

# Accelerating Application Development with Objects



Eric Falsken  
Roman Stoffel  
German Viscuso

**db4objects**  
BY VERSANT

# Agenda

- **Referential Identity**
- **Useful Application Patterns**
- **NerdDinner**
- **ODBMS Implementations**
- **Useful db4o Details**

# Referential Identity

# Referential Identity

- Many O-R Mappers and ODBMS use referential identity to track object updates.

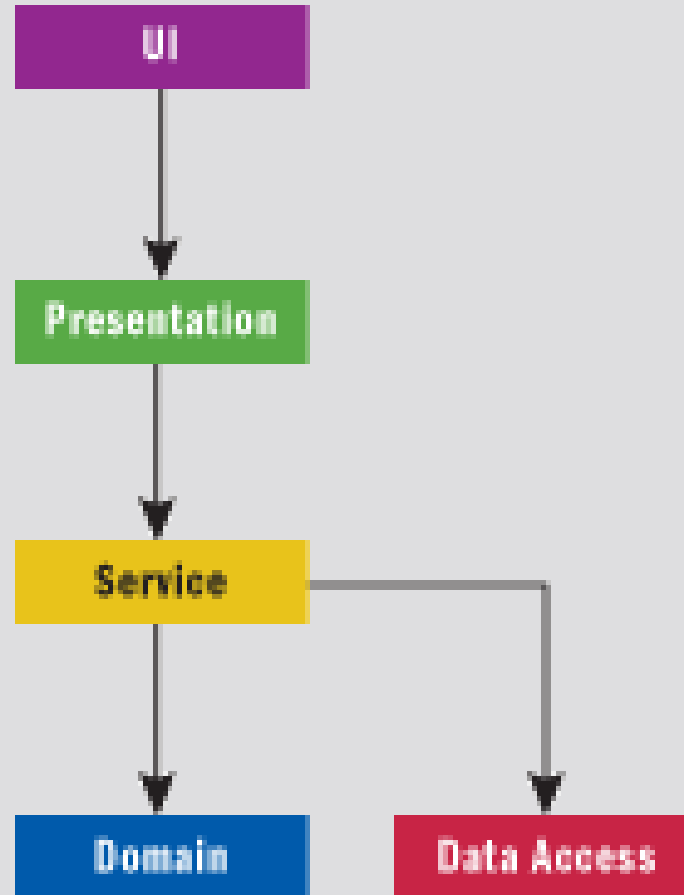
```
var dinner1 = new Dinner() { Title = "A Dinner" };  
var dinner2 = new Dinner() { Title = "A Dinner" };  
// FALSE dinner1 == dinner2  
// TRUE dinner1.Equals(dinner2)
```

- Object reference in memory IS the unique ID of your data model object.
- NOT the ID property you may/not add.
- Watch out in multi-transactional systems and where serialization might be involved.

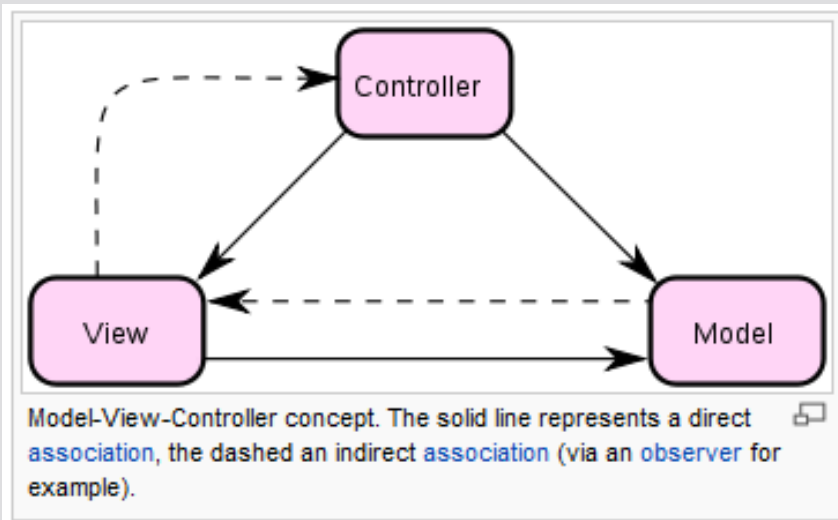
# Useful Application Patterns

# Multi-Tiered

- Row-Based Results
  - ADO.NET DataSet
  - JDBC ResultSet
- *n*-Tier
- Manual Binding
  - Object-to-UI mapping
- ❖ Assisted Binding



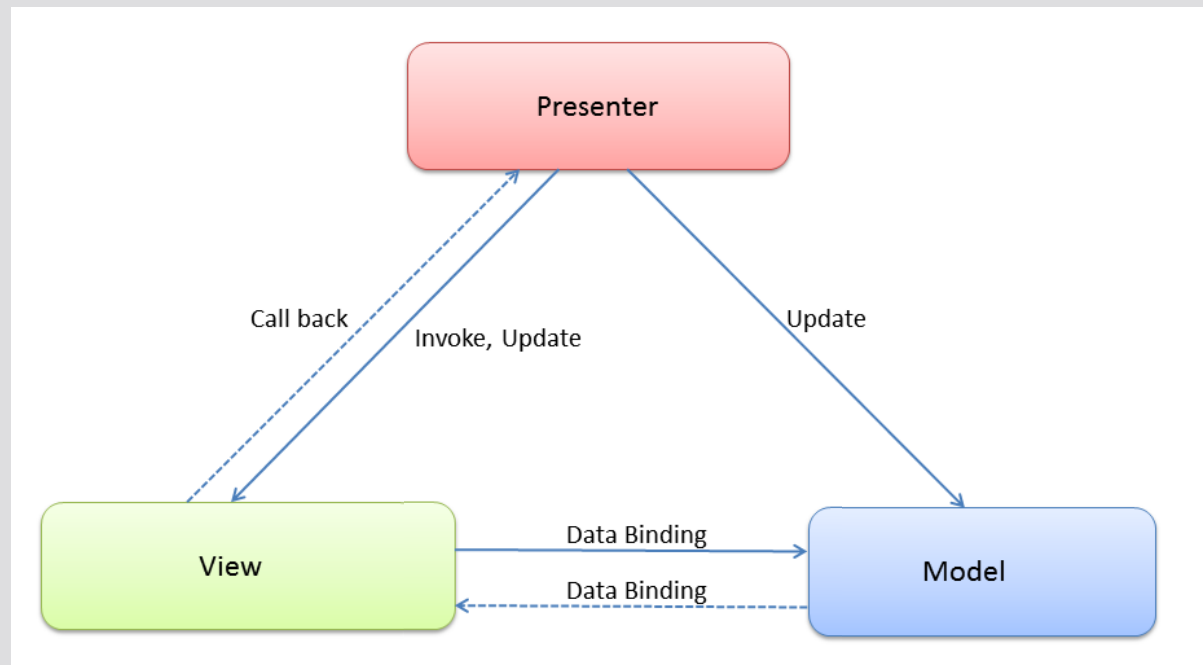
# Model-View-Controller (MVC)



- Decouple Models and Views
  - Decrease complexity
  - Increase flexibility.
- In practice, Views and Controllers are quite closely associated
  - Controller and View are dependent on the Model
  - Difficult to test
  - Difficult to reuse

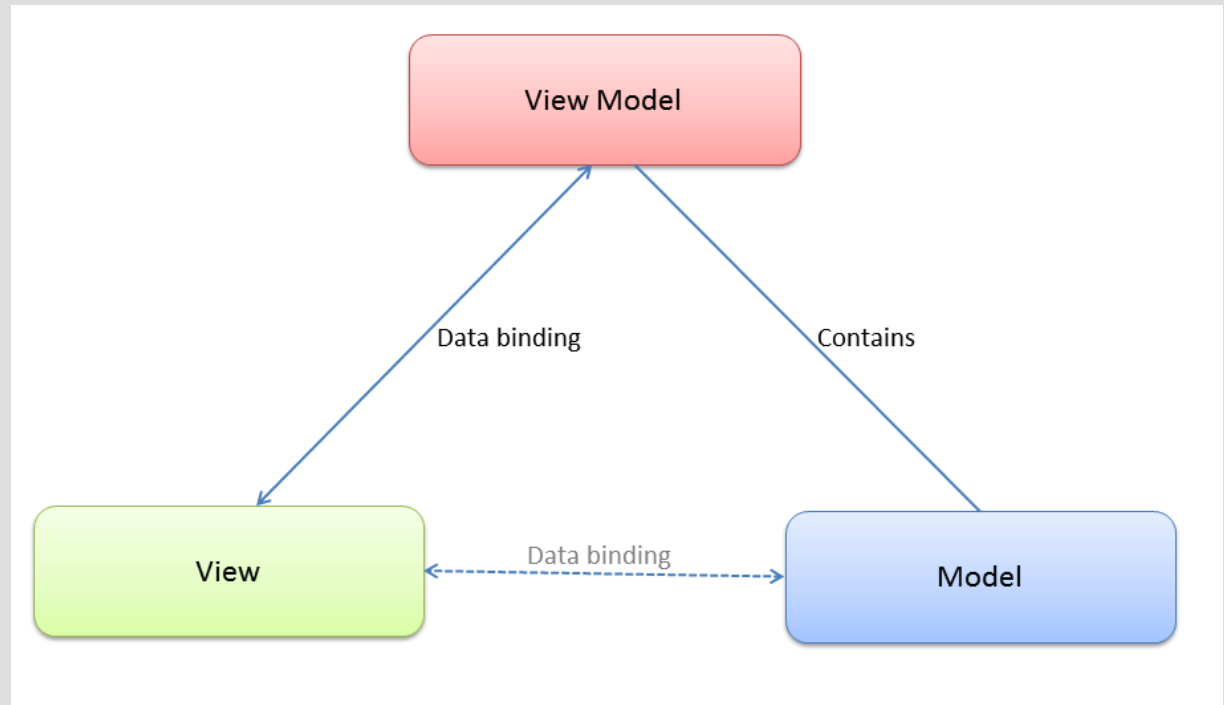
# Model-View-Presenter (MVP)

- View has no dependency on the Model.
  - Tightly bound to the Presenter
- Presenter orchestrates the view
  - Not easily reusable



# Model-View-ViewModel (MVVM)

- **ViewModel – Model of the View**
  - Contracted working set
  - UI hints, helpers & interfaces
  - Syncs back to the Model





nerddinner



**db4objects**  
BY VERSANT



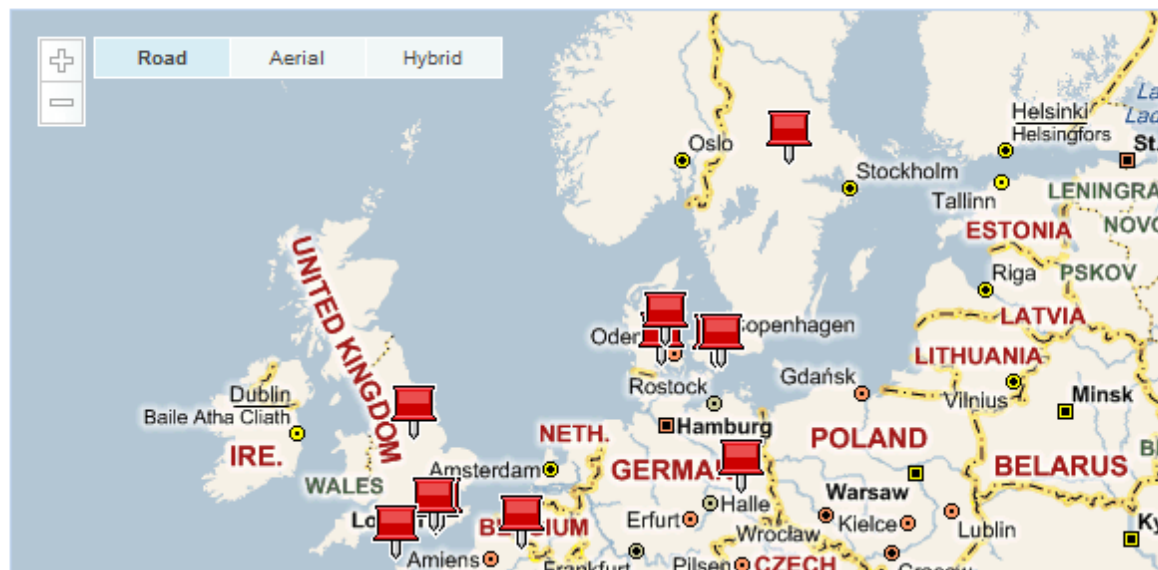
**Organizing** the world's nerds  
and helping them eat in packs.

Enter your location or [View All Upcoming Dinners](#).

Hamburg, germany

Search

## Find a Dinner



## Popular Dinners

[Burek @ Fadila](#)  
Oct 9 with 11 RSVPs

[2010 Opening Ceremony](#)  
Dec 1 with 48 RSVPs

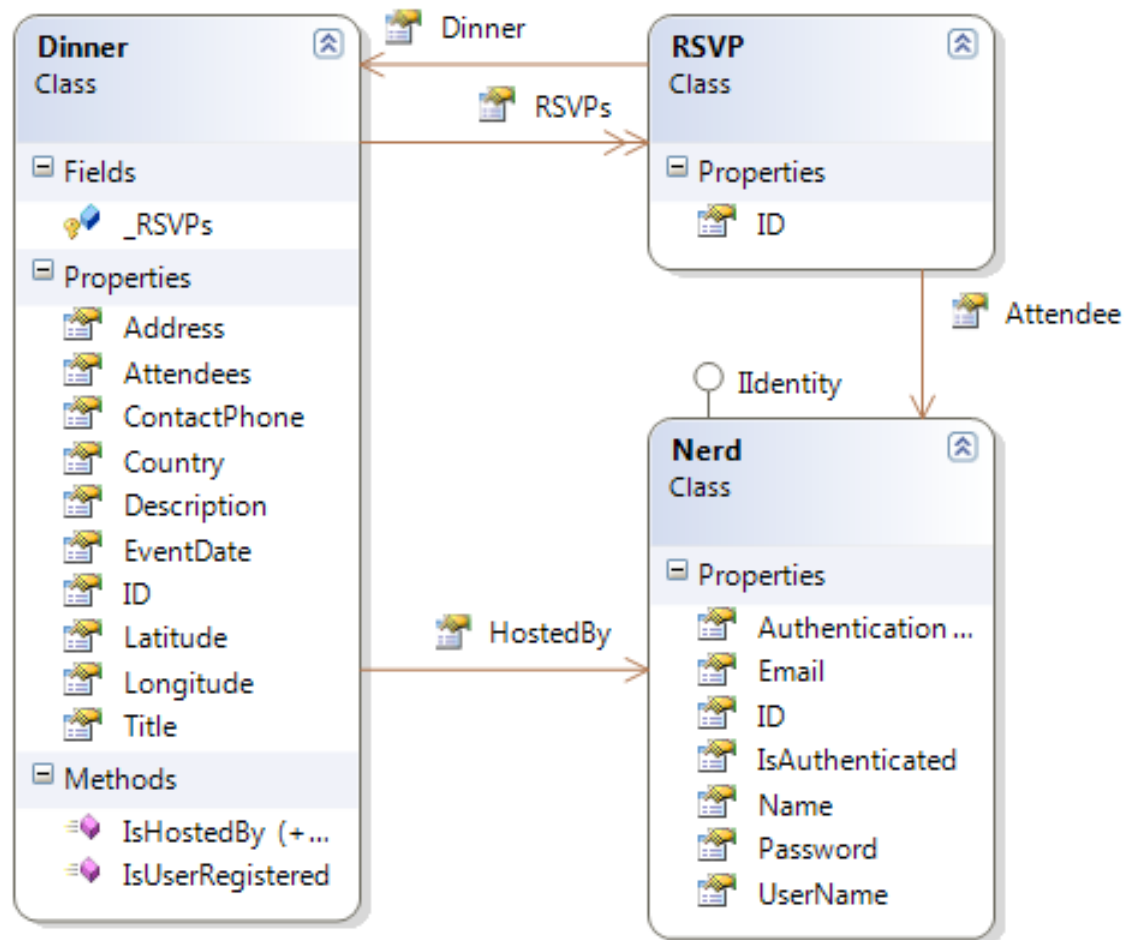
[Chicken Teriaki](#)  
Dec 2 with 19 RSVPs

[New Year Dinner](#)  
Dec 30 with 5 RSVPs

[WarpSpeedDating @  
Science Museum, London](#)  
Oct 20 with 19 RSVPs



# NerdDinner Model



# NerdDinner ODBMS Implementations



# Android

- **What is it?**
  - Java-based open-source mobile OS.
- **Benefits**
  - Direct UI Binding for Collections
  - Event-driven UI architecture
- **Drawbacks**
  - No automated object binding for non-list controls
- **Shortcuts**
  - Custom Adapters for Model-to-UI binding
  - ListAdapter, ArrayAdapter
- **Challenges**
  - No framework support for advanced UI patterns.
  - Poor UI designer often forces editing XML by hand.



# Android

- A Custom adapter to bind the View data

```
private void createAdapter() {  
    adapter = new DinnerAdapter(mDinner, this);  
}
```

- But requires a lot of manual binding

```
public Dinner finishEditActivity() {  
    dinner.setName(editActivity.getTitleText().getText().toString());  
    Date date = makeDate(editActivity.getDateDinner().getText().toString(),  
        editActivity.getTimeDinner().getText().toString());  
    dinner.setEventDate(date);  
    dinner.setDescription(editActivity.getDescText().getText().toString());  
    dinner.setHostedName(editActivity.getHostnameText().getText().toString());  
    dinner.setContactInfo(editActivity.getDinnerCInfoText().getText().toString());  
    dinner.setAddress(editActivity.getDinnerAddressText().getText().toString());  
    dinner.setCountry(getCountry(editActivity.getDinnerCountry().getSelectedItemPosition()));  
    dinner.setLatitude(AndroidND.bookmark.getLatitude());  
    dinner.setLongitude(AndroidND.bookmark.getLongitude());  
    dinner.setSatellite(AndroidND.bookmark.isSatellite());  
    dinner.setTraffic(AndroidND.bookmark.isTraffic());  
    dinner.setZoomLevel(AndroidND.bookmark.getZoomLevel());  
    // Store dinner object  
    dinnerProvider.store(dinner);  
    return dinner;  
}
```




# Android

3G 11:56 AM

View Dinner

## Beach Dinner



When  
09-19-2010 00:31

Where  
450-456 Green St, San Francisco, 94133, California


Description  
You are all invited

Organizer  
German Viscuso

RSVP for event

3G 11:59 AM

## NerdDinner for Android



Find Dinner

Add Dinner

Dinners

Map Mode

Settings

DB Service



# Spring Web Flow

- What is it?
  - Web Application Framework based on Spring
  - Stateless request/response framework
- Benefits
  - Designed to work with POJOs
  - Lightweight Deployment
- Drawbacks
  - Breaks referential identity, doesn't reuse object instances, stateless
  - Designed to work with Hibernate
- Shortcuts
  - Use Spring BeanUtils or Dozer to bind stored instances
  - Metadata class annotations for input validation
- Challenges
  - Without the above, Spring gives back objects, but no binding
  - Breaks referential identity. (**Disconnected Objects**)



# Spring Web Flow

- Controller receives a request for the page.
  - Retrieve Model from the Database.
  - Use a custom ModelMapper to map the View-Model.
  - Return JSON-encoded View-Model to the user.

```
@RequestMapping(value = "dinner-{id}.json", method = RequestMethod.POST)
@ResponseBody
public DataContainer<DinnerViewModel> editDinner(@PathVariable int id) {
    final Dinner dinner = logic.byIdOrNew(id);
    return DataContainer.createUnEncoded(DinnerMapper.toViewModel(dinner));
}
```



# Spring Web Flow

- Custom Mapper to convert between Model and ViewModel

```
public static Dinner fromViewModel(DinnerViewModel data, Dinner dinner) {  
    // Because the source is a new instance of a view model, we need to copy all the properties.  
    BeanUtils.copyProperties(data, dinner, new String[]{"eventDate"});  
    // Type converter from input string to a date.  
    dinner.setEventDate(parseDate(data.getEventDate(), data.getEventTime()));  
    return dinner;  
}
```

```
public static DinnerViewModel toViewModel(Dinner dinner) {  
    DinnerViewModel viewModel = new DinnerViewModel();  
    BeanUtils.copyProperties(dinner, viewModel, new String[]{"eventDate"});  
    return viewModel;  
}
```



# Spring Web Flow

- Uses modern AJAX UI library (ExtJS) to render UI and handle Request/Response UI Mapping to View-Model





# Spring Web Flow

- Controller receives a View-Model update from the page.
  - Receives JSON-encoded View-Model.
  - The View-Model validates itself. (hibernate-validator)
  - Get the current Model instance from the Database.
  - Updates the Model instance with the ModelMapper.
  - Stores the changes.

```
@RequestMapping(value = "saveDinner.json", method = RequestMethod.POST)
public ModelAndView saveDinner(@Valid DinnerViewModel data, BindingResult result) {
    if (this.userLogic.knowsUser() && !result.hasErrors()) {
        final Dinner dinner = logic.byIdOrNew(data.getId());
        dinner.setHostedByIdentity(userLogic.loggedInUser());
        DinnerMapper.fromViewModel(data, dinner);
        logic.store(dinner);
    }
    return list();
}
```



# Spring Web Flow



Logged in as <http://gamlerhart.myopenid.com/> [Log out](#)

Find Dinner

Host Dinner

About



### RSVP

Your Name:

#### Attending to:

Title: Sweden Dinner

Date: 09/30/2010

Time: 20:09

Description:

Host Name: Swen

Contact Info:

Address: Stockholm

Country: Sweden

### Host Dinner

Title: Hongkong Dinner

Date: 09/27/2010

Time: 20:09

Description:

Host Name:

Contact Info:

Address: Hongkong

Country:

Submit



# Swing

- **What is it?**
  - Java Desktop Application Framework
- **Benefits**
  - Cross-Platform UI Implementation
  - Write-Once, Run Anywhere, 100% Java, No Dependencies
- **Drawbacks**
  - Rendered in Java, not a “real [OS] application”.
- **Shortcuts**
  - BeansBinding: Utility Library
  - SwingX: Intelligent UI Components
- **Challenges**
  - BeansBinding is dynamically typed, string-based, fails silently
  - Swing without a binding library requires a lot of extra work.



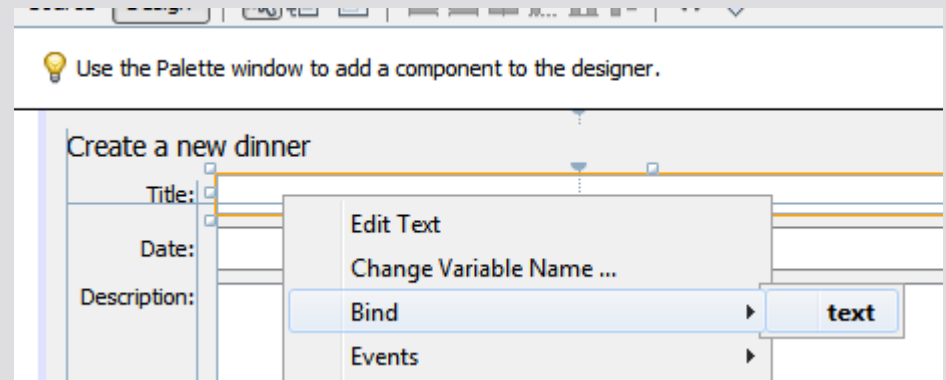
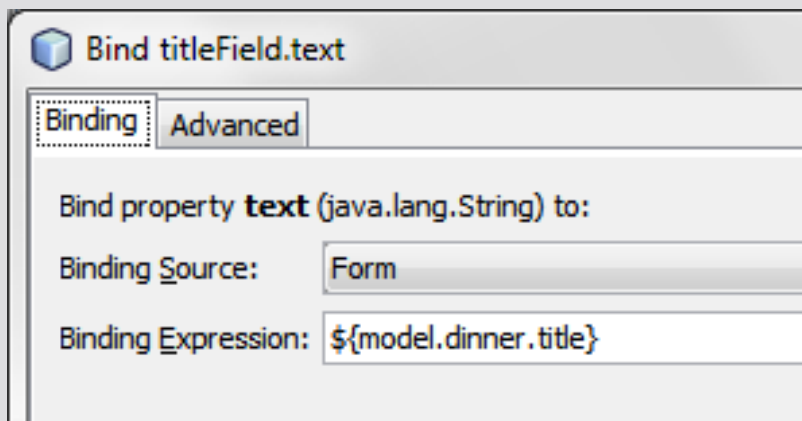
# Swing

## ■ Binding a form with the NetBeans form designer:

```
public EditDinnerModel getModel() {  
    return model;  
}  
  
public void setModel(EditDinnerModel model) {  
    this.model = model;  
}
```

Create a get/set method to handle the ViewModel.

Place a control on the form, choose Bind from the context menu.



The Expression should start with the get/set method name from step 1.



# Swing

- Manually setting up binding in the form code:
  - Normally automatically generated by the form designer.

```
// Create a two-way binding from the property 'eventDate' on the model  
// to the event-date components 'date' property.  
AutoBinding dateFieldBinding = Bindings.createAutoBinding(  
    AutoBinding.UpdateStrategy.READ_WRITE,  
    getModel(), ELProperty.create("${dinner.eventDate}"),  
    this.txtEventDate, BeanProperty.create("date"));  
// Add a required field validator.  
dateFieldBinding.setValidator(notEmptyValidation);  
// Specify a converter between date and string.  
dateFieldBinding.setConverter(new TextToDateConverter());  
dateFieldBinding.bind();
```



# ASP.NET WebForms

- **What is it?**
  - Microsoft .NET web application framework
- **Benefits**
  - Component oriented
  - Virtually stateful
- **Drawbacks**
  - State is sometimes serialized, complicated, bloated
- **Shortcuts**
  - ObjectDataSource
  - LinqDataSource
- **Challenges**
  - The (lack) of real page state cause disconnected objects.



# ASP.NET WebForms

- Automatically create and retrieve data from a DataContext

```
<asp:LinqDataSource
  ContextTypeName="ExampleDataContext"
  TableName="Products"
  Where="Price > 50"
  ID="LinqDataSource1"
  runat="server">
</asp:LinqDataSource>
```

- Manually select and return data to the LinqDataSource

```
protected void LinqDataSource_Selecting(object sender, LinqDataSourceSelectEventArgs e)
{
    var cities = from city in citiesArray
                 where city.CompareTo("B") > 0
                 select city;
    e.Result = cities;
    // Or we could set e.Result = citiesArray to return all rows.
}
```



# ASP.NET MVC

- What is it?
  - Stateless web application framework for Microsoft .NET
- Benefits
  - Excellent support for objects, validation, formatting and templating
  - Convention patterns are easy to follow and clearly defined
- Drawbacks
  - A lot of the framework is *By Convention*
- Shortcuts
  - `System.ComponentModel.DataAnnotations`
    - customize the templated view helpers and field html helpers
    - model validation
  - `ModelBinder`, `ValueProvider`
- Challenges
  - So much is by convention, it's a lot to learn



# ASP.NET MVC

- Updating the database with a POCO:

```
// Validate the view model
if (ModelState.IsValid) {
    // MVC is stateless, so get the Dinner we are modifying from the DB.
    Dinner dinner = dinnerRepository.GetDinner(id);

    // populate my current instance with updated form data
    UpdateModel(dinner);

    dinnerRepository.SaveChanges();
}
```



# .NET WinForms, WPF and Silverlight

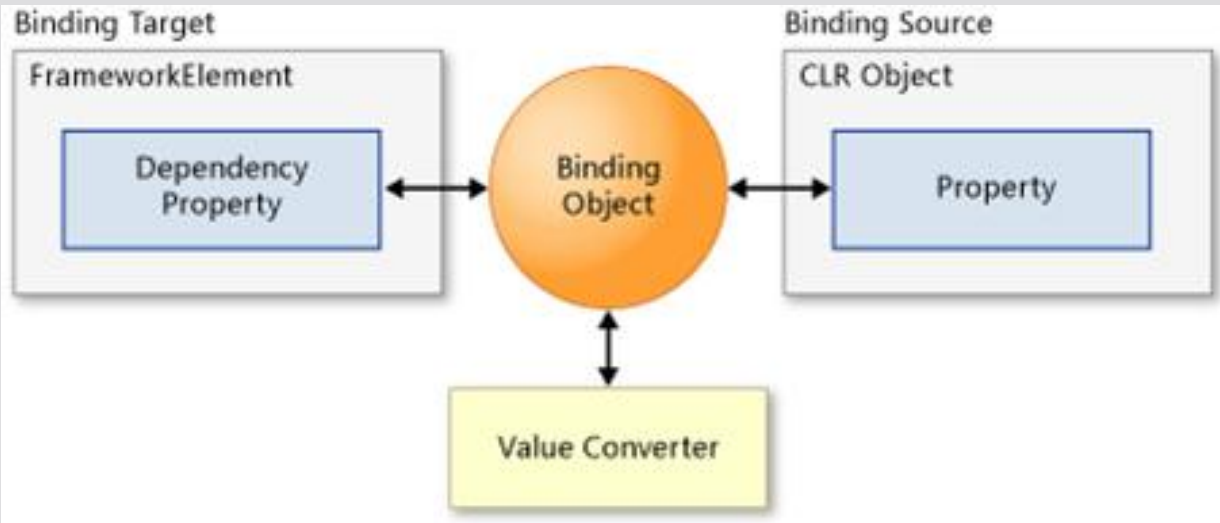
- What is it?
  - Windows application frameworks for Microsoft .NET 4.0
- Benefits
  - Easy Windows applications.
  - WPF allows easily designed rich client applications in XML.
- Drawbacks
  - Single platform...Windows, unless you use Silverlight
- Shortcuts
  - *System.Windows.Data* namespace
    - BindingOperations
    - Binding
    - IValueConverter
    - IDataErrorInfo
- Challenges
  - New, much improved in .NET 4.0
  - Silverlight includes only a subset of Binding and Templating abilities of WPF.



# .NET WinForms, WPF and Silverlight

- Assisted binding with *System.Windows.Data* Bindings class:

```
//make a new source  
MyData myDataObject = new MyData(DateTime.Now);  
Binding myBinding = new Binding("MyDataProperty");  
myBinding.Source = myDataObject;  
BindingOperations.SetBinding(myText, TextBlock.TextProperty, myBinding);
```



# Useful db4o Tips



# db4o & Transactions

- **Local Single Transaction Context**
  - `Db4oEmbedded.OpenFile()`
  - Best for Single-User applications like Android, Desktop
- **Local Multiple Transaction Contexts**
  - (Does not require the ClientServer library)
  - `Db4oEmbedded.OpenFile()`
  - `anyObjectContainer.OpenSession()`
  - Best used for single-server, multi-user applications
- **Networked**
  - (Requires db4o C/S library)
  - `Db4oClientServer.OpenServer()`
  - `Db4oClientServer.OpenClient()`

- In platforms with disconnected objects, add an ID field to your model classes for easy reference.
  - ASP.NET MVC
  - WebFlow
- When updating stored instances, retrieve the current object, update its values, store it back again.
- Use a new ObjectContainer for each request, but reuse it for as long as possible. Avoid re-opening another ObjectContainer unnecessarily.
- Don't leave your ObjectContainer client sessions to be GC'd. Close them!



# db4o Server Implementations

- 1.** Download our sample servers.
  - 2.** Add a reference to your model & NQ classes.
  - 3.** Compile and deploy.
- Db4o requires your class model loaded locally for many operations.
    - e.g. un-optimized queries (complex query edge cases)
  - .NET Native for Windows
  - Java Native
    - Windows, Linux, Mac OS X

**Download Everything:**  
**[www.ObjectDinner.com](http://www.ObjectDinner.com)**