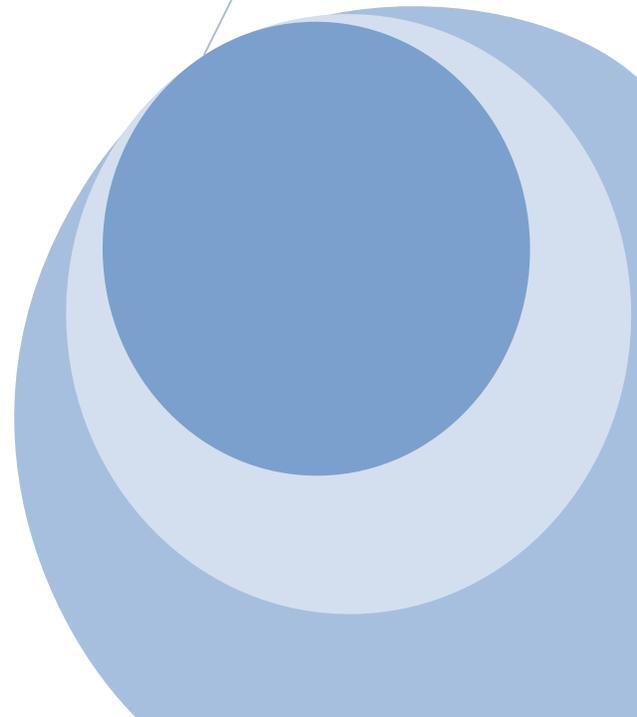


MAX 2009

LiveCycle Developer Lab:
*LiveCycle Guides Next - A
Common Data Model
Applied*





LiveCycle Guides Next: A Common Data Model Applied

Jim Leask

Senior Computer Scientist

Adobe Systems, Inc.



TABLE OF CONTENTS

EXERCISE OVERVIEW 4

EXERCISE 1: DEVELOPMENT ENVIRONMENT INTRODUCTION 6

 OBJECTIVES:.....6

 ASSETS PROVIDED:6

 TASKS6

Task 1: Explore Guide Designer 6

Task 2: Explore the Data Modeler..... 10

Task 3: Debugging Guides 12

EXERCISE 2: DATA MODEL TECHNIQUES17

 OBJECTIVES:.....17

 ASSETS PROVIDED:17

 TASKS17

Task 1: Add an Entity and Properties to the data model 17

Task 2: Explore the Source view of the Model 21

EXERCISE 3: GUIDE DESIGNER TECHNIQUES23

 OBJECTIVES:.....23

 ASSETS PROVIDED:23

 TASKS23

Task 1: Add the Emergency Contact Panel..... 23

Task 2: Use AquoStyle to set colors and branding for the guide..... 28

EXERCISE 4: DATA MODEL SERVICES31

 OBJECTIVES:.....31

 ASSETS PROVIDED:31

 TASKS31

Task 1: Import a web service into the data model..... 31

Task 2: Review the Data Model with the Service 34

EXERCISE 5: USING SERVICES IN A GUIDE36

 OBJECTIVES:.....36

 ASSETS PROVIDED:36



LiveCycle Guides Next - A Common Data Model Applied

TASKS36

Task 1: Configure the LookupNewHire Entity and Service36

Task 2: Activate the Search Screen40

EXERCISE 6: CREATING A DYNAMIC GUIDE.....44

 OBJECTIVES:.....44

 ASSETS PROVIDED:44

 TASKS45

Task 1: Create and use a rule to control visibility of a field.....45

Task 2: Modify the 'displayDetailsPanel' rule49

Task 3: Explore other dynamic techniques51

Task 4: Using nested panels and dynamic panels (optional exercise).....53

EXERCISE 7: RUNNING THE GUIDE IN A PROCESS (OPTIONAL EXERCISE).....56

 OBJECTIVES:.....56

 ASSETS PROVIDED:56

 TASKS56

Task 1: Run the Guide directly from a browser.....56

Task 2: Run the Guide from Workspace.....59

Task 3: See the data in a PDF within Workspace.....61

Exercise Overview

Exercise	Summary	Estimated Time
1	<p>Development Environment Introduction</p> <p>Introduce the Guide Designer and Data Modeler tools within Workbench, and become familiar with their basic usage by exploring the supplied "<i>Employee Onboarding</i>" application. Introduce basic development techniques, including preview and debugging.</p>	10 min
2	<p>Data Model Techniques</p> <p>Add a new data model <i>Entity</i> to the data model to hold "Emergency Contacts" information. Link this entity to the existing model, and add business rules as appropriate.</p>	15 min
3	<p>Guide Designer Techniques</p> <p>Add a new Panel to the Guide and use the variable column layout to display "<i>Emergency Contact</i>" information. Add a custom style for the AQUO branding of the guide.</p>	15 min
4	<p>Data Model Services</p> <p>Add a web service to the data model and investigate the capabilities of services within the model.</p>	10 min
5	<p>Using Services in a Guide</p> <p>Extend the Guide's model by adding a data object that is tied to the service provided by the data object. Create a search screen that will trigger the service to be executed on demand, and copy the data to the appropriate location in the main business model.</p>	15 min

6	<p>Creating a Dynamic Guide</p> <p>Add a "Rule" and use it to control the visibility of the "Record not found" field on the search screen. Modify an existing rule and configuration that controls the ability to dynamically display and hide other panels depending on the results of the search operation.</p>	15 min
7	<p>Running the guide in a process (optional)</p> <p>Run the completed guide within a browser and submit the data. Launch Workspace and view the same guide and data in the Workspace environment, and again as a PDF within Workspace.</p>	10 min

Exercise 1: Development Environment Introduction

This exercise introduces the *Guide Designer* and *Data Modeler* tools within *LiveCycle Workbench*, and common development techniques, including *preview* and *debugging*.

Objectives:

- By the end of this exercise you will be familiar with the *Guide Design* and *Data Modeler* tools within *Workbench*.
- You will be able to preview a guide, and do basic debugging operations.

Assets Provided:

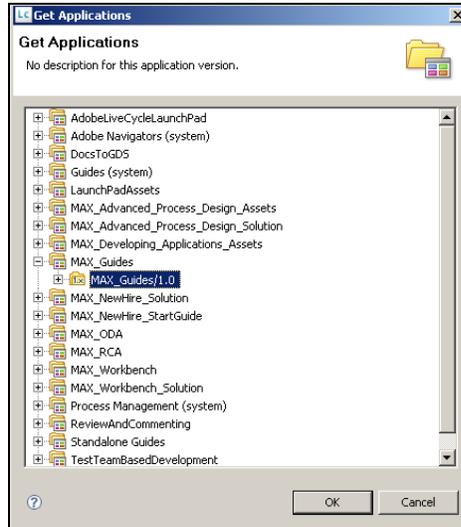
None.

Tasks

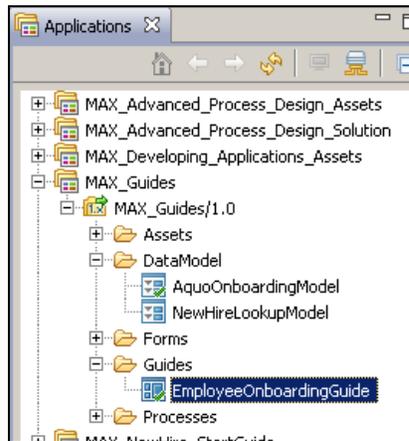
Task 1: Explore Guide Designer

In this first task, you will load an application into *Workbench* and explore the basic structure of *Guide Designer*.

1. Launch *Workbench* and login using the credentials:
UID="Administrator"
PWD="password"
2. In the Applications view, verify the application *MAX_Guides* is listed. If necessary, retrieve it from the server by selecting the menu item **File->Get Application...** and select the "**MAX_Guides/1.0**" application from the *Get Applications* dialog.

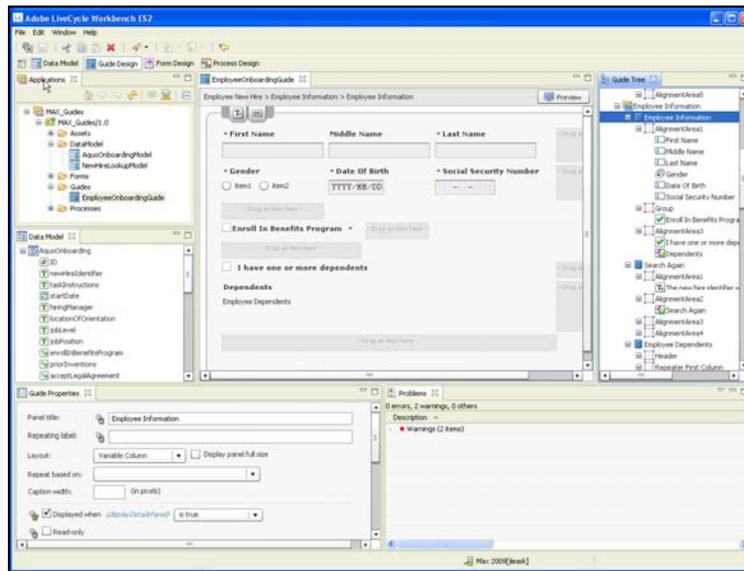


3. In the Workbench Applications view, open the folder "MAX_Guides/1.0/Guides". You should see *EmployeeOnboardingGuide*. The folder "DataModel" holds the "AquoOnboardingModel" that will be used throughout the lab.

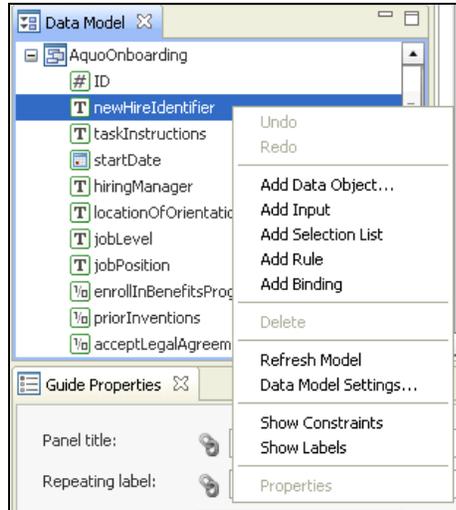


4. Double click *EmployeeOnboardingGuide* from the Guides folder to open Guide Designer with this guide. Note the following key areas of Guide Designer

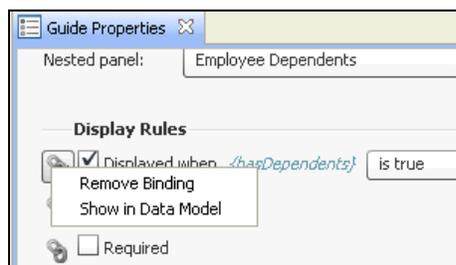
- Guide Canvas** Shows an interactive view of the panel layout
- Guide Tree** An outline structure of the guide, showing sections, panels, and items
- Guide Properties** The properties page corresponding to the selected item in the Guide Tree
- Data Model Tree** The outline view corresponding to the data model associated with the data model used by this guide.



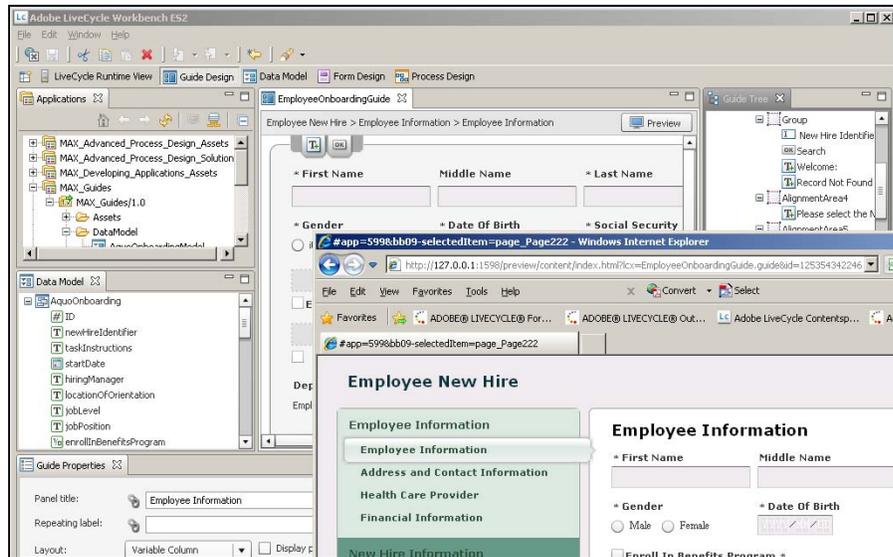
5. Right-click anywhere on the *Data Model* tree. Note the popup menu of commands that are available.



6. Hover the mouse over one of the link buttons in the *Guide Properties* page, such as the one immediately to the left of the "Displayed" checkbox. You may need to scroll down in the *Guide Properties* view to see the display rules section. These buttons are used with drag/drop operations from the *Data Model* tree to bind a data element to the property. The link image will change when bound. Click on the link icon and notice there is a popup menu available with commands to remove the binding if present, and show the location of the data element in the data model tree if appropriate.



7. In the top-right corner of the Guide canvas, note the "Preview" button. Click this button to bring up the running guide in a browser. This is a very common activity while developing a guide. The following image shows the preview running within the Firefox Browser. This illustration shows the *Guide Designer* visible in the background, with the *Preview* button used to launch the browser.

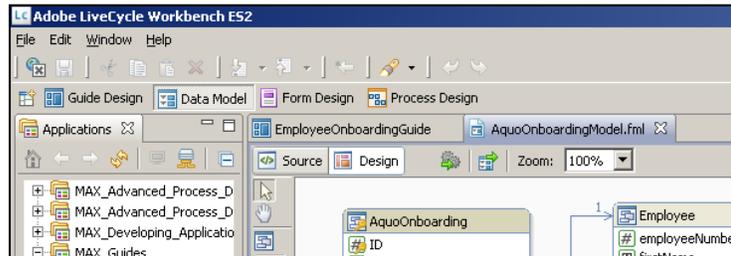


8. Close the browser, but leave the *Guide Canvas* open. It will be used again in a later exercise.

Task 2: Explore the Data Modeler

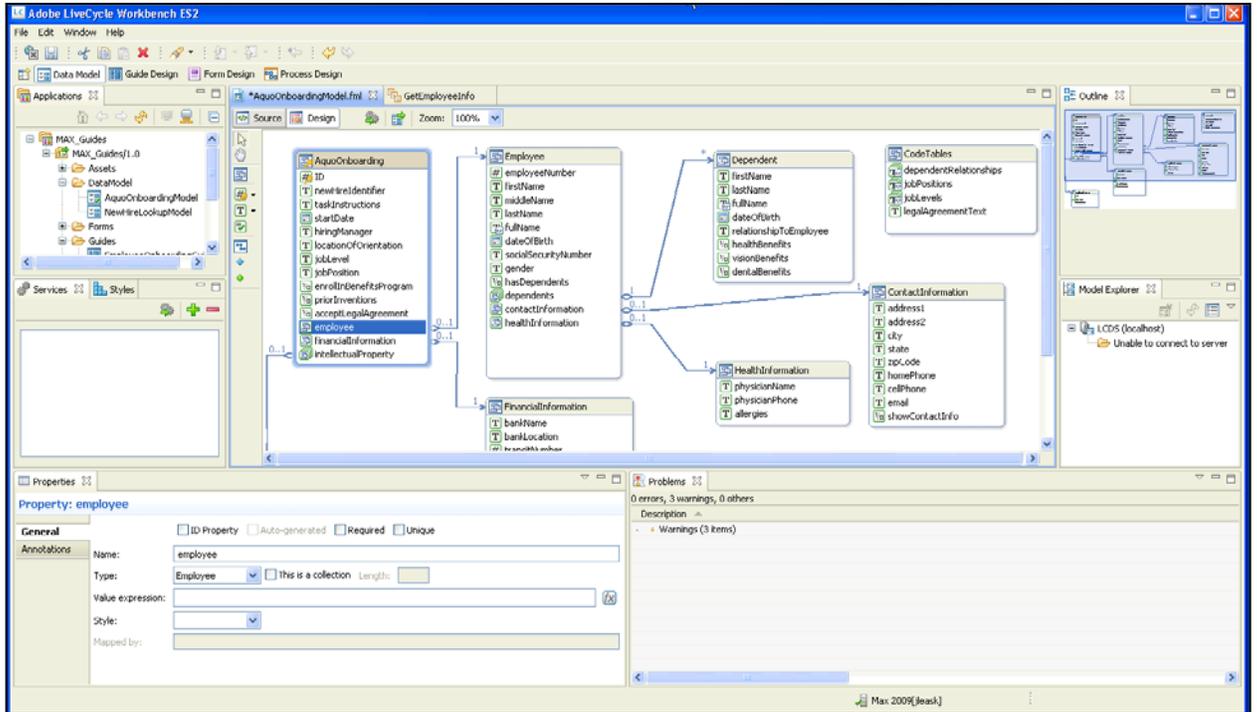
In this task you will explore the basic structure of the data modeler, and see how it interacts with the Guide Designer.

1. In the Applications view of Workbench, open the *MAX_Guides/1.0/DataModel* folder and double click the *AquoOnboardingModel* to open this model in the *Data Modeler*
2. In this version of *Workbench*, you will need to manually click on the *Data Model* perspective, which is located above the editor area. When switching between a *Guide Design* and *Data Model* canvas, this perspective switching is not automatic.



3. Note the following key areas of the *Data Modeler*:

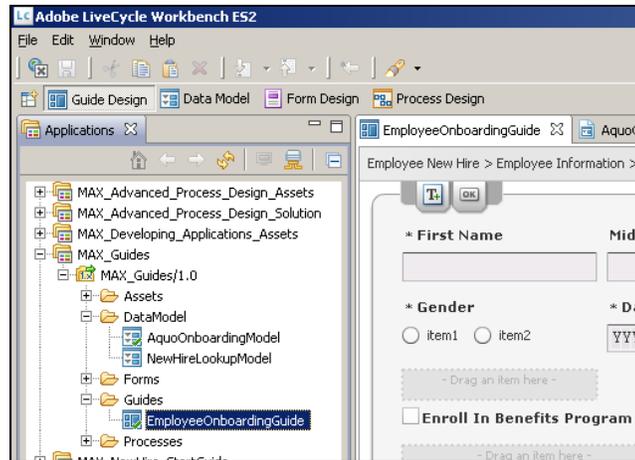
- | | |
|-----------------------|--|
| Design Canvas | Shows a graphical view of the entities, properties and relationships within the model. |
| Outline | A thumbnail view of the model that indicates the subset of the model that is visible in the design canvas. |
| Properties | Contains the properties that correspond to the entity or property that is selected in the design canvas. |
| Services | Shows the services that are contained within the model, and allows them to be imported, added or deleted. |
| Styles | Shows the styles that are defined within the model. The styles may be applied to properties to ensure consistency. (by default, styles are in a view that is behind the <i>Services</i> tab) |
| Problems | All errors or warnings from the model are reported in the problems list. You may need to right-click on the error to see extra detail about the error. |
| Model Explorer | This view is not used by the <i>Guide Designer</i> for this release. |



Task 3: Debugging Guides

In this task you will add a panel that allows the internal data of the guide to be viewed at runtime. This is only for use during development and debugging.

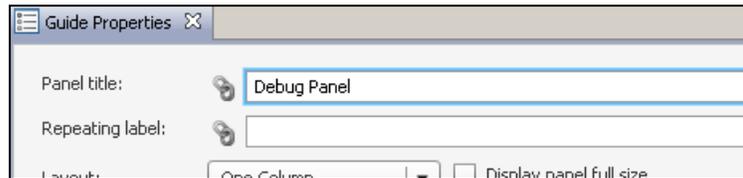
1. Open the *EmployeeOnboardingGuide* if it is not already open and switch to the *Guide Design* perspective if necessary.



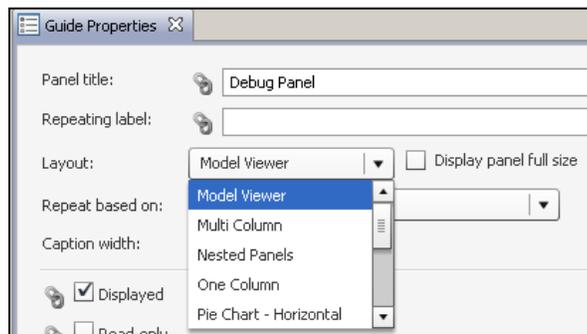
2. In the *Guide Tree*, note the section titled "*Debug*" at the bottom of the tree (you may need to scroll the tree to the bottom to see this)
3. Right click on the *Debug* section, and select "**New Panel**". If prompted, check out the guide.



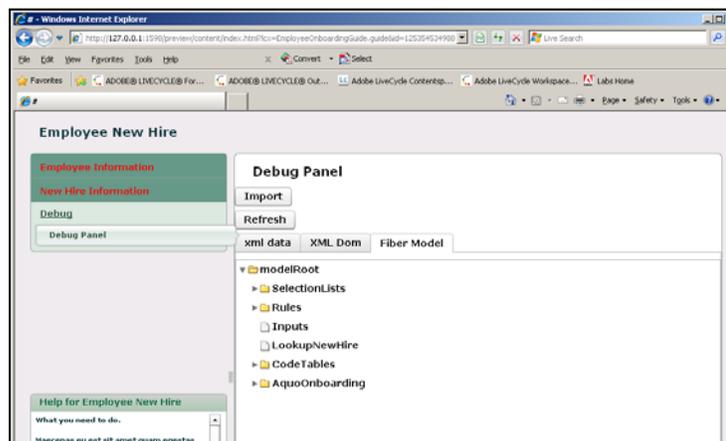
4. Ensure the newly added panel is selected. In the *Guide Properties* view, rename the panel to "**Debug Panel**".



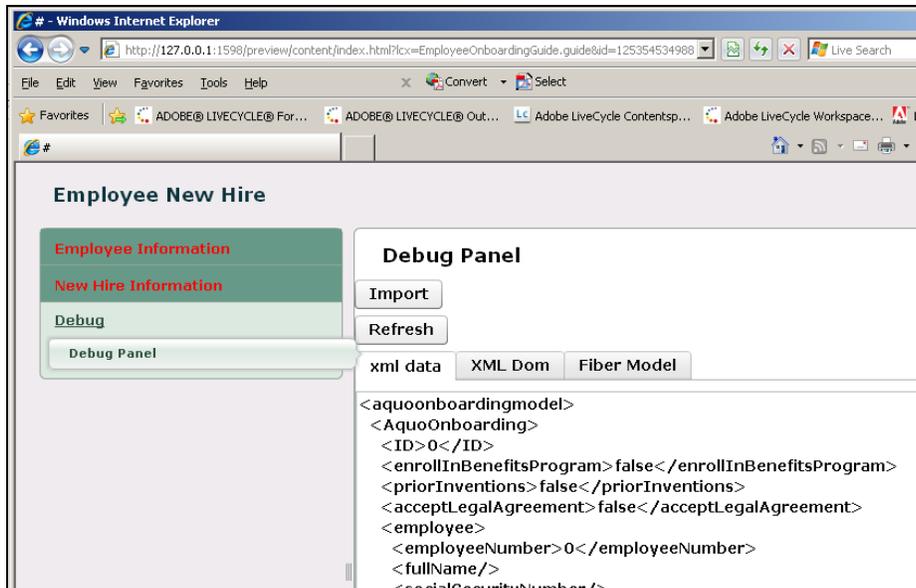
5. Use the Layout drop down and select "Model Viewer".



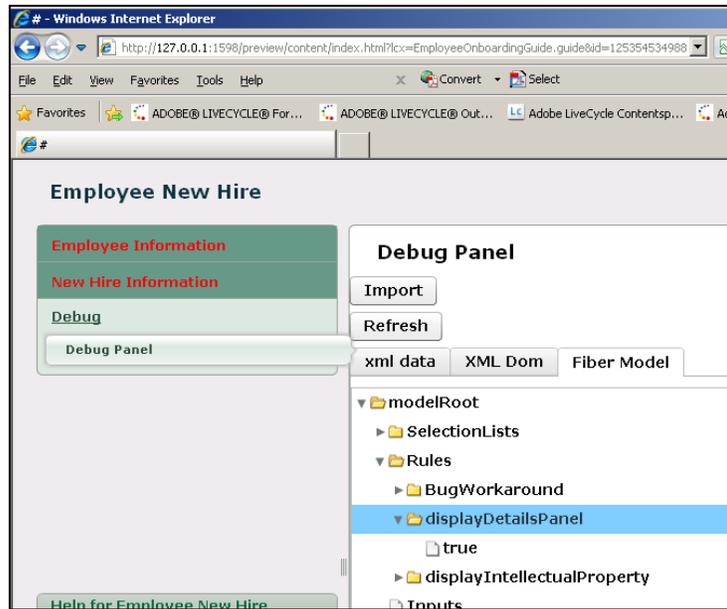
6. Click on the *Preview* button to view the guide. Note that a panel is called "Debug Panel" is available at the end of the guide. This contains a browser that can be used to investigate the data currently available to the guide.



- Click on the "xml data" tab to view the XML data that is currently held by the guide. This is the data that will be submitted by the guide, and is useful for debugging data or data organization problems.



- Click on the "Fiber Model" tab and open the following path: "**ModelRoot>Rules>displayDetailsPanel**". It is worth noting that there is no script in a guide, so you never have to debug script code! However, you do set up rules or expressions (covered in a later section) and looking at the results of the rule is useful when debugging.



9. Close the browser.
10. Save your data model.

Exercise 2: Data Model Techniques

In this exercise you will add a new data model entity to the data model to hold "Emergency Contacts" information and link this entity to the existing model.

Objectives:

- By the end of this exercise, you will be able to add an *Entity* to the data model, add *Properties* to it and set attributes as appropriate. The new entity will be associated with another entity in the model with a *Relationship*.

Assets Provided:

None.

Tasks

Task 1: Add an Entity and Properties to the data model

In this task you will add a new Entity to the model to hold "Emergency Contact" information for the application. This will be linked to the existing "Employee" entity.

1. Open the *AquoOnboardingModel* from the 'DataModel' folder if it is not already open. (Remember to manually switch to the Data Model perspective)
2. From the tool pallet on the left of the design canvas, find the "Entity" tool (the tooltip gives the name of each tool) and **drag it onto the canvas**. Alternatively, you can right-click on the canvas and select "New Entity". Check out the model if prompted.



3. Select the newly added Entity, and from within the Properties view **set the name to "EmergencyContact"**. Double clicking the title and editing it directly can also modify the entity name.

Note: The entity name must not have any spaces, and must start with a capital letter. An error will be generated if the above rules are not followed.



4. From the tool palette, drag a text property onto the *EmergencyContact* entity. Alternatively, right-click within the entity and select '**New Property**' from the popup menu.



5. Select the newly added property. You will use the Properties view to set the name and datatype for the following properties. Use a datatype of *'string'* in all cases. (Property names typically start with a lowercase letter)

contactName

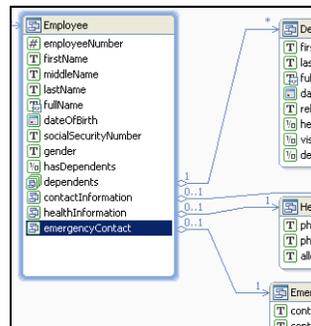
contactHomePhone

contactEmail

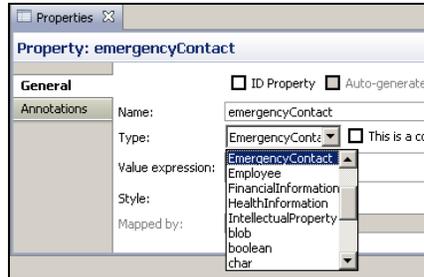
relationshipToEmployee



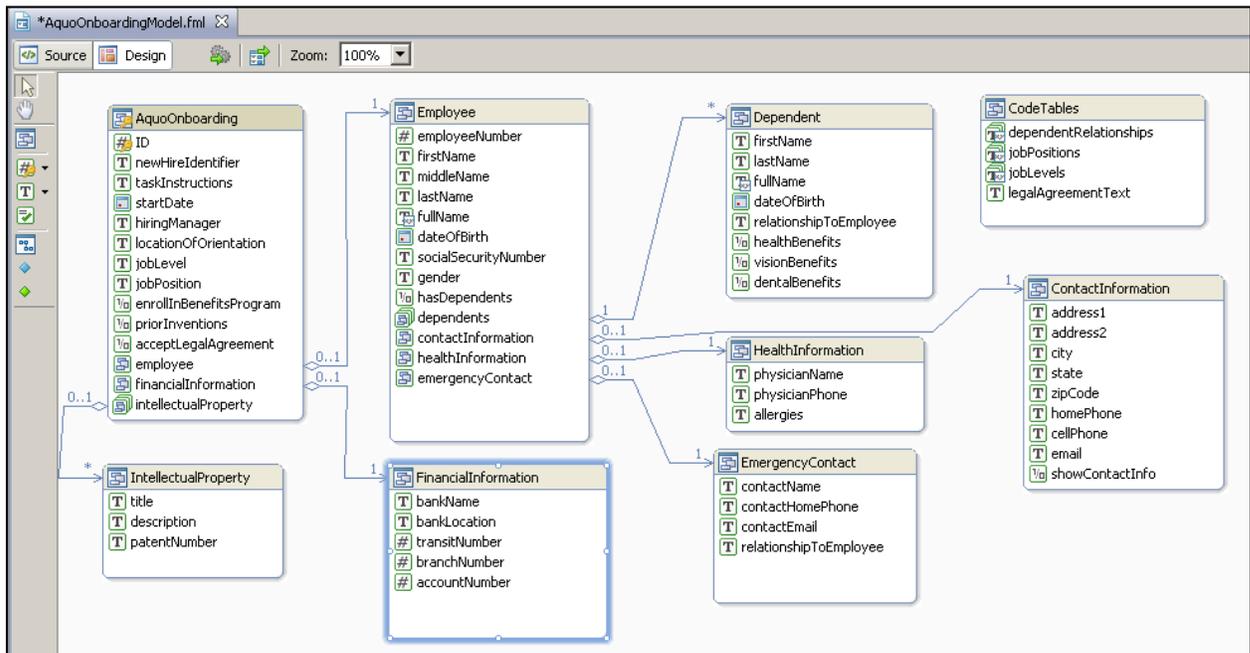
6. In the "Employee" Entity, add a new property and give it the name **'emergencyContact'**



7. Select the newly added *'emergencyContact'* property and open the *'Type'* drop down list. Note that "EmergencyContact" is now available in this list as a datatype. Select "EmergencyContact" from the drop list.



8. Review your model. It should appear as shown.



9. Save the data model. Keep the data model open.

Task 2: Explore the Source view of the Model

In this task, you will investigate the XML based source view of the data model

1. The data model is stored in XML format. **Click the Source button** in the top-left corner of the *Design Canvas* to view the XML source of the data model. This example shows the "EmergencyContact" entity that was just added in the previous task.



```

132     <property name="branchNumber" type="integer"/>
133     <property name="accountNumber" type="integer"/>
134 </entity>
135 <entity name="EmergencyContact">
136     <annotation name="VisualModeler">
137         <item name="x">527</item>
138         <item name="y">329</item>
139         <item name="width">165</item>
140         <item name="height">126</item>
141     </annotation>
142     <property name="contactName" type="string" required="true"/>
143     <property name="contactHomePhone" type="string"/>
144     <property name="contactCellPhone" type="string"/>
145     <property name="contactEmail" type="string"/>
146     <property name="relationshipToEmployee" type="string"/>
147 </entity>
148 <style name="MandatoryNewHireIdentifier">

```

This XML may be edited directly in the "Source" view. Any errors will be reported in the editor, as well as on the "Problems" view. Changes made to the XML will immediately show up in the graphical design view when it is displayed.

2. Copy the "contactHomePhone" line and paste on the line just below.

3. Modify the new property to have the name "contactCellPhone".

Tips for speed editing:

Control-Alt Up/Down Arrow duplicate the current line in the editor and place it above/below the original location.

Alt Up/Down Arrow moves the current line up/down

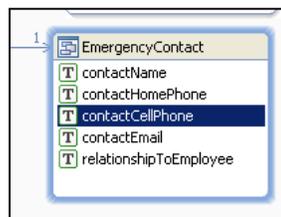
Control-d deletes the current line

```

<entity name="EmergencyContact">
  <annotation name="xmi:isModeler">
    <attribute name="isModeler" data-type="boolean" value="true"/>
  </annotation>
  <property name="contactName" type="string"/>
  <property name="contactHomePhone" type="string"/>
  <property name="contactCellPhone" type="string"/>
  <property name="contactEmail" type="string"/>
  <property name="relationshipToEmployee" type="string"/>
</entity>

```

4. Return to *Design View* and verify the new "contactCellPhone" property is in the "EmergencyContact" entity.



5. Save the model. This will cause the guide to recompile if it is still open.

Exercise 3: Guide Designer Techniques

In this exercise you will add a new panel to the guide and use the variable column layout to display "Emergency Contact" information. You will use a custom style to change the colors and logo to configure the AQUO branding of the guide.

Objectives:

- By the end of this exercise you will understand how *Sections* and *Panels* give structure to your application
- You will be able to add a panel of controls used to display and manipulate data.
- You will be able to set the colors and styling for your guide.

Assets Provided:

1. The AquoStyle.css file, and AQUO logo
 - a. Location - C:\MAX2009\Guides\Exercises\AquoStyle
 - b. Purpose - Change the colors and styling for the guide
 - c. Misc - compile this CSS to generate a SWF using the Flex compiler (mxmclc)

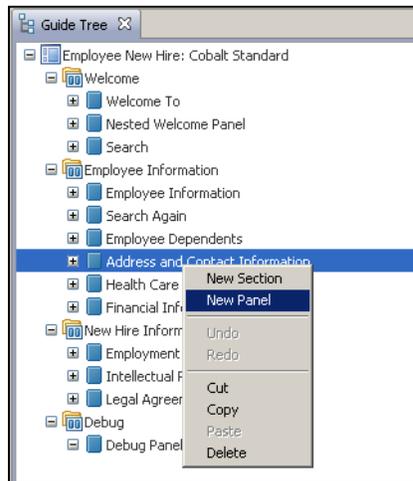
Tasks

Task 1: Add the Emergency Contact Panel

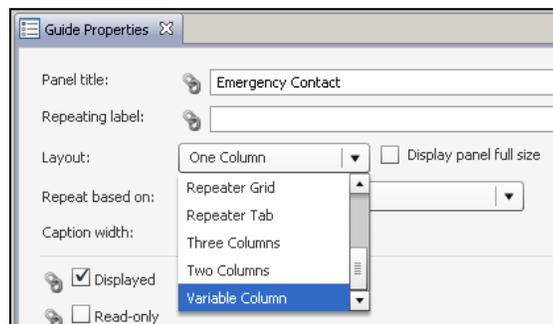
In this first task, you will add a panel to hold the *Emergency Contact* information that has been added to the data model.

1. Open the "**EmployeeOnboardingGuide**" if it is not already open. (Select the *Guide Design* perspective if necessary)
2. In the *Guide Tree*, note the structure of this guide. Expand and collapse levels of the tree to show or hide information as you see fit. For this example, all "Panel" nodes have been collapsed so that the structure of the guide is evident.
 - Welcome, Employee Information, New Hire Information, and Debug are "Sections" They can be viewed a folders that simply contain panels, or optionally other sections.
 - Each Section can contain panels, which will be visible to the user in the resultant guide.

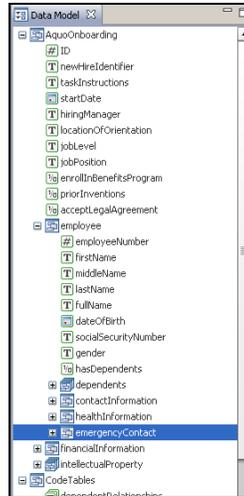
- Right-click on the 'Address and Contact Information' panel. Select 'New Panel' from the popup menu. This will add a new panel under this location.



- Select the newly added panel in the *Guide Tree*, then go to the *Guide Properties* view to set the pane title to "Emergency Contact"
- Set the 'Layout' drop down to 'Variable Column'. This layout gives the greatest amount of control in laying out the panel in a grid format that is common for business applications.



- In the *Data Model* tree, find the 'Emergency Contact' entry that was recently added to the data model. It should be located in:
AquoOnboarding->employee->emergencyContact



- Expand the '*emergencyContact*' node and drag '**contactName**' from the *Data Model* tree onto the drop zone of the Guide canvas. (drop it onto the small square labeled "Drag an item here")



- Note that when the data element is dropped onto the canvas, a new control is added to both the canvas and the Guide Tree. This item is selected, and its properties are available in the Guide Properties view.
- Set the width property to 100% (you may need to scroll the Guide Properties view to see the width setting)
Note: 100% resizes the field the full width of the panel. If you just specify '100' (without %) you will only have a field 100 pixels wide.

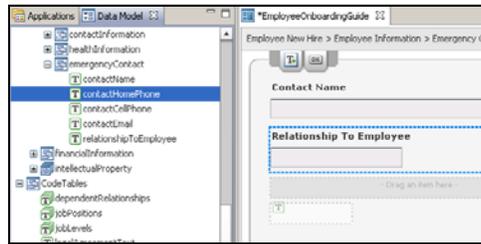


LiveCycle Guides Next - A Common Data Model Applied

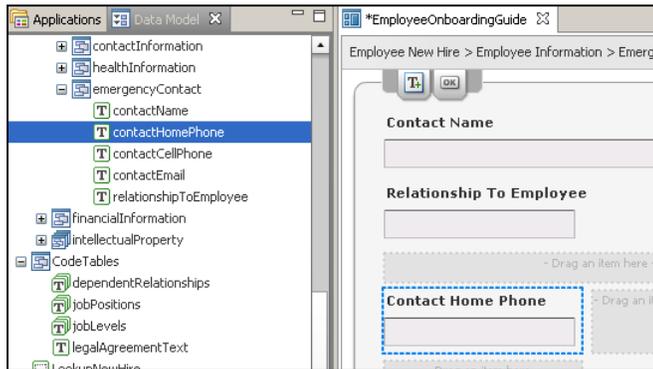
Control Type	
Display as:	Text Input
Display format:	
Style:	default
Width:	100% (in pixels or %)
Height:	(in pixels)

- 9. Drag **relationshipToEmployee** under the "Contact Name" field.
- 10. Drag **contactHomePhone** to the drop zone at the very bottom of the canvas. This drop zone is for a different alignment area, which allows the phone numbers to be aligned independently from the fields above it.

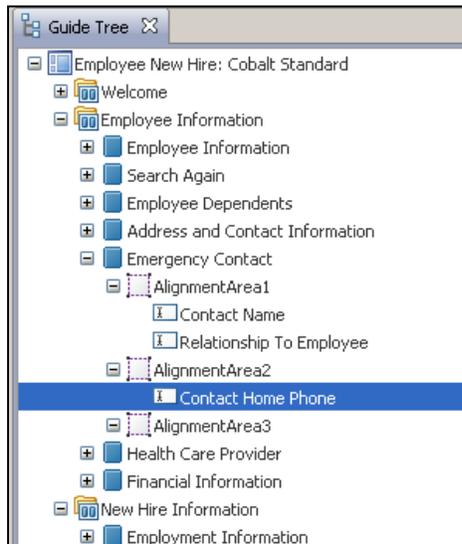
During the drag, the drop zone will go white if it will accept the drop.



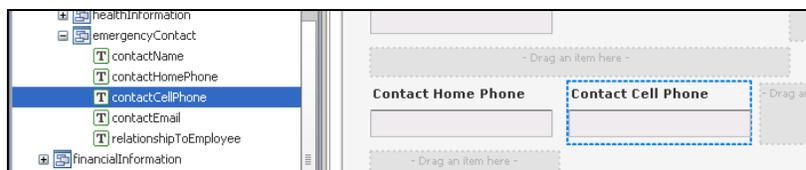
After dropping the field is created.



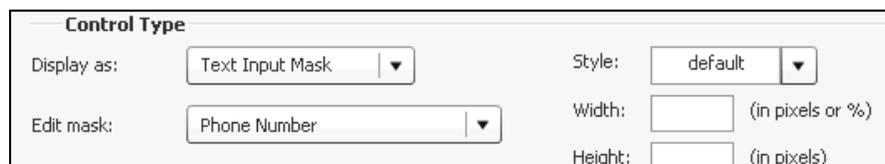
By looking in the *Guide Tree*, you can verify that the "Contact Home Phone" field has been added to a different alignment area from the first two fields.



11. Drag **contactCellPhone** and drop it on the Guide Canvas to the right of the "ContactHomePhone" field.



12. By default, string properties use a "Text Input" control. These can be modified as is done for the phone number fields. Select the **contactHomePhone** phone number and set the "Display as:" drop down list to "Text Input Mask" and select an "Edit mask:" to "Phone Number".



Modify **contactCellPhone** to use the phone number masks as well.

13. Add the remaining properties to the view using the same drag/drop technique. The result should resemble the following canvas.

The screenshot shows a canvas titled "Employee New Hire > Employee Information > Emergency Contact Information" with a "Preview" button in the top right. The canvas contains several form fields and labels:

- Contact Name**: A text input field with a dashed blue border and a "Drag an item here" placeholder to its right.
- Relationship To Employee**: A text input field with a "Drag an item here" placeholder to its right.
- Contact Home Phone**: A masked input field with a "() -" pattern and a "Drag an item here" placeholder below it.
- Contact Cell Phone**: A masked input field with a "() -" pattern and a "Drag an item here" placeholder to its right.
- Contact Email**: A text input field with a "Drag an item here" placeholder to its right.

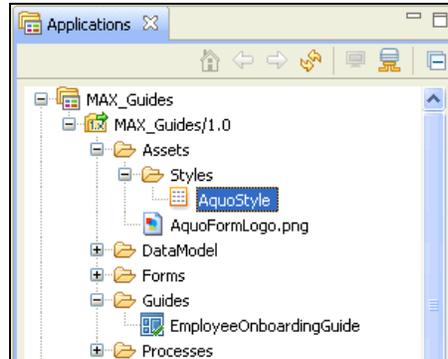
 There are also several other "Drag an item here" placeholders scattered across the canvas, indicating areas where additional components can be added.

14. Use Preview to view the results in a browser.
15. Save the guide.

Task 2: Use *AquoStyle* to set colors and branding for the guide

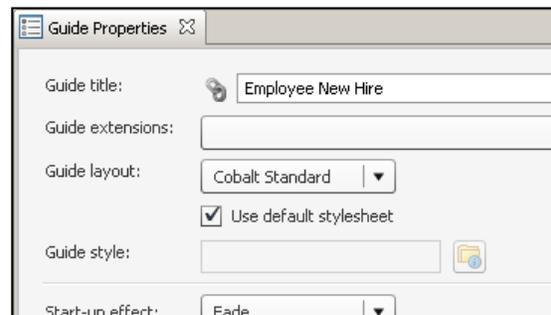
It is important to use the colors and logo for the organization that owns the application. This can be done by applying styling to the guide.

1. In the *Applications* view, open the "**Assets->Styles**" folder in the *MAX_Guides* application. A file called '*AquoStyle*' exists at this location that is a SWF file that was generated by compiling a CSS file with the Flex compiler.

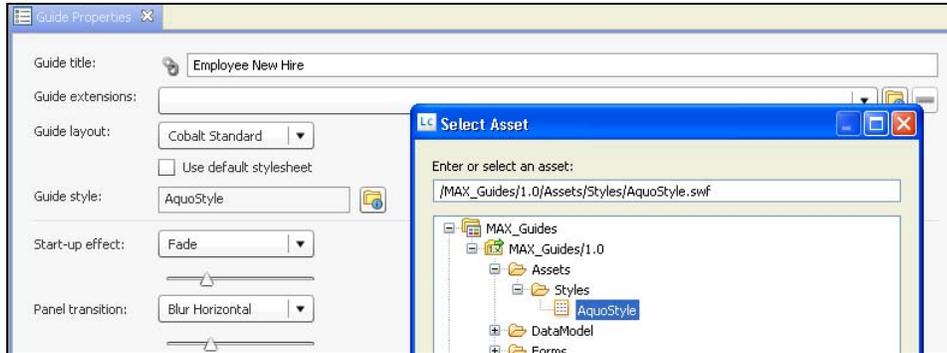


2. Select the top-level node in the *Guide Tree*, called "**Employee New Hire: Cobalt Standard**". This top node represents the guide itself, and has high level properties available in its *Guide Properties* page.

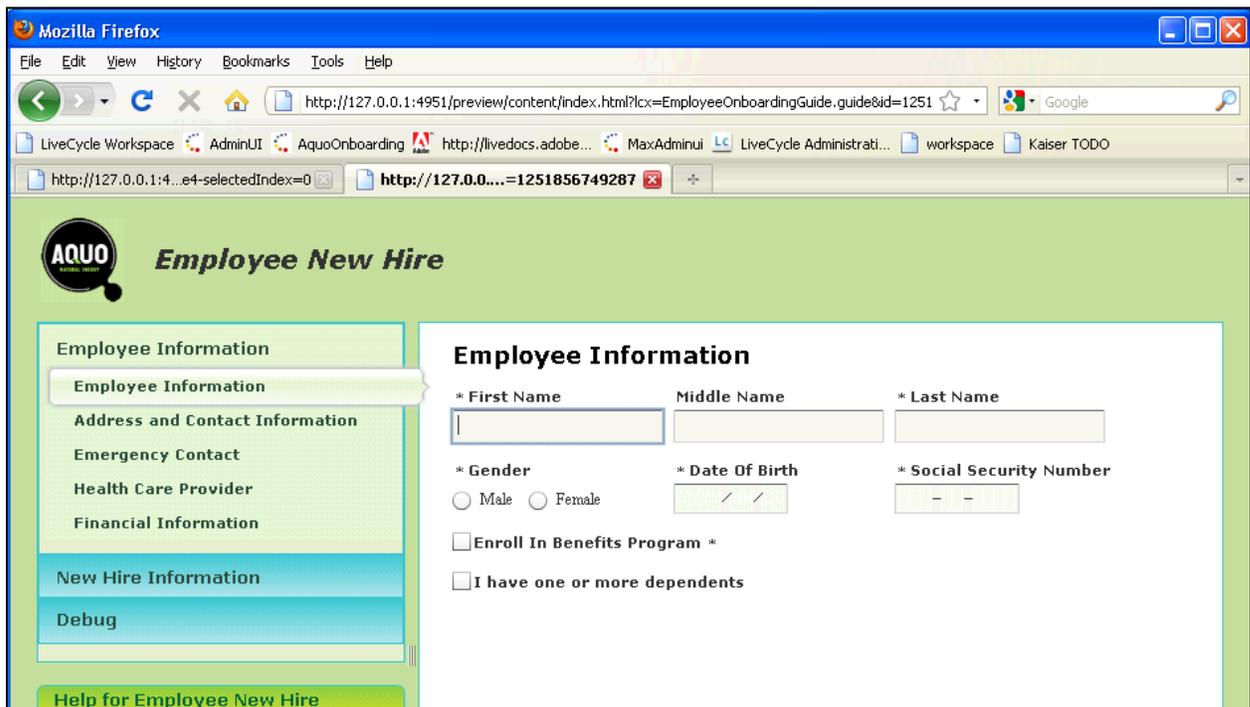
Note that this guide is using the "*Cobalt Standard*" guide layout. This controls the high level navigation used by the guide. Other options are available and may be explored time permitting.



3. Uncheck "**Use default stylesheet**" checkbox in the *Guide Properties* view.
4. Click the folder icon to the right of the "Guide style:" property. This brings up an asset picker. Navigate to "**MAX_Guides/1.0->Assets->Styles->AquoStyle**" and select it.



5. Save the guide.
6. Click on the **Preview** button to view the guide in a browser. Note that the guide now uses the green color scheme and logo that match the styling guidelines for *Aquo*.



Exercise 4: Data Model Services

Web services can be accessed by the data model and used by client applications, including Guides. This exercise will add a web service to the data model that looks up user information based on a unique identifier that is known to the individual being hired by Aquo.

Objectives:

- By the end of this exercise you will be able to import a web service to a data model.

Assets Provided:

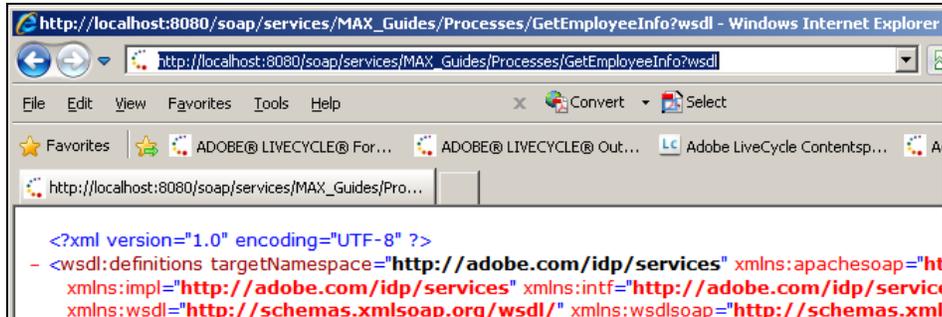
1. This lab uses a process called "*GetEmployeeInfo*". This process provides a web service interface to do an employee lookup. This WSDL is needed to configure the web service.
 - a. WSDL:
http://aquo.com:8080/soap/services/MAX_Guides/Processes/GetEmployeeInfo?wsdl
 - b. Purpose – Lookup user information based on a "*newHireIdentifier*" number.

Tasks

Task 1: Import a web service into the data model

In this first task, you will import a web service to the data model. This allows the web service functionality to be used by clients to the data model, including *Guide Designer*.

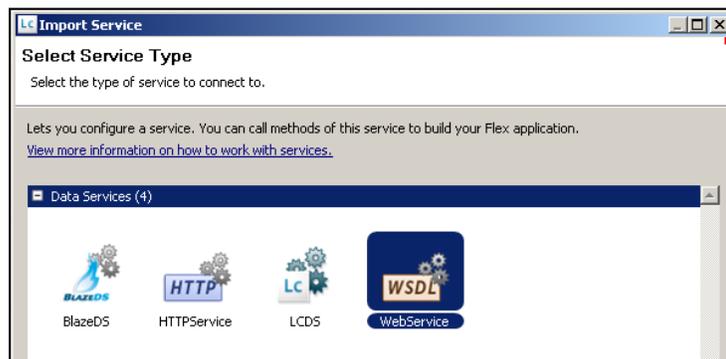
1. Open the **AquoOnboardingModel** if it is not already open. (Switch to the *Data Model* perspective if necessary)
2. The WSDL for the web service to be used must be available. Open the browser (Internet Explorer) and select the bookmark: '**Favorites->Guides ES Next->GetEmployeeInfo WSDL**'
3. The contents of the WSDL may be interesting, but you can ignore the details. You need the URI, so copy it from the entry line in the browser.



4. Return to Workbench Data Modeler.
5. In the *Services* view, click the  icon. This opens the web service import dialog.

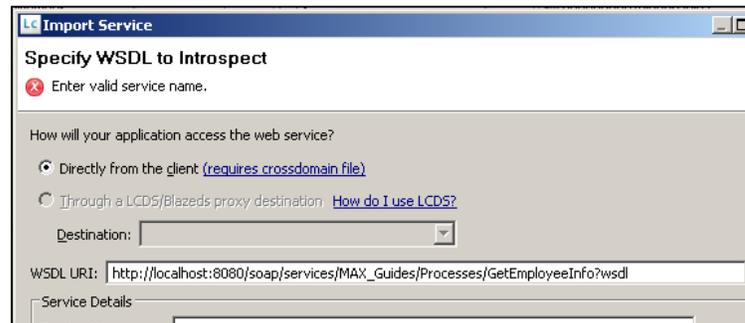


6. Select the **WebService** icon from the dialog and click Next.

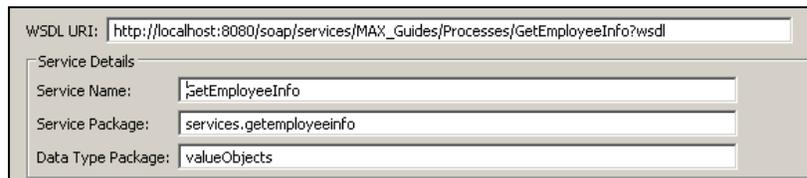


7. **Paste the WSDL** that you copied from the browser in step 3 into the "WSDL URI" field of the *Import Service* dialog.. The results should be:
http://localhost:8080/soap/services/MAX_Guides/Processes/GetEmployeeInfo?wsdl

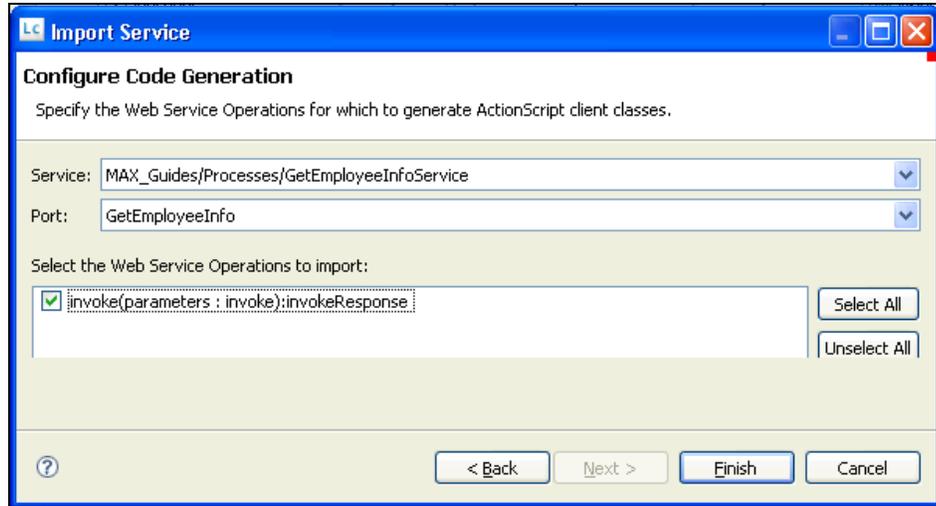
Instead of *localhost* in the above URL, you should use the name of the server. In this case you could replace *localhost* with '*aquo.com*'. However, either works for this lab.



8. Tab out of the WSDL URL field will automatically populate the Service Details with appropriate values.



9. Click Next.
10. This web service supplies an *'invoke'* operation. We want to include this in our data model service. Select the check box beside *'invoke'*.



11. Click Finish. You should notice that a new service has been added to the *Services* view.

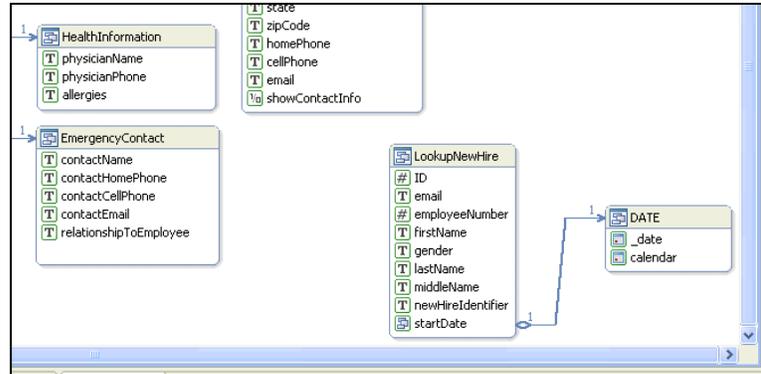


Task 2: Review the Data Model with the Service

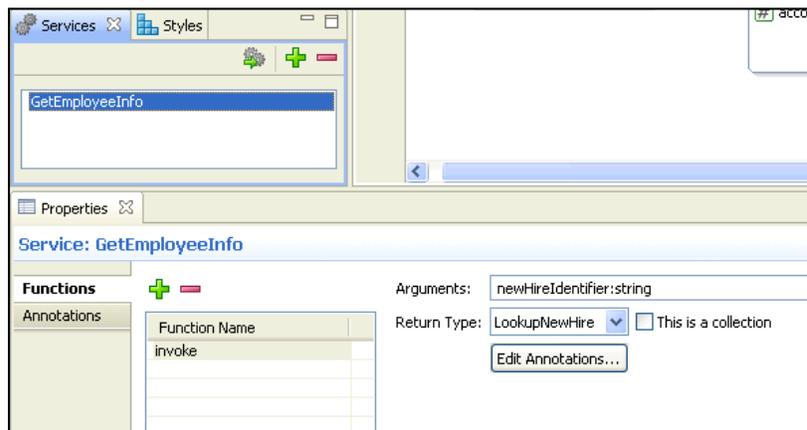
Importing the web service has modified the data model. Entities corresponding to the return type of the web service have been added, as well as the service and call information. This exercise reviews these modifications to the model.

1. This web service returns an entity called '*LookupNewHire*' and '*DATE*'. This entity has been added to the model. They may be placed by default over top of existing entities in your model.

Drag the *LookupNewHire* and *DATE* entities from the top-left corner to the bottom-right corner of the model.



2. Notice that the web service is now visible in the Services view, and if it is selected, the Properties view shows the callable functions (invoke for this service) as well as the arguments to each function. **Click on the invoke function** to show the arguments and return type.



3. Save the model.

Exercise 5: Using Services in a guide

In this exercise, you will extend the guide's model by adding a data object that is tied to the service from the data model, modify the search screen to execute the service, and copy the data to the appropriate location in the main business model.

Objectives:

- By the end of this exercise, you will be able to use a service provided by the data model to lookup data and present it to the user.

Assets Provided:

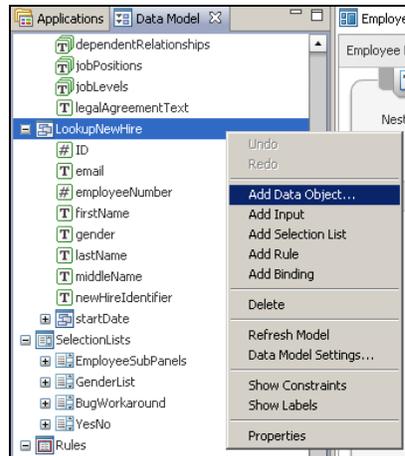
None.

Tasks

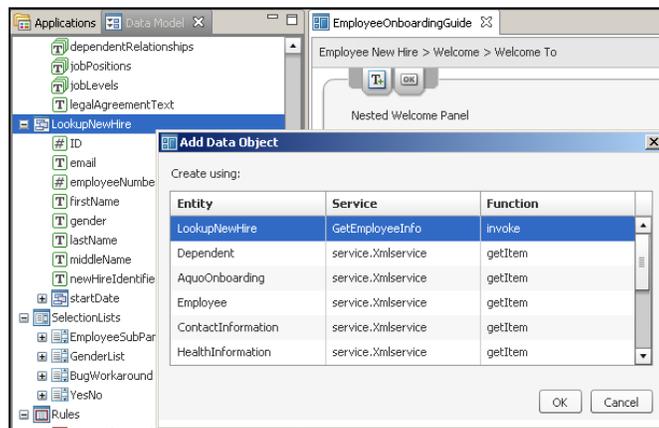
Task 1: Configure the LookupNewHire Entity and Service

In this task, you will extend the guide's data model with a 'LookupNewHire' entity. This object is the result of calling the 'LookupNewHire' service in the data model.

1. Open the **EmployeeOnboardingGuide** if it is not already open. (Select the *Guide Design* perspective if necessary)
2. Right-click on the Data Model tree, and select '**Add Data Object...**' to open the *Add Data Object* dialog.



3. The 'Add Data Object' dialog box provides a list of Entities that can be dynamically added to the guide's internal data model. In this case we want to call the **GetEmployeeInfo** service from the model, and store the result in a 'LookupNewHire' entity.

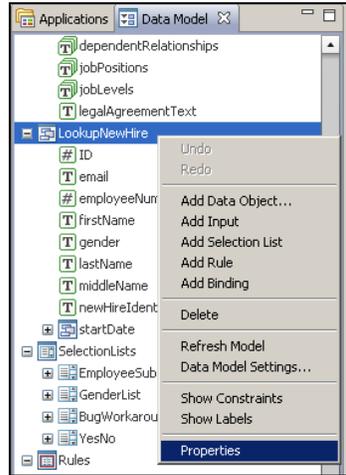


You would normally pick this combination from the list, as shown in this diagram. This operation has already been done for you.

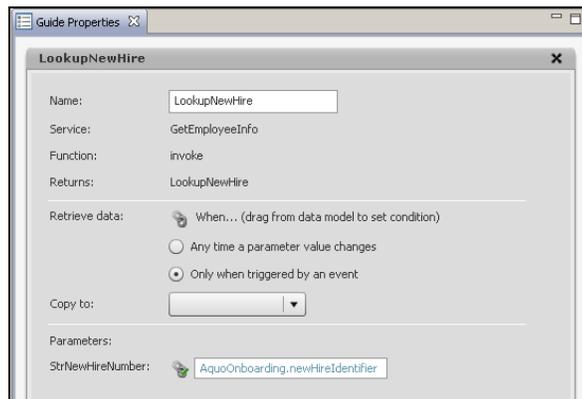
4. **Cancel** from the Add Data Object dialog.



5. Right-click on "lookupNewHire" in the Data Model tree, and select Properties.



This will display the *LookupNewHire* properties within the *Guide Properties* view. These properties allow the service call to be configured.



6. In the 'Retrieve Data' section of the *LookupNewHire* properties, verify the 'Only when triggered by an event' is selected. We do not want the service to execute until we are ready.

Retrieve data:	<input checked="" type="radio"/> When... (drag from data model to set condition) <input type="radio"/> Any time a parameter value changes <input type="radio"/> Only when triggered by an event
Copy to:	<input type="text"/>

7. Locate the 'Parameters' section of the *LookupNewHire* property page. Verify this parameter is bound to 'AtuoOnboarding.newHireIdentifier'.

If you need to configure this parameter, locate '*AquoOnboarding->newHireIdentifier*' and drag it onto the link button for the 'StrNewHireNumber' parameter within the *LookupNewHire* properties.

The screenshot shows two overlapping windows. The top window, titled 'Data Model', displays a tree view of the 'AquoOnboarding' data model with the following items: ID, newHireIdentifier, taskInstructions, startDate, hiringManager, locationOfOrientation, jobLevel, and jobPosition. The 'newHireIdentifier' item is selected. The bottom window, titled 'LookupNewHire', shows the configuration for a service. It includes fields for Name (LookupNewHire), Service (GetEmployeeInfo), Function (invoke), and Returns (LookupNewHire). The 'Retrieve data' section is identical to the one shown in the previous image, with 'Only when triggered by an event' selected. In the 'Parameters' section, the 'NewHireIdentifier' parameter is configured with the value 'AquoOnboarding.newHireIdentifier'.

8. Verify the **small green checkbox** on the link button, and the path to the linked object in the corresponding field. This indicates that the input parameter to the service is linked to the *newHireIdentifier* data.

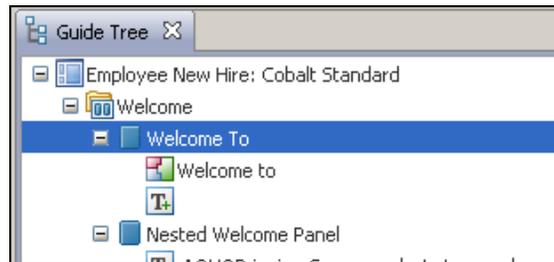


9. **Close** the *LookupNewHire* property page by clicking the X in the top-right corner of its title. This property page will slide down out of view.
10. Save the guide.

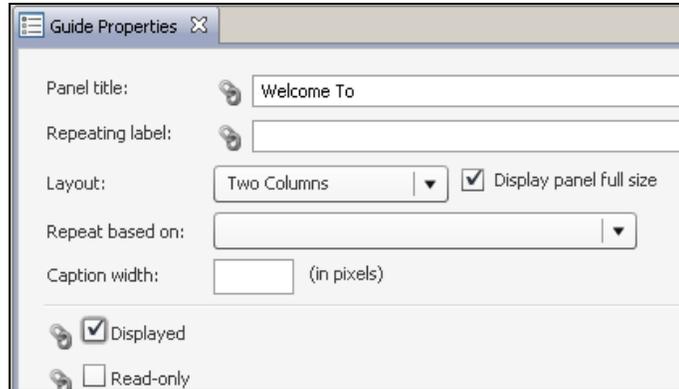
Task 2: Activate the Search Screen

The *Welcome and Search* screen has been turned off until this point. In this exercise, you will turn this screen on and configure it to use the new lookup data provided by the data model service.

1. In *Guide Tree*, locate the "Welcome" section, and select the "**Welcome To**" panel.



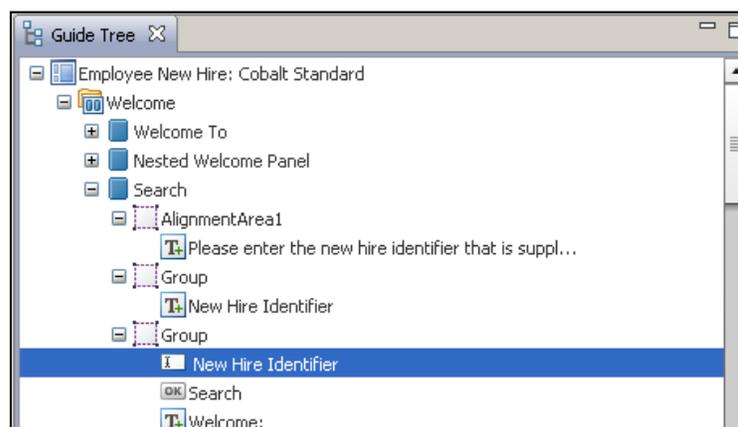
2. Note that the 'Displayed' property for the 'Welcome To' panel is turned off. Check the '**Displayed**' checkbox, so the panel will be visible at runtime (or Preview).



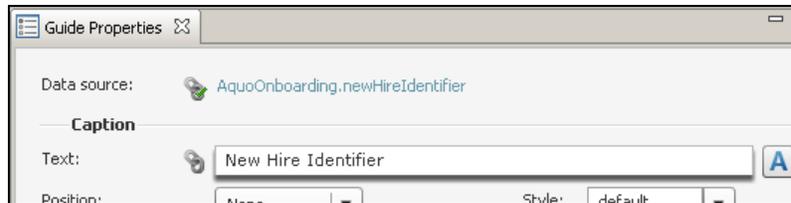
3. The 'Welcome To' panel is constructed using a technique called 'nested panels'. 'Welcome To' holds the 'Nested Welcome Panel', which in turn holds the 'Search' nested panel. The 'Search' panel contains the contents of interest to execute the search operation.

Due to time constraints, we do not have time to go into details of using nested panels. In short though, a nested panel acts like a control, and can be used in any location that a normal control can be used. They provide a significant amount of flexibility in building complex applications. Further information will be available on this technique in other forums.

4. Select the 'New Hire Identifier' field within the 'Search' panel.



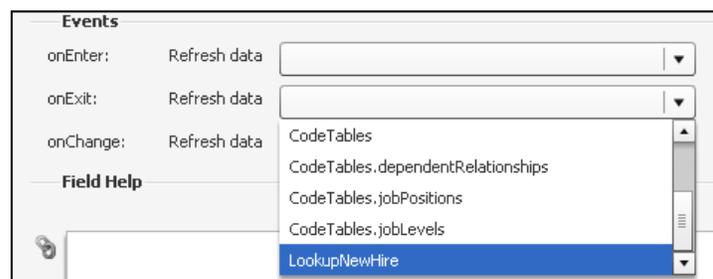
5. Verify that the 'New Hire Identifier' field is bound to the data source 'AquoOnboarding.newHireIdentifier' by checking the 'Data source' property in the *Guide Properties* view.



This is the field that was used to configure the input parameter to the web service call. The following sequence will occur when the user types into the 'New Hire Identifier' field.

- As this field is bound to the data item 'AquoOnboarding.newHireIdentifier', the data model will be updated with the value typed by the user.
- *AquoOnboarding.newHireIdentifier* is bound to the input of the web service call (see the properties for *LookupNewHire* to confirm this). The web service will see the modified data change.
- We have configured the service to only be triggered on an explicit event. If this were not done, the web service would update immediately, as it's input parameter has changed.

6. Ensure the 'New Hire Identifier' field is still selected in the *Guide Tree* view, and select '*LookupNewHire*' from the '*onExit*' event.



This event will trigger when the user exits the field. It causes the *LookupNewHire* data element to be refreshed, and since it is tied to the input parameter of the web service, the web service is executed.

7. Find the 'Search' button on the form. In a similar manner, add '**LookupNewHire**' to the button's `onActivate` event.
8. Save the guide
9. Test your search by running Preview.
 - Enter '**12345**', or '**54321**' as valid '*newHireIdentifier*' numbers and observe the results.
 - Enter a bad value, such as '**11111**'.

Note: *Record Not Found* will always display, and the *Welcome* line may display old data if a second search fails. We will fix this next...

Exercise 6: Creating a Dynamic Guide

In this exercise, you will add a "Rule" to the guide and use it to control the visibility of the "Record not found" field on the search screen. You will also modify an existing rule and configuration that controls the ability to dynamically display and hide other panels depending on the results of the search operation.

Objectives:

- By the end of the exercise you will know how to use rules to control the display of fields and panels. This allows the application to become dynamic and respond to variations in data.

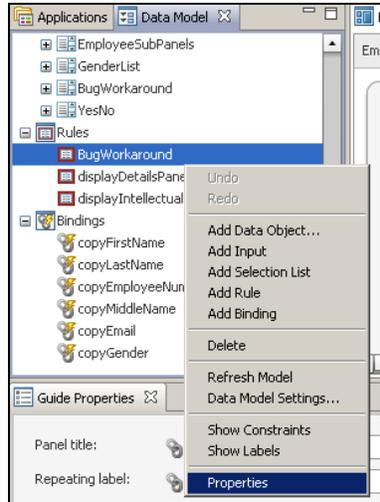
Assets Provided:

None.

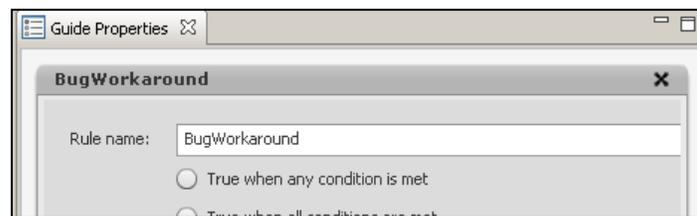
Bug Workaround:

Due to a bug in this version of Guide Builder, it is necessary to always go to the properties page of the 'BugWorkaround' rule before accessing the properties of any other rule. If you don't do this workaround, you may be presented with the property page for the incorrect rule and not see the correct data.

Right-click on 'Rules->BugWorkaround' and select it's properties.



Immediately close the properties page for the *'BugWorkaround'* rule.

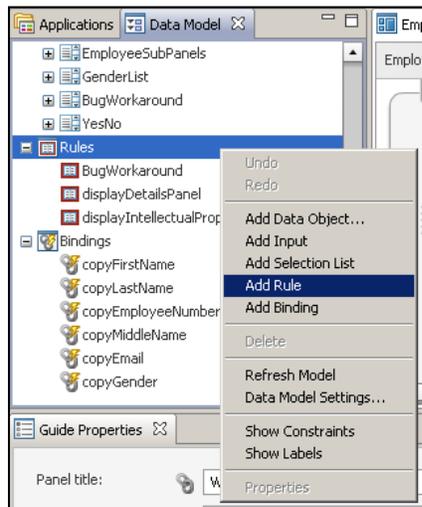


Tasks

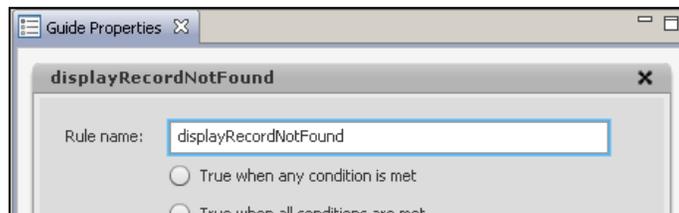
Task 1: Create and use a rule to control visibility of a field

In this task you will create a rule that contains the logic for whether or not the search has successfully completed. This rule will be used to control the visibility of the "Record Not Found" field.

1. Open the **EmployeeOnboardingGuide** if it is not already open. (Select the Guide Design perspective if necessary)
2. Open the 'BugWorkaround' properties, and close it again.
3. In the Data Model view, right-click and select '**Add Rule**'.



4. In the *Rule* property page, name the rule '**displayRecordNotFound**'. The name can not have any spaces.

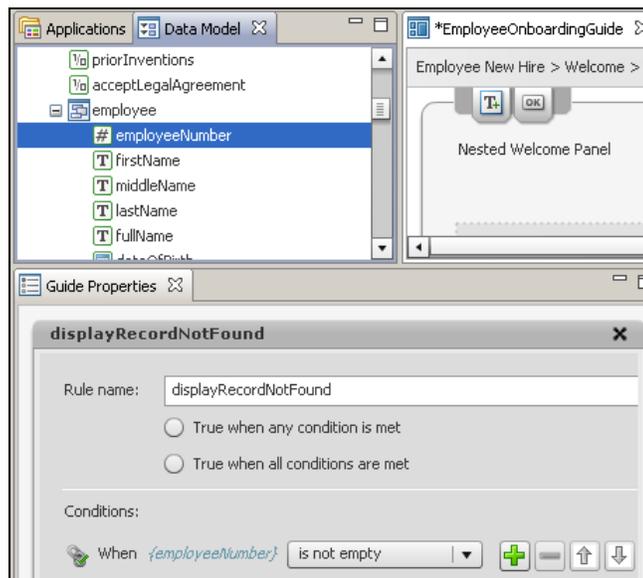


5. We need to create a condition to determine if the search has been successful. We know we have succeeded in the search, if the employee record has an employee number, as a successful search is the only way this data could be supplied. The following are the values for employee number that could be used:

-1 = the search failed.

0 = the search has never been executed. This is the default value of this data item.
>1 the search as succeeded.

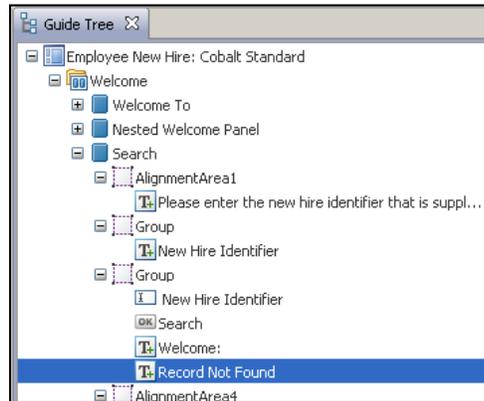
Drag '**AquoOnboarding->employee->employeeNumber**' from the *Data Model* tree onto the link button beside the first condition in the rule property page.



6. Set the operator to '**is equal to**' and the value to '**-1**'.

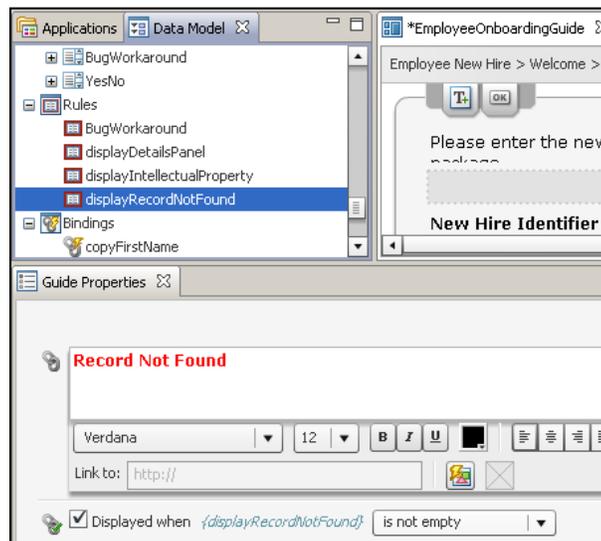


7. **Close** the property page.
8. Open and close the property page for 'BugWorkaround' so it is ready for the next use.
9. Select the '**Record Not Found**' field from the 'Search' panel in the *Guide Tree*.



10. In the corresponding *Guide Properties* view for this field, note that the '*Displayed*' property is checked. Therefore the field will always display, which is not what we want.

Drag the newly created '**displayRecordNotFound**' rule onto the 'Displayed' link for the Record not Found' field.



11. Set the condition to 'is true'



The result of binding the displayed property to the *displayRecordNotFound* rule, causes the field to only be displayed when the search fails and the employee number is -1.

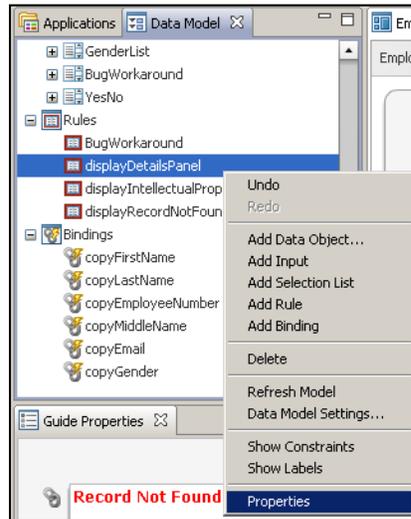
12. Save the guide.

Task 2: Modify the 'displayDetailsPanel' rule

A rule to display the initial 'Welcome' string, and the details panel is already present in the application. It is wired to the 'Displayed' property of all panels in the application. We only want these other panels to be displayed when the search is successful.

This rule has been deliberately broken to always display the panels. Lets fix it now.

1. In the *Data Model* view, find the 'displayDetailsPanel'. Right-click on it and select 'Properties' from the popup menu.



2. Note that this rule is also bound to the employee number, but the expression is not appropriate. Modify the expression so that it will be true when the employee number is **greater than 0** (which only happens if the search has been successful)



3. Save the guide.
4. Click on *Preview* to open the guide in the browser.
5. Type '12345' into the 'New Hire Identifier' field, click 'Search' and verify the welcome notice for 'John Jay Jacobs' is displayed.

New Hire Identifier

i

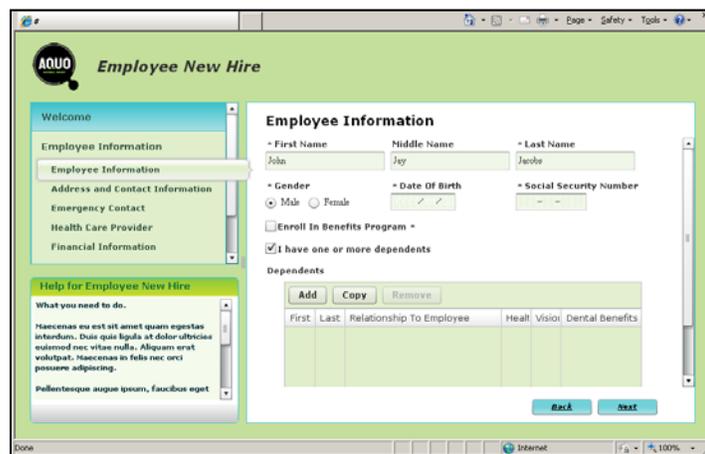
Welcome: John Jay Jacobs

6. Try '54321' and verify the record for 'Tony Blue' is found.
7. Type an invalid number, such as '11111', and verify 'Record Not Found' is displayed.
8. Keep the browser open.

Task 3: Explore other dynamic techniques

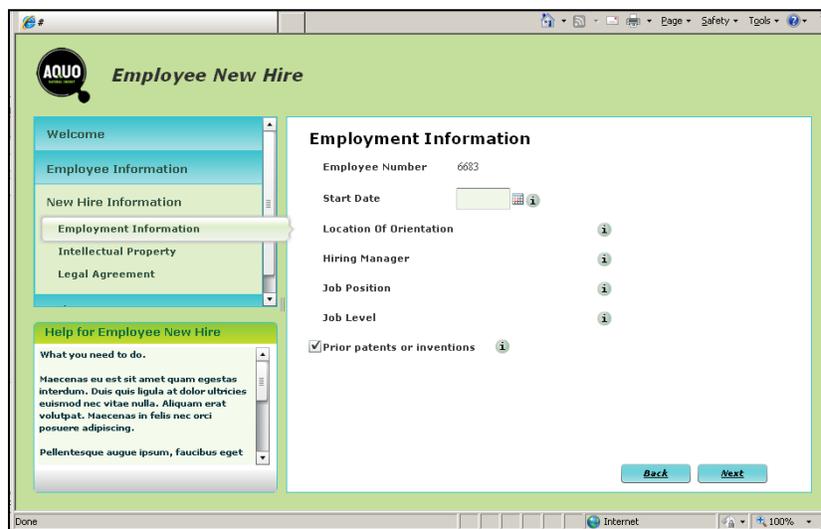
A rule to display the initial 'Welcome' string, and the details panel is already present in the application. It is wired to the 'Displayed' property of all panels in the application. We only want these other panels to be displayed when the search is successful.

1. In the preview guide running in the browser, enter the 'New Hire Identifier' **12345** to retrieve the record for John Jacobs.
2. Hit the Next button.
3. Ensure the *Employee Information* panel is selected in the guide navigator. Toggle the 'I have one or more dependents' check box.

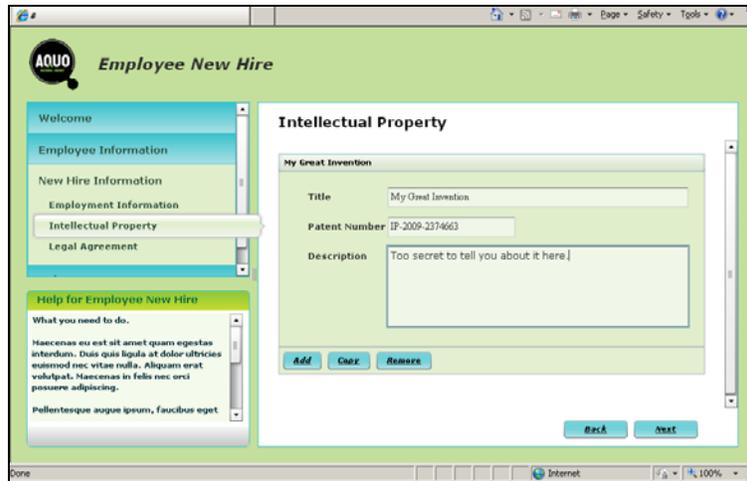


Note that the 'Dependents' panel is dynamically displayed on the screen.

4. Select the "New Hire Information" section of the guide
5. Select the "Employment Information" panel.
6. Toggle the "Prior patents or inventions" checkbox.



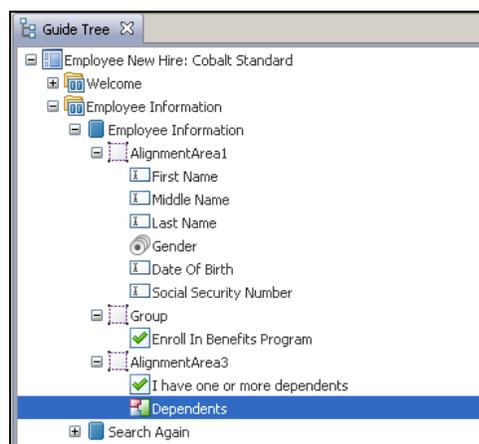
Note that the "Intellectual Property" panel is dynamically added or removed from the navigator section depending on the setting of this check box.



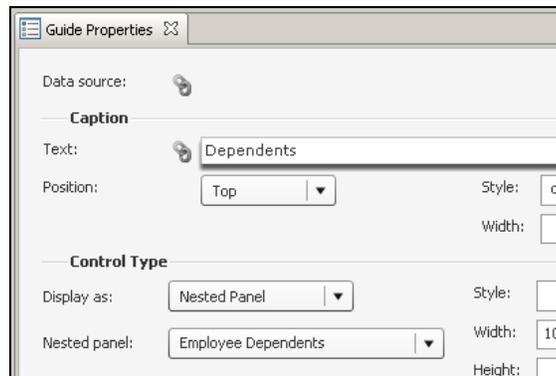
Task 4: Using nested panels and dynamic panels (optional exercise)

In this exercise you will be shown how to use nested panels to dynamically add a set of information to a guide.

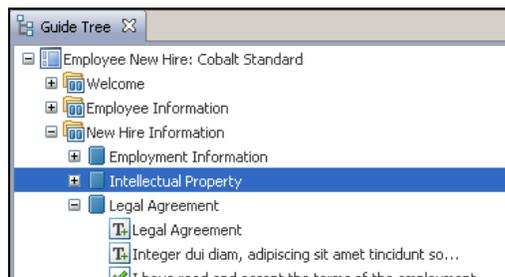
1. Open the *EmployeeOnboardingGuide* and select the **Employee Information** panel from the *Guide Tree*.
2. Select the **Dependents** field.



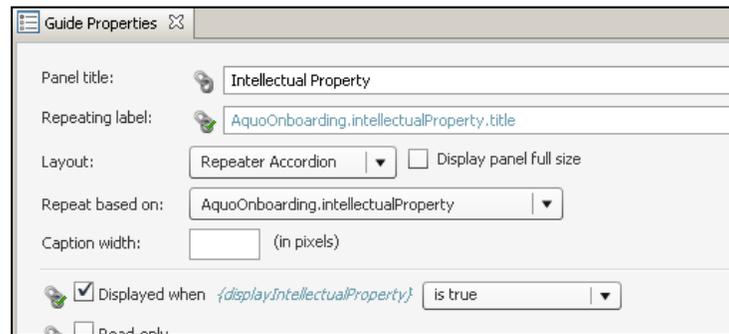
3. In the *Guide Properties* view, note that the control type is **'Nested Panel'** with the name of the panel that should be displayed at this location, in this case **'Employee Dependents'**.



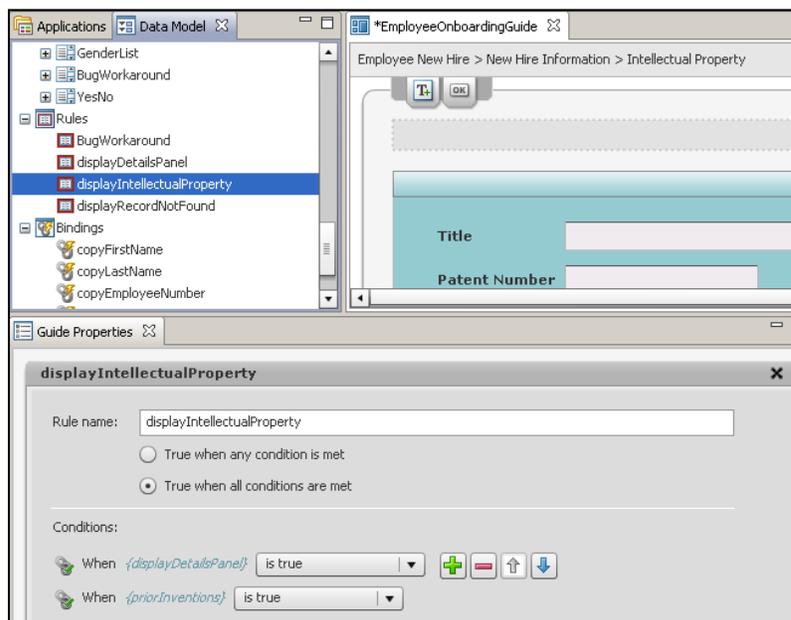
4. Select the **'Employee Dependents'** panel in the *Guide Tree* view. This panel is simply a normal panel, however in it's properties, the 'Displayed' property is unchecked. Therefore, the *'Employee Dependents'* panel will never be displayed in normal circumstances, however because it is used as a nested panel it is displayed inline as a control of another panel.
5. Select the **'Intellectual Property'** panel in the *'New Hire Information'* section of the *Guide Tree* view.



6. In the corresponding *Guide Properties* view, note that the 'Displayed' property is tied to a rule. This panel will only be displayed when the *'displayIntellectualProperty'* rule evaluates to true. This is the same technique as used for individual fields on a panel.



- Open the properties of the '**displayIntellectualProperty**' rule, and review the conditions that will cause this panel to be visible.



Exercise 7: Running the Guide in a Process (optional exercise)

In this exercise, you will run the completed guide within a browser and submit the data, launch *Workspace* and view the same guide and data in the *Workspace* environment, and again as a PDF within *Workspace*.

Objectives:

- At the end of this exercise you will have run your guide from a browser, outside of the development environment
- Your guide will be pre-populated with data
- Submitting data from the guide will trigger a LiveCycle user centric process. You will see the guide and data running from the *Workspace* environment.
- Submitting this guide will return another user task that shows the data in a PDF document.

Assets Provided:

Processes to run the guide from a REST endpoint, from within *Workspace*, and a PDF.

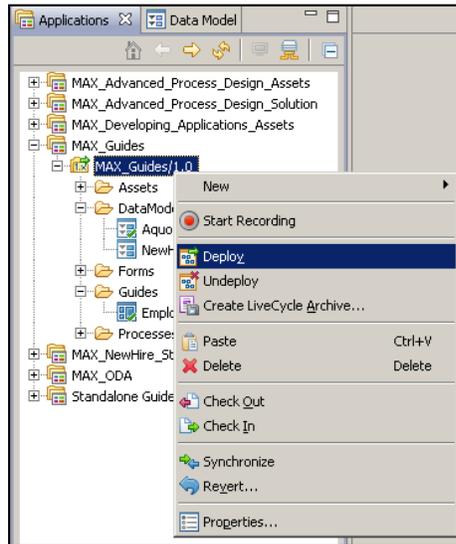
Tasks

Task 1: Run the Guide directly from a browser

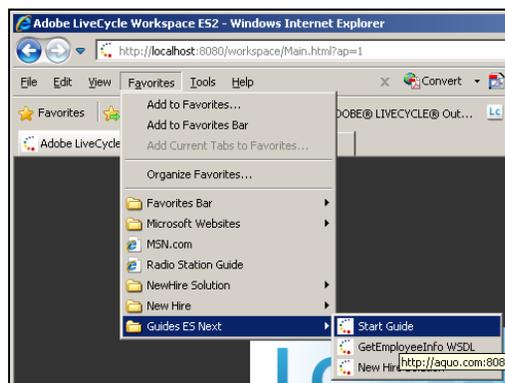
In this task you will run your guide from a browser.

1. Preview can be used on the development version of the application, but to run the application from outside of the Guide Design development environment the application needs to be deployed to the server.

Deploy the application by right-clicking on **MAX_Guides/1.0** in the *Applications* view, and select '**Deploy**' from the popup menu.



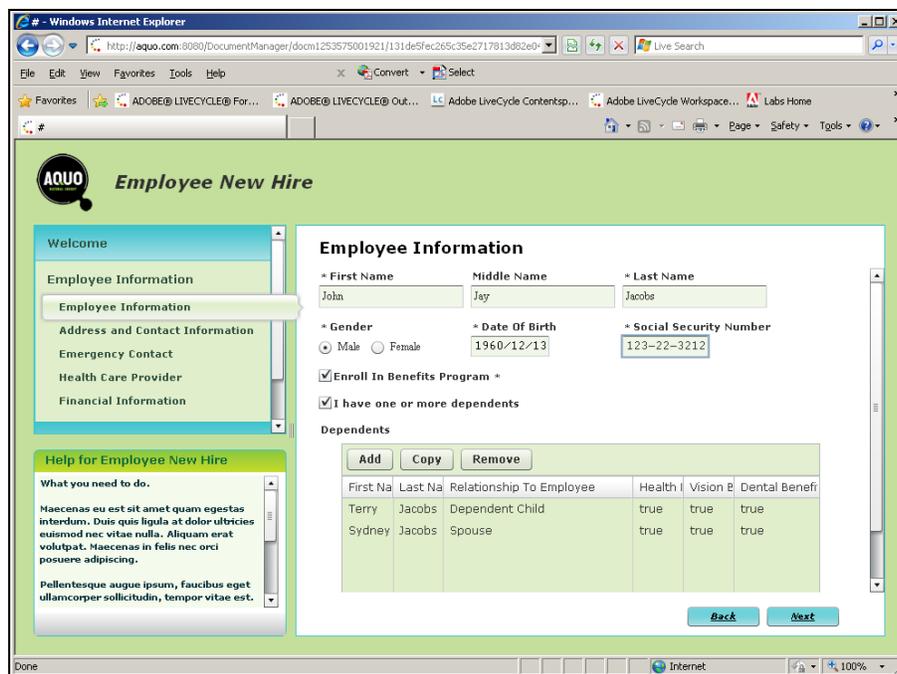
2. If prompted to save or check in files, select OK.
3. Open Internet Explorer and select the following link:
Favorites > Guides ES Next > Start Guide



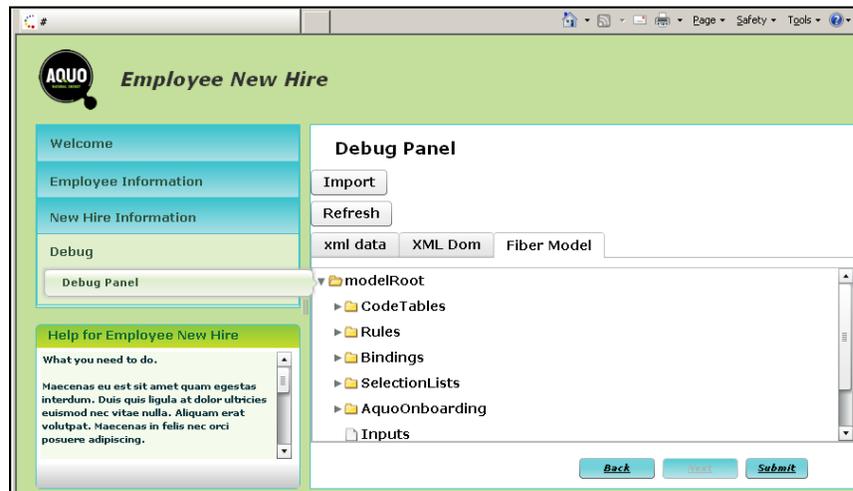
4. Notice that the guide is shown directly in the browser. It is similar to 'Preview' in the development environment, however, this can be run from any browser that has visibility to the LiveCycle server. This guide is also populated with data.

5. Enter a 'New Hire Identifier' of **12345** and click search. You should see the welcome notice for John Jacobs.
6. Click Next
7. Notice that most of the information on the screen has been pre-filled by LiveCycle. On the 'Employee Information' panel, enter the 'Date of Birth' and 'Social Security Number'.

Pick any date or number you like. The date is entered in 'yyyy/mm/dd' format.



8. Click the **Next** button and explore the panels in the application. Change any data as you see fit.
9. The last section and panel is the 'Debug Panel' that was introduced at the start of the lab. You can review all of the data that will be submitted in the 'xml data' tab. The 'Fiber Model' tab allows you to browse the data in a tree format.
10. Click the 'Submit' button that is visible when the debug panel is open. This will submit the data to LiveCycle.

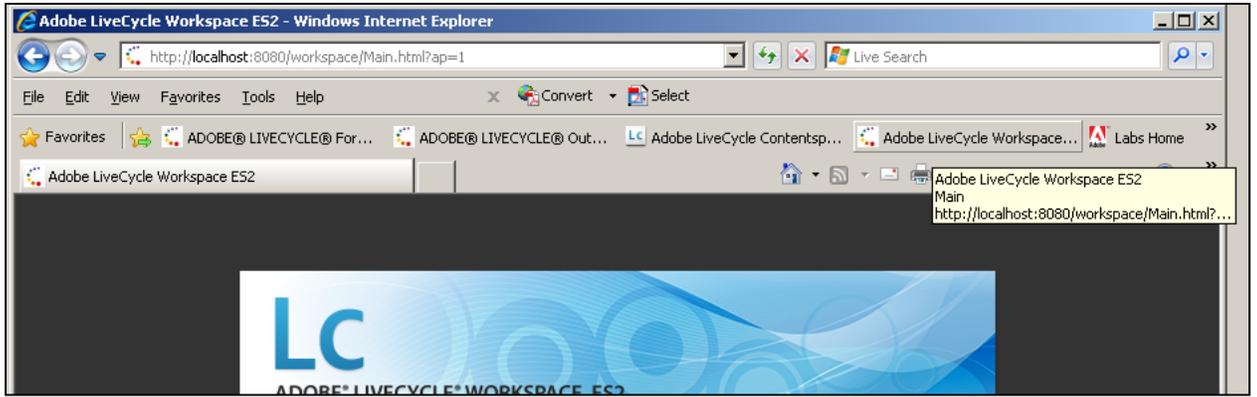


The "Submit" button is shown on the last panel of a guide. Normally, you would turn off the visibility of the Debug Panel, so this button would be on the previous panel. It is the intention of this finished guide to have the submit button on the 'Legal Agreement' panel.

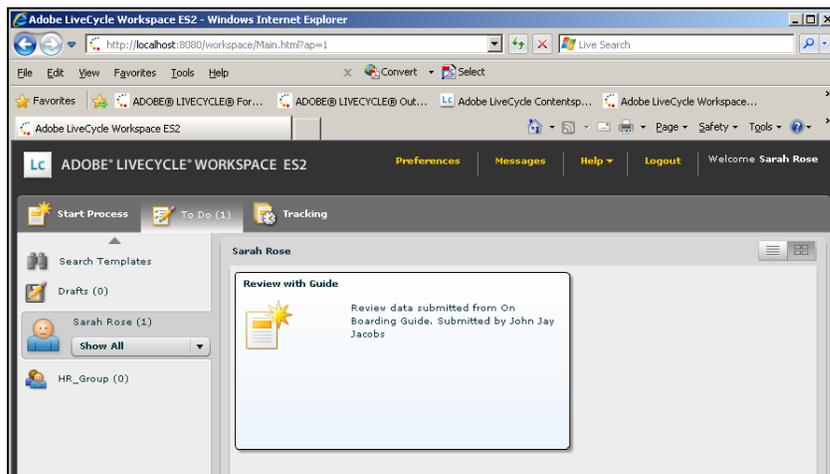
Task 2: Run the Guide from Workspace

In this task you will see the guide running within Workspace, including the data that was submitted from the browser in the previous step.

1. Launch Workspace by clicking the shortcut in the tool bar. The URL for workspace is: <http://aquo.com:8080/workspace>.

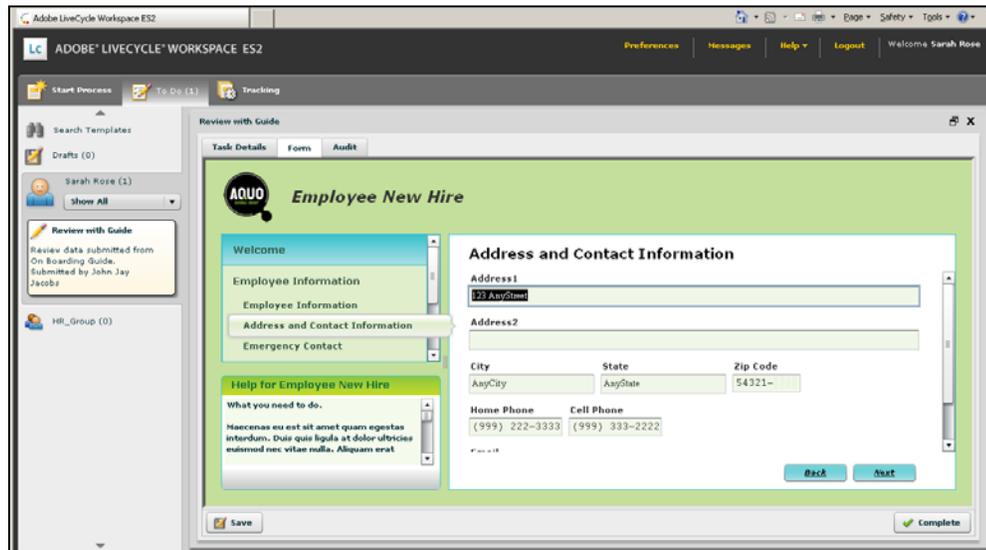


2. Log in to *Workspace* as Sarah Rose, the person that will review the entered data.
User: **srose**
Password: **password**
3. Notice there is a 'To Do' task in Sarah's work list. If you have this "To Do" list open when the browser submits the data, you will see the task card arrive.



Click the task card.

- The same form that was used in the browser is now displayed within Workspace. The data submitted from the browser is present.



- Bug: In this beta version of Guides, there is a bug that prevents the data information from being transferred correctly. You may need to reenter the data in the "Employee Information" panel.
- Click complete**, to submit this data back to LiveCycle.

Task 3: See the data in a PDF within Workspace

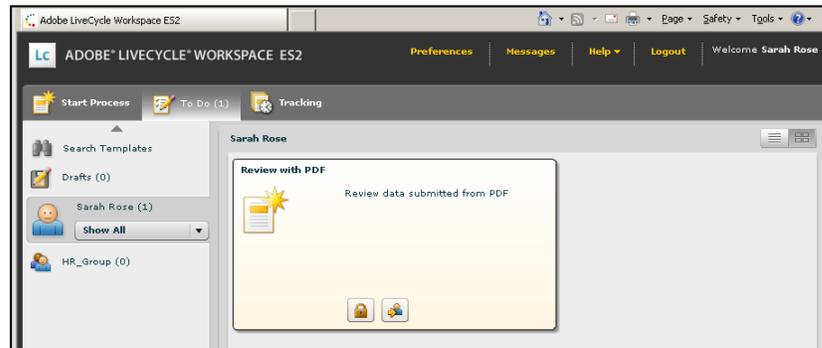
In this task you will see a PDF containing the data provided by the guide from the previous submission in Workspace

- Launch Workspace if it is not still open from the previous step.

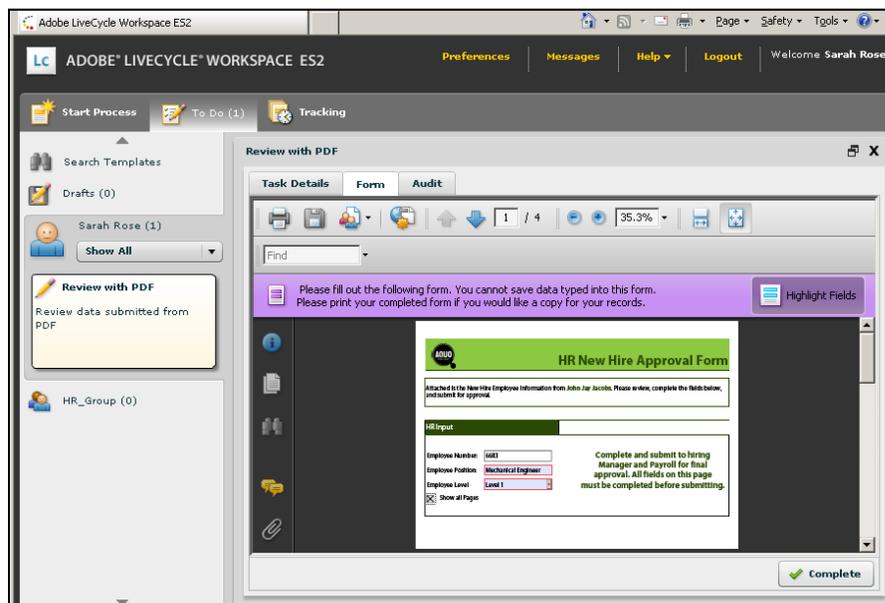
The PDF task will return to the same user that submitted the guide. This would not be a normal situation, but simplifies the demonstration. The PDF could be sent to any user of

LiveCycle according to the business requirements.

2. Click on the "Review with PDF" card to open the PDF in Workspace.



3. The document of record PDF contains the same information that was entered in the previous guide submitted by workspace. Click 'Show all Pages' to see all of the information in the PDF.



4. Click "Complete" to submit this PDF.