

# Inside Web Objects

---

## Getting Started With Direct to Java Client



May 2001

🍏 Apple Computer, Inc.  
© 2001 Apple Computer, Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Computer, Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Computer, Inc. Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this book. Apple retains all intellectual property rights associated with the technology described in this book. This book is intended to assist application developers to develop applications only for Apple-labeled or Apple-licensed computers

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Computer, Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Mac, Macintosh, and WebObjects are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Enterprise Objects and Enterprise Objects Framework are trademarks of NeXT Software, Inc., registered in the United States and other countries.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

Java is a registered trademark of Sun Microsystems, Inc. in the United States and other countries.

Simultaneously published in the United States and Canada

**Even though Apple has reviewed this manual, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD "AS IS," AND YOU, THE PURCHASER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS MANUAL, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

<b>Chapter 1</b>	<b>Introduction</b>	<b>5</b>
<hr/>		
	About This Book	5
<b>Chapter 2</b>	<b>Getting Started With Direct To Java Client</b>	<b>7</b>
<hr/>		
	Creating a Direct to Java Client Project	7
	What's in a Direct to Java Client Project?	9
	Building and Running the Application	9
	If the Client Application Doesn't Start	10
	If the Application Has No Windows	11
	Examining the Application	11
	Main, Enumeration, and "Other" Entities	14
	Main Entities	14
	Enumeration Entities	14
	"Other" Entities	15
	Customizing the Application	15
	The Assistant	16
	Configuring Entities	16
	Configuring Properties	18
	Configuring Widgets	20
	Typical Workflow	22
	Examining the Panes of the Query Window	23
	Configuring the Studio Pane of the Query Window	23
	Configuring the Customer Pane of the Query Window	23
	Configuring the Unit Pane of the Query Window	24
	Configuring the Video Pane of the Query Window	25
	Examining Form Windows	25
	Configuring the Customer Form Window	25
	Configuring the Unit Form Window	26
	Configuring Windows	27
	Changing the Title of the Query Window	27
	Other Assistant Settings	28

# C O N T E N T S

# Introduction

---

Direct to Java Client dynamically generates the user-interfaces for applications or for parts of them. The advantage of this technology is that it's not necessary to write source-code to generate or manage an application's user interface. This allows developers to focus on the business logic of an application, instead of on the user interface.

## About This Book

---

This document provides an overview of the applications you can create. It takes you step by step through the process of creating, building, and running a Direct to Java Client application, as well as through the process of customizing an application using the Direct to Java Client Assistant.

**Note:** There are other methods of customizing a Direct to Java Client application besides using the Assistant, but they are more advanced and are not covered in this document.

Before you begin doing this tutorial, you should be familiar with the Java Client technology. If you're not, work through the *Java Client Tutorial* to acquaint yourself.

This tutorial uses a sample OpenBase database, WOMovies, that contains information about movies and a video rental store. OpenBase and the sample WOMovies database come with WebObjects and are used by the WebObjects and Enterprise Objects Framework examples. Neither requires any special installation or configuration.

# C H A P T E R 1

## Introduction

# Getting Started With Direct To Java Client

---

This chapter is divided in sections that show you how to develop and customize a Direct to Java Client application.

- “Creating a Direct to Java Client Project” (page 7) shows you how to use the Project Builder Assistant to build a Direct to Java Client application from a sample model file.
- “Building and Running the Application” (page 9) shows you how to use Project Builder to build and run the application. It describes how to start the client application and the Direct to Java Client Assistant.
- “Examining the Application” (page 11) describes the elements of a Direct to Java Client project. In particular, it describes how Direct to Java Client uses the model file to define the user-interface of the application.
- “The Assistant” (page 16) shows how you use the Direct to Java Client Assistant to customize your application.
- “Typical Workflow” (page 22) describes a general workflow that facilitates the process of customizing Direct to Java Client applications.

## Creating a Direct to Java Client Project

---

The Project Builder Assistant creates a Direct to Java Client project for you using a model you specify. For this tutorial, you use the Movies and Rentals models included in the example JavaBusinessLogic framework that’s installed with WebObjects.

1. Launch Project Builder.

### Getting Started With Direct To Java Client

The Project Builder application is located in the `/Developer/Applications` directory. Navigate to that directory and launch the application.

2. Create a new project.

Choose File > New Project.

3. Select the kind of project to create.

Select WebObjects/Direct to Java Client Application.

Click Next.

4. Name the project.

Name the project `D2JCTutorial`.

Click Set and choose a location for the project.

Click Next.

5. Select the frameworks to use.

The Assistant includes the frameworks necessary for Direct to Java Client applications. For the tutorial, you add the `JavaBusinessLogic` framework. It provides client- and server-side business logic implemented in custom enterprise objects.

Click Add.

Navigate to the `/Library/Frameworks` directory.

Select `JavaBusinessLogic.framework` and click Choose.

Click Next.

6. Select the model(s) that the project is to based upon.

Direct to Java Client technology uses model files to generate applications. To make models available to Direct to Java Client, you add them to your project. You can add models directly to your project by adding them to the Resources group, or you can add them indirectly by adding frameworks that contain your models.

The `JavaBusinessLogic` framework that you added in the previous step already includes the appropriate models, `Movies.eomodeld` and `Rentals.eomodeld`.

Click Next.

7. Build the project.

## Getting Started With Direct To Java Client

At this point the Assistant has all the information it needs to create the project. It offers to build and launch the application for you. However, in this tutorial you do that manually.

Deselect “Build and launch project now” and click Finish.

## What’s in a Direct to Java Client Project?

---

The project that the Project Builder Assistant created for you is very similar to a normal Java Client project. It has the typical Application, Session, and Main classes for the Server target. The Client target, however, has no files associated to it. It has no interface controller class or nib file, because Direct to Java Client applications generate their user interfaces dynamically.

The project’s `Main.wob` component contains a `WOJavaClientApplet` that is configured in a special way. It’s `width` and `height` are set to 0, and the `applicationClassName` is set to “`com.webobjects.eogeneration.EODynamicApplication`”. The application class is important because it’s responsible for initiating the user interface generation process.

The Resources group of the project contains a `user.d2wmodel` file. The Direct to Java Client Assistant uses this file to store user interface configuration information.

## Building and Running the Application

---

Building and running the application is just like building and running any other Java Client application. Simply perform the following steps.

1. Build the application.

Click  in Project Builder’s main window.

2. Set execution options.

Click on the Targets tab.

Click the D2JCTutorial target in the Targets list.

Click the Executables tab on the content area.

## Getting Started With Direct To Java Client

Click Add on the Arguments panel.

On the text box type `-WOPort 8888 -WOAutoOpenInBrowser NO`. These arguments facilitate the development process.

- **WOPort:** Specify any port you want. It is not necessary to specify a port, but it is convenient. By specifying a port, the application URL (used to start the client application) doesn't change between executions. If the application URL is always the same, you can start the client application with the same command each time.
- **WOAutoOpenInBrowser:** Specify `NO` so the client application is not automatically launched in a browser. Java Client applications should be run as Java applications.

3. Start the server application.

Click  in Project Builder's main window.

4. Start the client application.

In a shell, navigate to the `D2JCTutorial/build/D2JCTutorial.woa/Contents/MacOS` directory.

Execute the script `D2JCTutorial_Client` with the HTTP address of the server application. You can copy it from the output that Project Builder produced when it launched the server application. The launch command should be similar to the following:

```
./D2JCTutorial_Client http://eBruce.apple.com:8888
```

## If the Client Application Doesn't Start

Verify that your application's URL is correct. Check the messages in the Run panel. At the bottom, you'll see the following messages:

```
The URL for webserver connect is:
http://eBruce.apple.com/cgi-bin/WebObjects/D2JCTutorial.woa/-8888
The URL for direct connect is:
http://eBruce.apple.com:8888/cgi-bin/WebObjects/D2JCTutorial
Waiting for requests...
```

Verify that the application URL you use to start the application has the same host name and port number as the one in the Run panel's output.

## If the Application Has No Windows

---

Verify that the project includes the JavaBusinessLogic framework. Direct to Java Client dynamically generates an application based on model files. In this application, the model files come from the JavaBusinessLogic framework.

## Examining the Application

---

This section illustrates the application's functionality by guiding you through the steps to search for records, modify records, and add new ones.

When the application starts, it opens a window with which you can search for database records. Each of the tabs in the Query Window corresponds to an entity in one of the application's models. Not all of the entities in the Movie and Rentals models have tabs in the Query Window. The way Direct to Java Client chooses which entities to represent in the Query Window is described in the section “[Main, Enumeration, and “Other” Entities](#)” (page 14).

1. Focus the Query Window on the Movie entity.

In the Query Window, click the Movie tab. The Query Window switches its contents to a user interface for searching for Movies.

Note that the default Direct to Java Client application allows you to search on `title` and `plotSummary`. You can easily configure the application to search on different attributes using the Direct to Java Client Assistant (Assistant for short). In later sections, you are shown how to change the search attributes.

2. Search for Movies whose titles begin with the letter “A”.

Type `A` in the Title text box, and click Find. The application lists the Movies meeting the search criteria.

Note that the application lists `title`, `category`, `dateReleased`, `posterName`, `revenue`, `trailerName`, and `plotSummary` for the Movies in the result set. As with the search attributes, you can easily configure the application to display different attributes in the table, which you do later.

3. Search for Movies whose titles begin with the letter “A” or “B”.

## Getting Started With Direct To Java Client

Click Clear. Although the “A” is cleared from the Name field, the Movies in the results table remain. To add Movies that start with the letter “B” to the results, type B in the Name text box, and click Append. Now the results table contains Movies that start with the letter “A” and the letter “B”.

The difference between Find and Append is this: Find replaces the records in the results list with the results of the most recent search. Append adds data from the most recent search to the records already in the results table. So using Append is a way to add the results of more than one query (for example, Movies whose titles start with “A” and Movies whose titles start with “B”).

#### 4. Open a Movie.

Select a Movie in the result list and click Open, or simply double-click the Movie’s title. Either action opens the Movie in a separate window that displays its attributes and relationships. The attributes—`title`, `category`, `dateReleased`, `posterName`, `revenue`, and `trailerName`—are in the top part of the window. You can edit any of the Movie’s attributes here. After making changes, you can save them (click Save at the top of the Movie window) or revert back to the original values (click Revert).

The Movie’s relationships are in the bottom part of the window. One relationship—`studio`—is displayed by itself in the middle of the window while the others are displayed below in a tab view. The reason for this is explained later in the section “Main, Enumeration, and “Other” Entities” (page 14).

In the Studio area of the Movie window, you can change the Movie’s Studio (Select), open the Movie’s Studio (Open), or unassign the Movie’s Studio (Deselect). Clicking the Open button simply opens the Movie’s Studio in a Studio window that is similar to the Movie window in that it displays Studio attributes and relationships. Clicking Deselect unsets the Movie’s `studio` relationship so that the Movie has no Studio.

#### 5. Select a new Studio.

In the Studio area of the Movie window, click Select. A dialog opens for searching for Studios. Type search criteria (A in the Name text box, for example), and click Find. Choose a Studio from the result list and click OK. The dialog closes, and the Movie’s Studio is changed from the original to the one you selected.

Note that the Movie window now has an asterisk (“\*”) in the title bar. The asterisk indicates that the Movie has changes that haven’t been saved. If you wanted to save the change, you would click Save. In this case, however, click Revert to discard the change.

## Getting Started With Direct To Java Client

## 6. Examine the relationship tabs at the bottom of the window.

The relationships for the Plot Summary and Voting tabs—`plotSummary` and `voting`—are to-one relationships. Their respective panes allow you to modify the destination objects directly. For example, if you change the contents of the Summary text box in the Plot Summary pane and click Save, you change the PlotSummary object's `summary` value and commit the change to the database.

The other relationships—`directors`, `reviews`, and `roles`—are to-many relationships. Their panes allow you to add destination objects to the relationships and to delete objects from them. Additionally you can open destination objects (in their own window) and edit them.

## 7. Create a new Movie.

In the Movie window, click New. Alternatively, if the Query Window's Movie tab is still selected, you can click New in the Query Window. Either action creates a new Movie window into which you enter information about the new Movie. Fields with blue titles are required; that is, you can't save the new Movie if the blue titled fields are empty.

Fill in the Movie's attributes as follows:

- **Title** Evil Dead 2
- **Category** Horror
- **Date Released** 4/1/1987

Click Save to save the new Movie.

## 8. Set the Movie's Studio to Elite Entertainment.

In the Studio area of the window, click Select. In the Studio dialog, type E in the Name text box, and click Find. No results are returned, so you have to create the Elite Entertainment Studio.

Click New in the Studio dialog. The first dialog closes, and a new Studio dialog opens for creating a new Studio. Name the Studio "Elite Entertainment" and provide a budget. Click Save.

The new Studio is saved to the database and assigned as the Studio for the *Evil Dead 2* Movie.

You can add directors and movie roles (and their actors) in a similar manner.

## Main, Enumeration, and “Other” Entities

---

Direct to Java Client divides entities into three types: main, enumeration, and other. Each of an application’s entities is exactly one of these types.

### Main Entities

---

A *main* entity is generally a top-level entity that users work with most frequently. Consequently, Direct to Java Client creates a tab for each main entity in the Query Window and provides form windows for editing each of the main entities. For example, in this tutorial application, the main entities are Customer, Movie, Studio, Talent, Unit, User, and Video.

Direct to Java Client by default defines a main entity as one that is not the destination of any relationships that

- propagate primary key
- own destination
- use the cascade delete rule

### Enumeration Entities

---

Direct to Java Client also defines the concept of an *enumeration* entity. An enumeration entity by default is an entity that conforms to the conditions for main entities and additionally conforms to the following conditions:

- The entity has fewer than five attributes.
- The entity has no relationships that are mandatory.
- All the entity’s relationships use the deny delete rule.

In practice, enumeration entities should define a collection of values that represent a list of choices. For example, in this tutorial application, the enumeration entities are FeeTypes and RentalTerms.

The values in enumeration entities are usually fairly static, and you usually don’t want a complex user-interface for changing them. Consequently, Direct to Java Client applications provide a single window for editing enumeration values, the Enumeration Window.

Choose Tools > Enumeration Window.

## Getting Started With Direct To Java Client

This opens the Enumeration Window containing a tab for each of the application's enumeration entities. The tab for a particular entity shows the complete set of values in that entity (that should be a small number of values). You can add a new value to the enumeration's collection (Add), delete a value from the collection (Remove), and modify a value (simply make the changes and click Save).

By displaying enumeration values in this special window, an application doesn't clutter the Query Window with tabs for enumeration entities. This approach simplifies the application's user interface in other ways, as well. The application doesn't provide form windows for enumeration entities, and relationships to enumeration entities can be represented simply with combo boxes (for to-one relationships) and pick lists (for to-many relationships) instead of with tables (to-ones) and select dialogs (to-manys).

### “Other” Entities

---

Entities that aren't main or enumeration entities are simply *other* entities. For example, in this tutorial application, the *other* entities are CreditCard, Director, Fee, MovieRole, PlotSummary, Rental, Review, TalentPhoto, and Voting.

*Other* entities can be manipulated through the master-detail user interfaces of main entities. This explains why a Movie's `studio` relationship is displayed by itself in the middle of the window while the other relationships are displayed below in a tab view. The destination entity of the `studio` relationship, Studio, is a main entity. Main entities have their own form windows with which you edit them. The destinations of the other relationships are *other* entities. Since they don't have their own form windows, they are edited in master-detail interfaces inside a main entity's form window.

## Customizing the Application

---

As you can see, the default application is fairly complete. Direct to Java Client uses a sophisticated set of rules to assemble the user interface. However, the default user interface isn't exactly what you want in every case.

In the next sections you make changes to the application, including the following:

- modify the set of main entities

### Getting Started With Direct To Java Client

- modify the search attributes in the Query Window, and specify the order in which the attributes are displayed. Similarly, modify the attributes displayed in the results table of the Query Window.
- specify the order of the attributes and relationships that can be edited in a form
- change the widget used to display and edit particular attributes

## The Assistant

---

The Direct to Java Client Assistant is an easy-to-use tool that performs the most common customizations to Direct to Java Client applications. It runs inside the client application so you can see and directly test changes while the application is running.

To open the Assistant, choose Tools > Assistant.

The Tools menu is available if the `E0AssistantEnabled` option on the server is set to YES. During production, you may not want the Assistant command to be available. You can specifically disable the Assistant using the `E0AssistantEnabled` user default.

The Assistant window has several tabs across the top: Entities, Properties, Widgets, Windows, Miscellaneous, and XML. The tabs allow you to configure the application in different ways. Each is discussed in later sections.

## Configuring Entities

---

The default set of rules that Direct to Java Client uses to identify main, enumeration, and *other* entities are a pretty good start. However, you might need to make some changes to the way Direct to Java Client classifies your application's entities. For example, this tutorial application doesn't use the User entity, so you need to remove it from the list of main entities.

### Getting Started With Direct To Java Client

1. Specify the User entity as an “Other” entity.

In the Assistant window, ensure that the “Entities” tab is selected. In the Main Entities list, select User. Click the arrow pointing to the left to move the User entity from the Main Entities list to the Other Entities list.

2. Rearrange the entities.

By default, the entities are ordered alphabetically. For example, in the Query Window, the Customer tab is first, then Movie, and so on. In this application, it makes more sense to group Movie, Talent, and Studio together and group Customer, Unit, and Video.

In the Main Entities list, select Movie. Click the arrow pointing up to move the Movie entity to the top of the list. Move Talent so it’s next in the list, then Studio. Customer, Unit, and Video, in that order, should be last in the list.

3. Apply the changes.

Click Apply. This updates the Assistant’s in-memory set of rules. The new rules don’t affect existing windows, but they are applied when Direct to Java Client generates new user-interfaces in the running client application.

To see that your changes have been recorded in the Assistant’s set of rules, choose Tools > New Query Window. The new Query Window that opens doesn’t have a tab for User, and the window’s tabs are ordered as you specified in step 2.

4. Restart the application.

Click Restart. This restarts the client application using the Assistant’s in-memory set of rules. In the new instance of the application, the Query Window immediately displays the changes you specified.

Since you can’t update an application’s existing user interface by applying changes, you can use Restart to reflect your changes throughout the entire application. If you don’t like the changes, you can revert them by clicking Revert.

5. Save your changes.

Click Save. This updates the set of rules saved in your project. Once you have saved the changes, you can no longer revert.

## Configuring Properties

---

The Assistant's Properties tab allows you to configure the way an application handles properties (attributes and relationships). For example, you can change the attributes that users search on in the Query Window and specify the order in which they appear.

1. Set the property keys for querying the Movie entity.

In the Assistant, click the Properties tab. The specification settings—the Question, Task, and Entity pull-down lists—should be set to <All>, query, and Movie, respectively. The settings are probably correct, but if any of the settings are different, change them. A simple way to set the Task and Entity pull-down lists is to select the Movie tab in the Query Window. The Assistant automatically updates to set the Task pull-down list to query and the Entity pull-down list to Movie.

Move `plotSummary` and `voting` to the Other Property Keys list. Move `category` and `studio` to Property Keys. Using the up and down arrows, order the properties as follows: `title`, `category`, `studio`.

Save the changes, and restart the application (from the Assistant). The Movie pane of the Query Window is reconfigured.

Note that a Studio Name field is now available. This is because you specified the `studio` relationship as a property key for the query task. When a to-one relationship (such as `studio`) is specified as a property key, Direct to Java Client uses *identify* property keys of the relationship's destination object to represent the relationship in the user interface.

2. Examine the identify property keys for the Studio entity.

In the Assistant, choose `identify` from the Task pull-down list and choose Studio from the Entity pull-down list. Observe that the Studio has one *identify* property key—`name`. This attribute is used to identify a particular Studio when it appears in the user interface as a destination of a to-one relationship.

## Getting Started With Direct To Java Client

One example of how the identify property keys are used is in the Movie pane of the Query Window. Movie's `studio` relationship is specified as a query property key. To represent `studio` as a property key in the Query Window, Direct to Java Client uses the identify property key (`name`) of the relationship's destination entity (Studio).

Another example is the Studio part of a Movie window, in which the Movie's Studio is identified by its `name`.

3. Modify the identify property keys for the Movie entity.

Choose Movie from the Entity pull-down list, keeping the Task as `identify`. Movie has one identify property key, `title`. Since it's quite common for movie remakes to use the same name as the original, a movie's title isn't always enough to identify it. Add `dateReleased` to Movie's list of identify properties, and save the change.

4. Configure the property keys for listing Movies.

The property keys for the list task are used in result lists such as the ones in the Query Window. When you search for Movies, for example, the Movies in the result set are listed in the Query Window's table. The result table has a column for the list property keys, `title`, `category`, `dateReleased`, `posterName`, `revenue`, `trailerName`, and `plotSummary`. The columns for these properties are crowded into the table, and many aren't particularly helpful to users.

**Note:** In addition to the property keys listed above, `voting` is also a list property key for the Movie entity. The `voting` relationship doesn't have a column in the results table. This is because Voting's identify property key is its `movie` relationship, which Direct to Java Client omits from the table view to avoid recursion.

To pare down the set of columns, choose list from the Assistant's Task pull-down list, and choose Movie from the Entity pull-down list (if it isn't selected already). Move `posterName`, `revenue`, `trailerName`, and `plotSummary` to the Other Properties column.

So users can verify that all the search criteria they specify is applied correctly, the list property keys should be a superset of the query properties. Since users can search for Movies by Studio, the list property keys should include `studio`, as well. Add `studio` to the list property keys.

Save the changes. To see them, open a new Query Window.

5. Configure the property keys for Movie form windows.

## Getting Started With Direct To Java Client

In the Query Window, click **New** to create a **Movie** form window. The Assistant updates to set the **Task** pull-down list to **form** and the **Entity** pull-down list to **Movie**.

Examine the form window. As is most commonly the case, the default property keys for the form task are correct for this application. In general, the form window should allow users to edit all of an entity's client class properties. However, you might want to reorder the properties so they appear in different locations in the form window.

Using the up and down buttons, order the properties as follows: `title`, `category`, `dateReleased`, `rated`, `revenue`, `posterName`, `trailerName`, `plotSummary`, `directors`, `roles`, `voting`, `reviews`, `studio`. Save the changes, and open a new **Movie** form window to see how they affect the form window layout.

## Configuring Widgets

---

The Assistant's **Widgets** tab allows you to configure the widgets used to display and manipulate properties (attributes and relationships). For example, you can change the widget that's used to display a particular attribute, and you can set the widget's resizing behavior.

1. Change the widget for the **PlotSummary** entity's `summary` attribute to a text area.

Click the **Widget** tab in the Assistant. Choose **form** from the **Task** pull-down list, **PlotSummary** from the **Entity** pull-down list, and `summary` from the **Property** pull-down list.

**WARNING**

Don't use **Movie** as the **Entity** and `plotSummary` as the property key. You're configuring the widget used to display the `summary` property of the **PlotSummary** entity, not the widget for displaying the `plotSummary` relationship of the **Movie** entity. So be sure the **Entity** pull-down list is set to **PlotSummary** and the **Property** pull-down list is set to `summary`.

Set the **Widget Type** to **EOTextAreaController**. Save the change, and open a new **Movie** form window to see the result.

### Getting Started With Direct To Java Client

#### 2. Remove the Summary label.

The `summary` Widget label is redundant with the PlotSummary tab label, so you should remove it.

If you opened a new Movie form window to see the change you made in step 1, you first need to restore the Assistant's Widget tab settings. Set the Task to `form`, the Entity to `PlotSummary`, and the Property to `summary`.

Now set the `summary` property's "Show Label Component" setting to `False`. Save the change, and open a new form to see it.

#### 3. Set sizing information for the `summary` widget.

Resize the window so it's much larger than the default size. Notice that the `summary` widget doesn't resize vertically. To prevent the waste of screen real estate and to display more of the summary upon window resizing, you need to set the widget to resize.

Restore the Assistant Task, Entity, and Property settings to `form`, `PlotSummary`, and `summary`, respectively. Now set the `summary` property's "Vertically Resizable" setting to `True`. Save the change, and open a new form to see it.

#### 4. Change the widget for the Review entity's `review` attribute to a text area.

In the Widget tab of the Assistant, set the Task to `form`, Entity to `Review`, and Property to `review`. Set the Widget Type to `EOTextAreaController`. Save the change.

#### 5. Set sizing information for the `review` widget.

In a new Movie form window, click the Reviews tab. Resize the window so it's much larger than the default size. Notice that the Review widget doesn't resize vertically. In this window, this behavior is appropriate because the table above the Review widget does resize to take up the added area.

However, you might want a larger area in which to display movie reviews. If you set the text area to resize vertically, the widget still doesn't resize because the table view above it does. So to provide a larger text area, you can set a minimum height.

Restore the Assistant Task, Entity, and Property settings to `form`, `Review`, and `review`, respectively. Now set the `review` property's `Minimum Height` to `200`. Save the change, and open a new form to see it.

## Typical Workflow

---

The most natural way to customize an application's property keys and widgets is to examine and modify one window at a time, as outlined by the steps below:

1. With the Assistant open, click the first tab in the Query Window.
2. Examine the search properties for the current entity. Remove any fields necessary, add others, and then order the property keys as you want them to appear in the Query Window.
3. Choose `list` in the Assistant's Task pull-down list, and modify the list property keys. Since the Query Window displays list properties as well as query properties, you can see what changes you need to make.
4. Click the next tab in the Query Window. The Assistant automatically updates to track your movements in the application. This saves you setting the Entity pull-down list in the Assistant.
5. Repeat steps 2 through 4, until you've set the query and list property keys for all the main entities.
6. Click the first tab in the Query Window.
7. Click `New` to open a form window for the current entity. Make necessary changes to the entity's form property keys.
8. Make any necessary changes to property keys of any other entity that appears in the current entity's form window. To do so, choose `list` or `identify` from the Assistant's Task pull-down list, and set the Entity pull-down list to the other entity.
9. Click the next tab in the Query Window.
10. Repeat steps 7 and 9.

The following sections demonstrate this approach. Before continuing, open the Assistant if it isn't open already, and select the Properties tab.

## Examining the Panes of the Query Window

---

You've already configured the Movie pane of the Query Window, so examine the next pane, Talent. The window allows you to search on a talent's first and last names, and it shows the first and last names in the results table. This is fine, so move on to the next pane, Studio.

### Configuring the Studio Pane of the Query Window

---

The Studio pane allows you to search for studios by their name and budget. It seems unlikely that users want to search on budget.

Remove `budget` from the query property keys list.

The results table shows the studio name and budget. This seems fine, so move on to the next pane.

### Configuring the Customer Pane of the Query Window

---

The Customer pane allows you to search for customers by their city, first name, last name, credit card number, and credit card authorization. Users should not look up customers by credit card information, so you should remove it.

1. Remove `creditCard` from the query property keys.

The `creditCard` property is a to-one relationship that Direct to Java Client expands to the identify property keys (`cardNumber` and `authorizationNum`) of the relationship's destination entity (a `CreditCard`). By removing `creditCard` from the property keys, you remove the `cardNumber` and `authorizationNum` fields from the query interface.

2. Add `memberSince` and `phone` to the property keys.

Two other Customer properties, `memberSince` and `phone`, seem like reasonable attributes to query on, so add them.

3. Order the keys as follows: `firstName`, `lastName`, `city`, `phone`, `memberSince`.
4. Modify the list property keys.

The results table is far too crowded with properties to be useful, so pare down the set of list property keys. Remove `state`, `streetAddress`, `zip`, and `creditCard` from the list property keys. Order the remaining keys as follows: `firstName`, `lastName`, `city`, `phone`, `memberSince`.

### Getting Started With Direct To Java Client

Move on to the next pane.

### Configuring the Unit Pane of the Query Window

---

The Unit pane allows you to search for units by unit ID, date acquired, movie name, and movie release date. It seems unlikely that users want to search for particular units by movie release date, so remove it.

Note that the query property keys for Unit are `unitID`, `dateAcquired`, and `video`. Through the `video` key, Direct to Java Client gets both a movie name and movie release date, which correspond to the Movie `identify` property keys. You want to keep the movie name in the interface, but not the movie release date.

1. Remove `video` from the property keys.

Using the `video` key, you can't get one `identify` Movie property without the other one, so remove it. You'll add the Movie `name` property to the query interface another way.

2. Add `video.movie.title` to the property keys.

Type `video.movie.title` in the Additional Property Key Path text box at the bottom of the Properties pane, and click Add.

3. Save and restart to observe these changes.

Adding key paths to an entity's property keys flattens attributes into the entity for display purposes: the attribute identified by the key path is treated in the user-interface as if it's one of the entity's own attributes rather than an attribute of a related entity.

4. Change the label for `video.movie.title`.

The label for `video.movie.title`, "Video Movie Title", seems unnecessarily wordy. "Movie Title" is better.

Click the Widgets tab in the Assistant. Set the Task pull-down list to `<ALL>` if it isn't already, then choose `video.movie.title` from the Property Key pull-down list. In the Label field, delete the word "Video" so the label is simply "Movie Title".

5. Save and restart to observe this change.

### Getting Started With Direct To Java Client

Notice that the movie title label is now “Movie Title” for both the search field and for the column in the results table. To set one, but not the other, you would simply set the Task pull-down list to the appropriate task before making the change.

#### 6. Examine Unit’s list properties.

The `notes` property contains free-form text, and doesn’t display well in a list, so remove `notes` from Unit’s list property keys.

Move on to the next pane.

### Configuring the Video Pane of the Query Window

---

Video’s query interface has the same problem as Unit’s. It seems unlikely that users want to search for videos by movie release date. Remove it from the query interface the same way you did for Unit. That is, remove the `movie` relationship from the property keys and add `movie.title`.

## Examining Form Windows

---

Now that you’ve configured the panes of the Query Window, move on to the form windows. You’ve already configured Movie’s form window, so start with Talent.

Open a Talent form. It looks fine, so move on to Studio. It, too, looks fine, so move on to Customer.

### Configuring the Customer Form Window

---

All the Customer properties are property keys. This is fine; users need to be able to edit all of a customer’s data. However, the properties are ordered poorly. A customer’s name should come first, and the components of a customer’s address should be grouped together and ordered so they appear as a proper address.

1. Order the Customer form property keys as follows: `firstName`, `lastName`, `memberSince`, `phone`, `streetAddress`, `city`, `state`, `zip`, `creditCard`, `rentals`.
2. Examine the Credit Card pane at the bottom of the Customer window.

The property keys are poorly ordered, and the “Num” in the “Authorization Num” label should be spelled out.

## Getting Started With Direct To Java Client

- Using the Properties Assistant, order CreditCard's form property keys as: `cardNumber`, `expirationDate`, `limit`, `authorizationNum`, `authorizationDate`, `customer`.

Note that `customer` is a to-one relationship from CreditCard to Customer. Ordinarily, a to-one relationship is represented in a form, but `customer` isn't represented here. That's because Direct to Java Client filters the `customer` property key out of the user interface rather than display it recursively. Since the CreditCard form is contained within a Customer form, Customer information is already displayed. This feature is called *entity hierarchy filtering*.

**WARNING**

Be careful about removing relationships from the set of property keys. Even though a relationship isn't necessary for a window you're looking at, it might be needed by another window.

As an example, consider the MovieRole entity, which has to-one relationships to Movie and Talent. Similarly, Movie and Talent have to-many relationships back to their MovieRoles. Since MovieRole is an *other* entity, MovieRole records are manipulated in a master-detail user interface in Movie and Talent form windows. In a Movie form window, the interface for displaying and setting a MovieRole's value uses only the `roleName` attribute and the `talent` relationship. Entity hierarchy filtering prevents the interface from using the `movie` relationship.

When you're looking at the Movie form window, you might consider removing MovieRole's `movie` relationship from the form task's property keys. It's filtered out, but you might think of removing it anyway for clarity. However, if you remove it, the `movie` relationship won't be displayed in the Talent form window either. In the Talent window a MovieRole's `movie` relationship is important information.

- In the Widgets pane of the Assistant, change the `authorizationNum` label to "Authorization Number".
- Examine the Rentals pane.

The Rentals pane looks OK, so move on to the next form window.

## Configuring the Unit Form Window

---

The form property keys for Unit are correct, and their order is fine, too. However, the notes property should be in a text area instead of a field.

### Getting Started With Direct To Java Client

1. Change the `notes` widget to a text area.

Click the Widgets tab in the Assistant. Set the Task pull-down list to form, and choose `notes` from the Property Key pull-down list. Set the Widget Type to `EOTextAreaController`. Save the change, and open a new Unit form window to see the result.

2. Examine the Rentals area of the window.

You need to resize the windows to read the Rental table columns. Each Rental line displays Customer identify properties: `city`, `firstName`, and `lastName`. The `city` attribute isn't really necessary, and it would be more helpful if a customer's last name were to appear first.

3. Modify Customer's identify property keys.

Remove `city` from Customer's identify property keys. Order `lastName` first. Save your changes.

Move on to the next form window, Video. It's fine.

## Configuring Windows

---

After configuring properties and widgets, the bulk of your Assistant customizations are done. The final change you make to your application in this tutorial is to change the title of a window. To do this, you use the Windows tab of the Assistant.

The Windows tab allows you to do things such as

- setting the title of windows
- changing the default position and size of a window
- determining whether to save window position and size in user defaults
- specifying disposal and reuse behavior to tune performance

### Changing the Title of the Query Window

---

To change the title of the Query Window to "Find", perform the following steps.

### Getting Started With Direct To Java Client

1. Click the Windows tab in the Assistant.
2. Click the Query Window.

This focuses the Assistant on the Query Window, setting the Question to window, setting Task to `queryWindow`, and disabling Entity (because the Query Window displays multiple entities).

3. In the Label field, type `Find`.
4. Save your changes and restart the application.

The Query Window's title is now "Find".

## Other Assistant Settings

---

In addition to Entities, Properties, Widgets, and Windows tabs, the Assistant also provides Miscellaneous and XML tabs. The Miscellaneous tab allows you to specify application wide default settings for alignment and label placement (for all widgets in all windows for all entities).

The XML tab is more advanced. It allows you to view the output of the rules system. The output, which is the resolution of rules, is represented as XML. As you become more familiar with the Direct to Java Client technology, you can read the XML to verify user interface configurations instead of restarting the application to see the actual effects of your changes.

You can also use the XML tab to save the XML to a file. You typically do this when you want to customize an application by *freezing* its XML code. Freezing allows you to explicitly state the result of a rule system request. The freezing technique is beyond the scope of this tutorial.