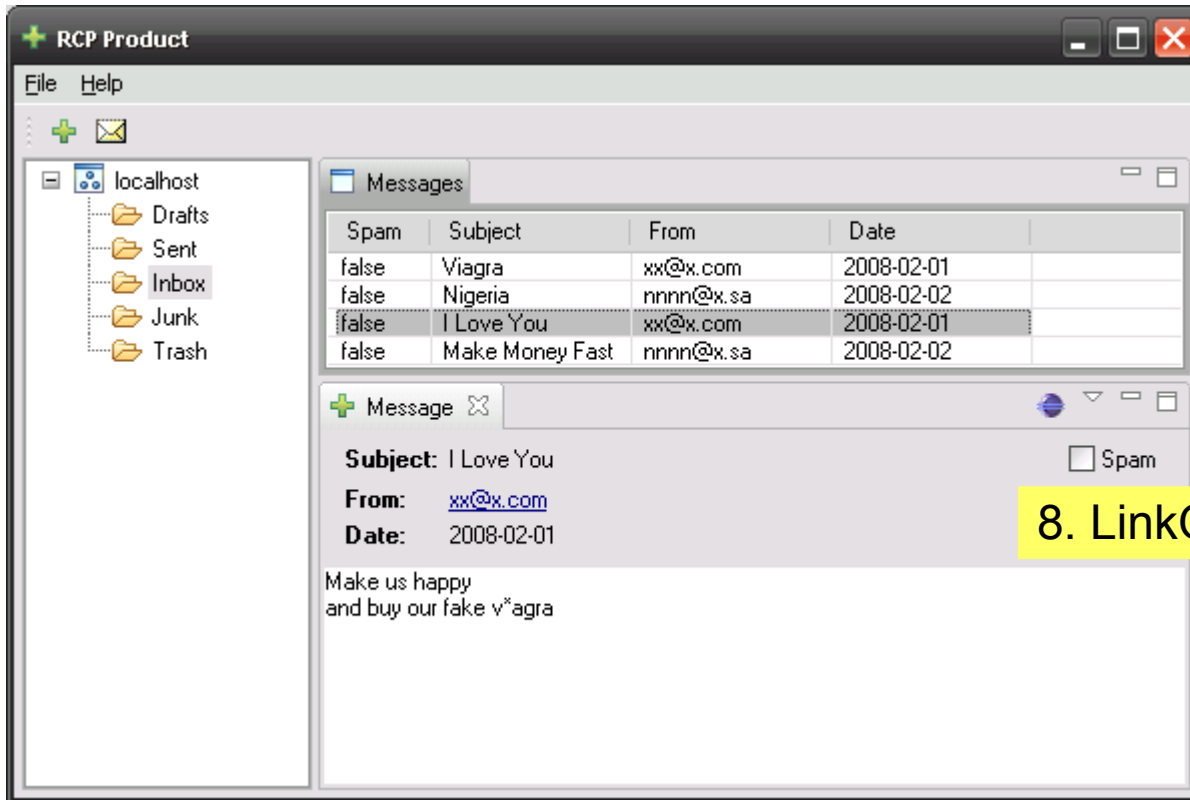
Master-Detail Binding

JFace Data Binding: Master Detail Selection Change

v1.1



RCP Mail enhanced

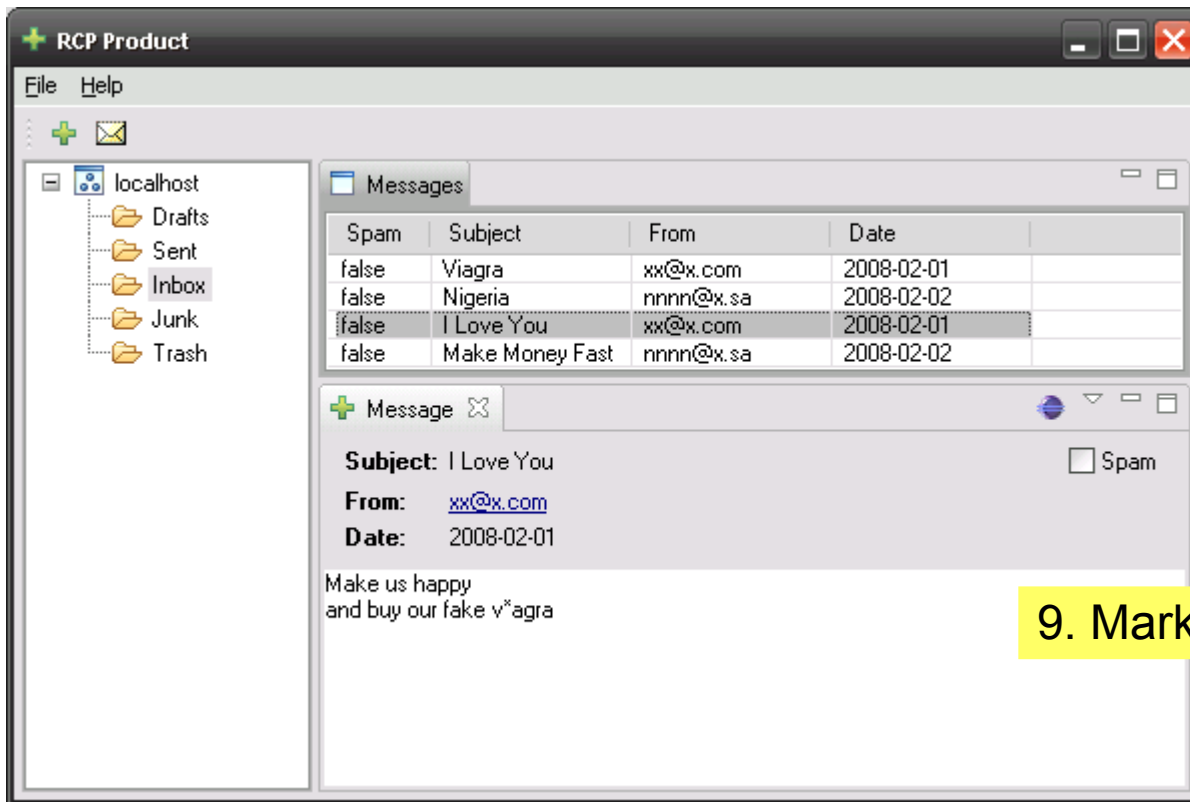


8. LinkObservableValue

Factories

- Observables
- BeansObservables
- SWTObservables
- ViewersObservables
- MasterDetailObservables
- ...add your own!

RCP Mail enhanced



9. Mark As Spam (Action)

Benefits

- Provides structure
- Makes it easy to create multiple views onto the same model objects
- Setting up bi-directional bindings as easy as the uni-directional case
- If your model is changed from a non-UI thread
- Master-detail – just try coding it by hand
- Makes it easy to reuse validation code

Disadvantages

- Learning curve
- Generic implementation makes it complex (create your own helper classes!)
- Overhead (space and time)
- Requires model that fires change events
- Hard to debug when things go wrong

Recap: Features (1)



- **IObservable** and **Binding**
 - Source and target are wrapped
- No superclasses or interfaces required
 - Standard Java beans, with PropertyChangeEvents
 - Other models possible (e.g. EMF, ...)
 - Even other UI toolkits (e.g. GWT, Swing, ...)
- Bindings are set up once
 - Careful design can avoid rebinding (master-detail)
- **Validators**: context collects validation errors
- **Converters** : model-to-UI, UI-to-model
- Pluggable **update strategies**
(immediately, when validation is ok, manual, custom)

Recap: Features (2)



- Binding of
 - **Values** (String, Boolean, Integer...), **1:1**
 - **Lists, n:n**
 - **Master-Detail**
- Can bind to UI state
 - Model.locked to Text.enabled
 - Color, visibility, many more
- Can bind to UI state, selection of list or table
 - Zero additional UI logic required
- Can bind to validation errors

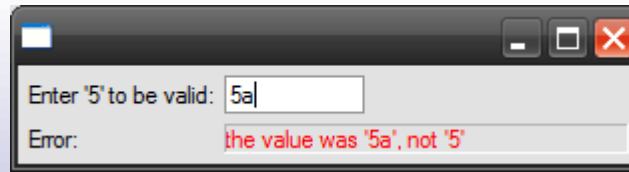
Non-Features

- Can only bind to one level of properties
 - Not a.b.c directly
 - need to bind a-b and b-c
 - Workaround only for values (1:1), not for lists (n:n)
- Trees are not supported well
 - Internal, provisional
- Little documentation, only examples

How-to

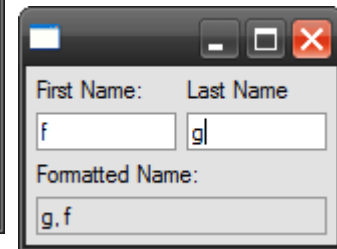
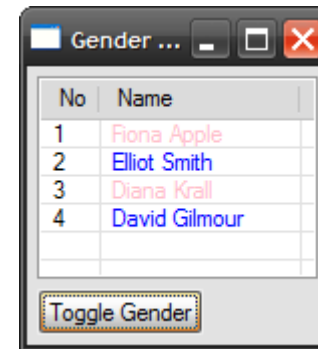
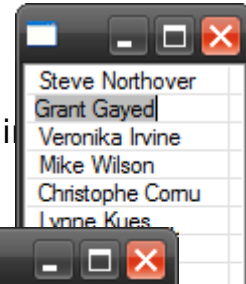
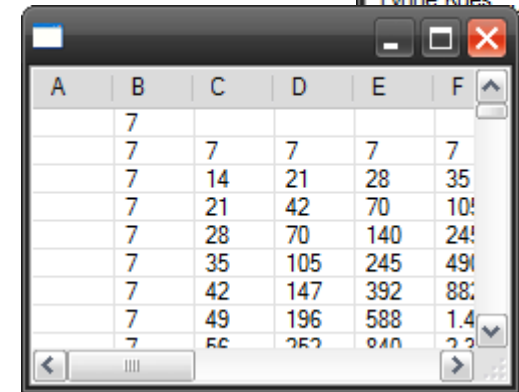
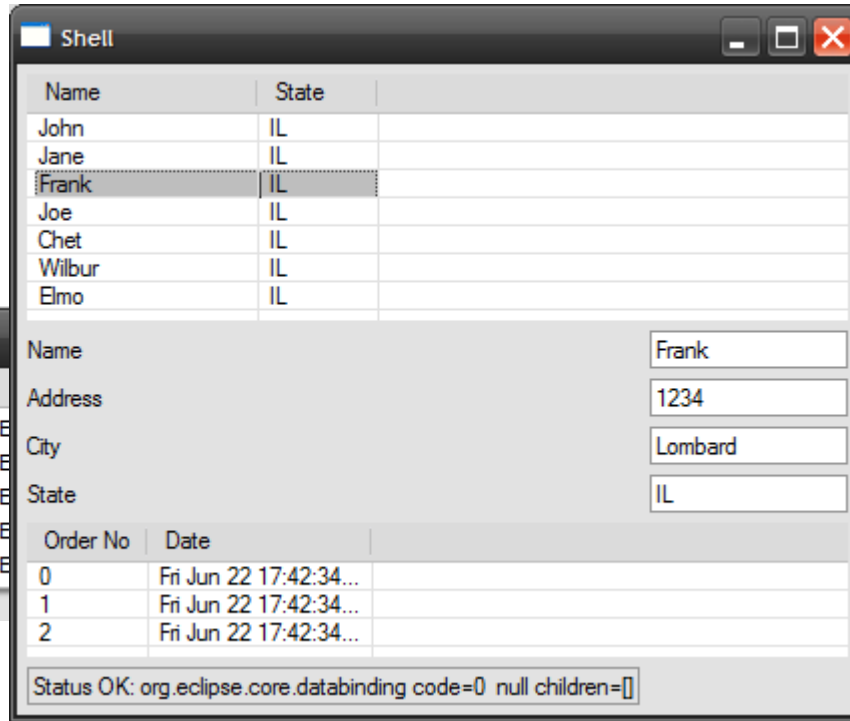
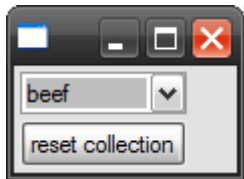
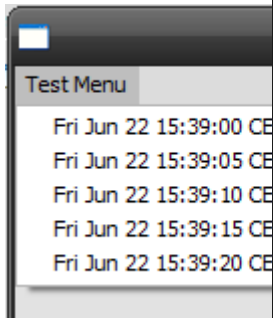
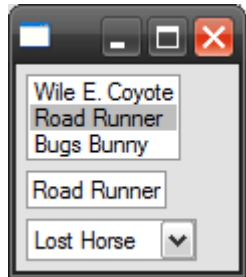
- Decide which type to use
 - Value (1:1), list (n:n), master detail
- For values
 - Wrap model into Observable using BeansObservables Factory
 - Wrap UI into Observable using SWTObservables or ViewerObservables
- For lists
 - Use supplied ContentProvider and setInput

Examples



- From CVS

:pserver:anonymous@dev.eclipse.org:/cvsroot/eclipse/org.eclipse.jface.examples.databinding



Resources

- JFace Databinding Wiki page
 - http://wiki.eclipse.org/index.php/JFace_Data_Binding
- Newsgroup
 - eclipse.platform with [DataBinding] in subject
- Bugzilla
 - Product=Platform, Component=UI, Summary with [DataBinding]
- Examples from CVS

Copyright

- © 2008 Gerhardt Informatics Kft./
Michael Scharf, WindRiver/
IBM Rational Software
- Distributed under Creative Commons
Attribution-Noncommercial-Share Alike
3.0 United States License 

<http://creativecommons.org/licenses/by-nc-sa/3.0/us>