

## Introducing JFace Data Binding



Code name: Krispy Kreme  
(with inspiration from Glazed Lists)

## Purpose

- Explain why you should care about data binding
- Describe how Krispy Kreme works
- Show Krispy Kreme in action
- Point to further information on Krispy Kreme

## Why data binding?

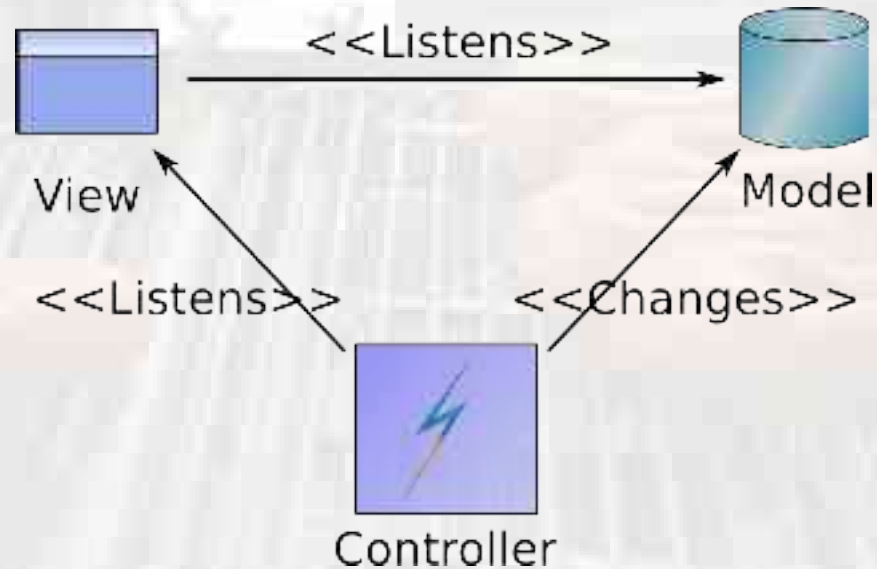
- Writing data driven user interfaces by hand is painful
  - There is lots of repetitive boilerplate event handling code
  - Constantly converting between the model data type (ie: int age) and the UI's data type (ie: String getText / setText) is a pain
  - Validating user data entry is a combinatorial problem where  $n ==$  the number of controls on a data entry form
- To speed up development
  - Current project: ~40% team velocity improvement with Krispy Kreme
- It tastes good 😊

## Why data binding with JFace Krispy Kreme?

- Helps automate some of the most painful areas in an RCP application
- Compatible with standards:
  - Uses standard Java interfaces like `java.util.List` whenever possible
- Extensible:
  - For example: bind from/to remote objects across a network
- ***Result: You deliver Eclipse RCP applications faster, more easily, with fewer defects.***

# What is Krispy Kreme?

- Improves and updates Model-View-Controller

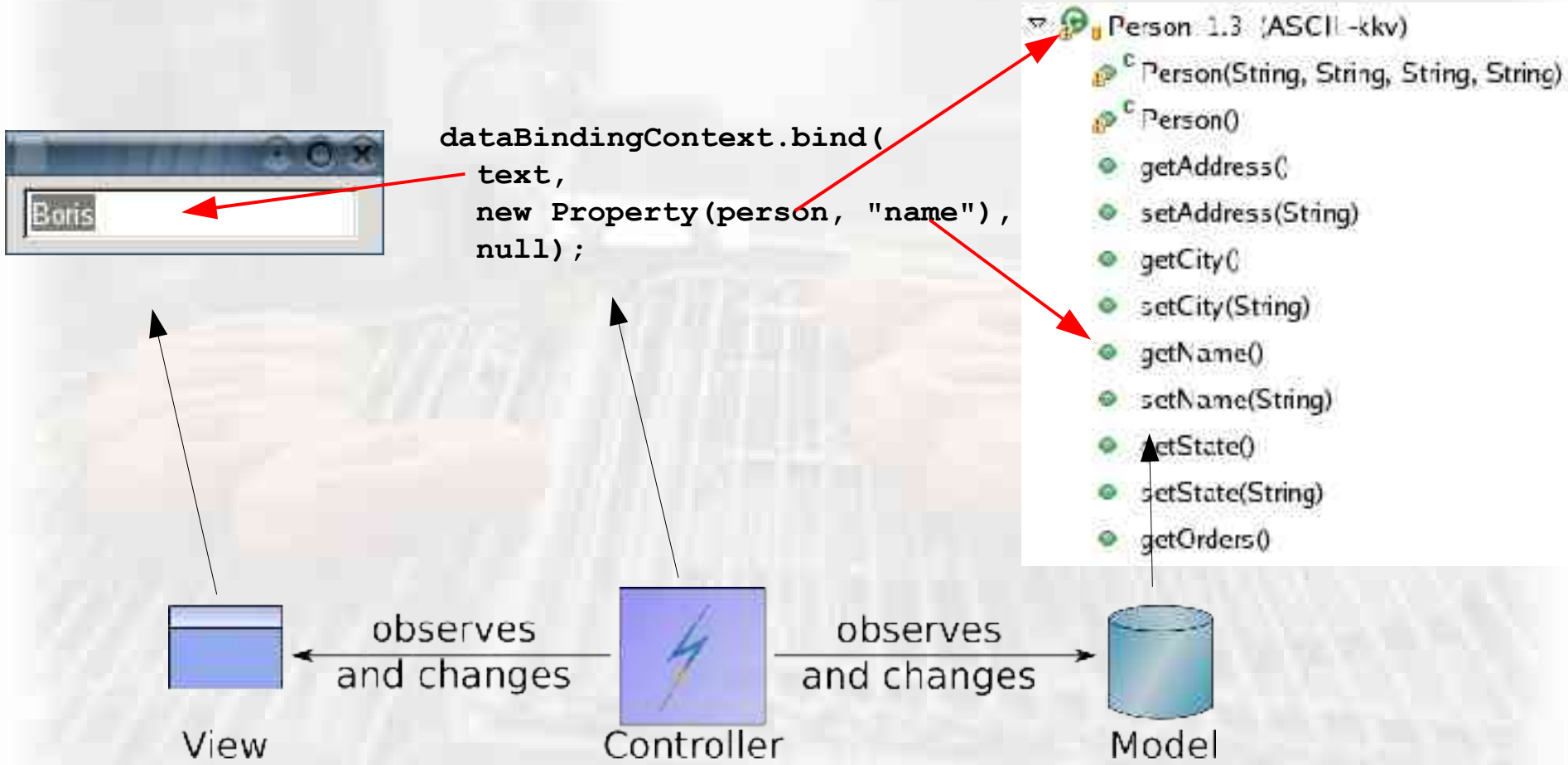


# What is Krispy Kreme?

- Improves and updates Model-View-Controller
  1. Model the data flow directly
  2. Convert and validate data automatically / as necessary



# Getting Started with Krispy Kreme



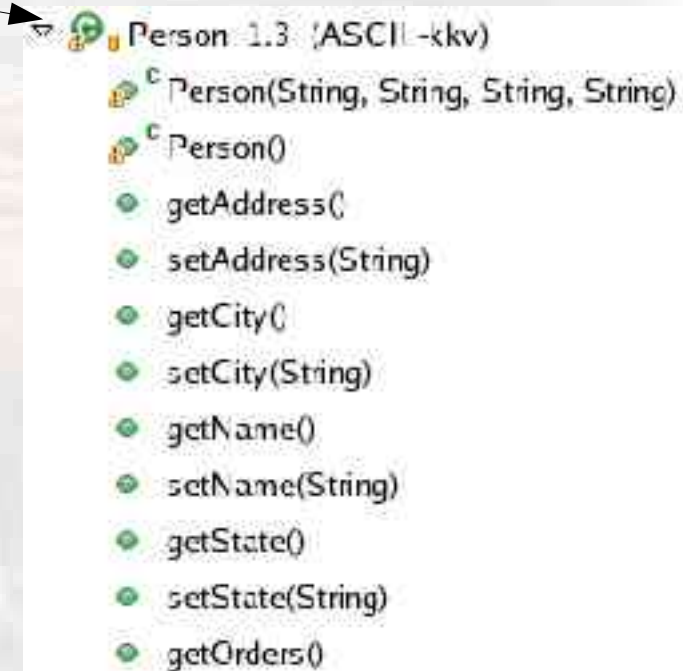
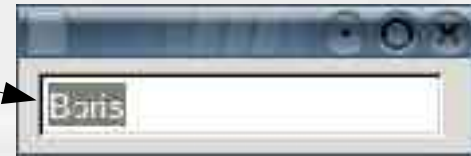
## Getting Started with Krispy Kreme

```
// Make SWT Text control
Text text = new Text(shell, SWT.BORDER);

// Make Person (domain model) object
Person person = new Person("Boris",
    "2670 Queensview Dr", "Ottawa",
    "Canada");

// Bind the Person's name to the SWT Text
DataBindingContext dbc =
    BindingFactory.createContext(shell);

dbc.bind(
    text,
    new Property(person, "name"),
    null);
```





## Extending and customizing Krispy Kreme

- *DataBindingContext methods*
  - *Listen to binding events*
  - *Modify data flow behavior*
  - *Attach global validators*
- *BindSpec parameters*
  - *Change local validation*
  - *Change local data type conversion*
- *Extend the type hierarchy*

```
dataBindingContext.bind(  
    text,  
    new Property(person, "name"),  
    null);
```

## Krispy Kreme:

- Is Tasty 😊
- Helps automate your user interface development
- Is compatible with existing Java standards
- Can be extended to bind to and from anything
- ***The bottom line: With Krispy Kreme you deliver Eclipse RCP applications faster, more easily, with fewer defects.***

## Questions?

- **To learn more:**
  - [http://wiki.eclipse.org/index.php/JFace\\_Data\\_Binding](http://wiki.eclipse.org/index.php/JFace_Data_Binding)
  - **Post to `eclipse.platform.rcp` with `[DataBinding]` in subject**
  
- **Thanks to the following kind folks on Flickr for releasing their Krispy Kreme photos/video under a CC license:**
  - <http://flickr.com/photos/pochacco20/59141976/>
  - <http://flickr.com/photos/francois/3959737/>
  - <http://flickr.com/photos/roboppy/10023293/>
  - <http://flickr.com/photos/joshbousel/25359924/>