Exercise: Behaviors

This lab illustrates the process of how to implement basic behaviors that show how to include, exclude, and alter a page entity. The behavior file for the Workers Comp Quick Quote transaction uses many unconditional behaviors to differentiate the Workers Comp Quick Quote and the Workers Comp Full Quote transactions.

You can find the behavior file for these transactions in the training project's WEB-INF/workerscomp/behaviors directory.

- Open the workersCompBehavior.xml file.
- Look at the existing behaviors.
- Compare the page entities being exclude/included with the wcCommons.xml page library.

Remember that the path for the <for/> element is in the form:

```
<for>transaction_id/page_id/pageElement_id/fieldElement_id</for>
```

In many cases, you'll have the text shown in the browser for a given AgencyPortal UI element (the transaction name, the page title, the section legend, and/or the field label), so you'll search for this text to find the corresponding identifier (TDF XML id attribute value) to be used in behaviors or other references.

To find the desired transaction, page, section and/or field *id* for a given AgencyPortal UI element, you can use either of the following methods:

1. Eclipse (main menu): Search > File

   Choose the WEB-INF folder before invoking this, and use Scope = "Selected Resources"
   - In this way you're eliminating all or most of java, jsp and javascript artifacts
   - This allows for the item to be defined in any product (found in WEB-INF/<product> folders), including shared products (typically, WEB-INF/shared and WEB-INF/commercial).

   In most cases, the Page title, Section legend or Field label will be implemented as plain text in transaction files (found in WEB-INF/<product>/definitions/*.xml) or page libraries (WEB-INF/<product>/pagelibraries/*.xml)

   NOTE If the UI has been localized, the text you're searching for may be defined in a resource bundle rather than in TDF XML. The resource bundle text id (property name) specifies the tx.pg.sec.fld identifiers.

2. AgencyPortal Debug Console

- In some ways, this can be easier than trying to find the corresponding source artifact and searching within it
- This requires that you have the application running and an open workitem for the desired transaction
- In general, the Debug Console shows the runtime representation of a given artifact
  - This includes injection of resource bundle text
  - This can help avoid the need for a multi-step search "chain" to find the desired id
- "TDF" shows the current *runtime* representation of transaction *with behaviors applied for the current workitem*
  - That is, this shows only the pages, sections and fields that aren't excluded by behaviors (as evaluated w/r/t current workitem preconditions)
    - …unlike the transaction definition and page library source artifacts, which includes *all possible* pages, sections and fields, independent of any workitem data / preconditions + behaviors
    - …so if you can't see the desired item in the AgencyPortal (browser) UI, you won't find it here either
  - This will in effect combine the transaction definition and any imported pages into a single TDF transaction document
- "Navigation Menu" can be a slightly more convenient way to obtain transaction and page ids
  - Again, this shows the current *runtime* representation of transaction page menu *with behaviors applied for the current workitem; shows only the pages not excluded by behaviors*
  - The root node name is the transaction id
  - <rootTxId>.pages.page.@name values == page ids

Exercise 1: Exclude Billing/Contact Info for quickQuoteWorkersComp Transaction

1. In the application, start a new "Workers Comp **Quick Quote**" work item.
   Note that in the navigation menu, there is a "Billing/Contact Information" page. This is the page you are going to **exclude** from the Workers Comp transaction.
2. Open the WEB-INF/workerscomp/pagelibrary/workerscompCommons.xml file.
3. Search for "Billing/Contact Information" in the workerscompCommons.xml file.
4. Note that the id attribute for the <page> is "billingContactInfo". You are going to use this id to identify the page entity to which the behavior applies.

5. Open the WEB-INF/workerscomp/behaviors/workersComp**QQ**Behavior.xml file.
6. Add the following behavior to that file:

```
<behavior>
```

```
    <!--  Exclude Billing/Contact Info sections for quickQuoteworkersComp transaction -->
    <do action="exclude" />
    <for>quickQuoteWorkersComp/billingContactInfo/*</for>
</behavior>
```

…excluding all sections on that page.

7. Save and refresh the page (Ctrl-F5 in Internet Explorer).
8. Verify that the Billing/Contact Information page is READ-ONLY since the editable fields have been excluded in #6.

Exercise 2: Include the Billing/Audit Information section for the quickQuoteWorkersComp transaction.

9. Review the definition of the billingContactInfo page disabled in the previous exercise. Note that the first section (pageElement) is called "Billing/Audit Information". A new behavior will be created to include the "Billing/Audit Information" section in the Workers Comp **Quick Quote** transaction.
10. This pageElement does not currently include an id; an id is required so that the pageElement can be addressed by behaviors. Add an id attribute set to "billingAuditInfo".
11. Add the following behavior the workersComp**QQ**Behavior file, substituting the appropriate values for the <page_id> and <pageElement_id>:

```
<behavior>
   <!--
        Include Billing/Audit Info section on the Billing/Contact Info page
     page for quickQuoteWorkersComp transaction
   -->
   <do action="include" />
   <for>quickQuoteWorkersComp/billingContactInfo/billingAuditInfo</for>
</behavior>
```

12. Save and refresh the page.
13. Verify that the Billing/Contact Information page is available, but only the Billing/Audit Information section is available. This demonstrates that an include behavior takes precedence over an exclude. Note that the order of the exclude and include in the behavior file are irrelevant.
14. Try changing the exclude behavior added in the previous exercise, removing the trailing "/*" in the <for> spec. Save and review the new runtime behavior…

Exercise 3: Change default value for Experience Mod for both MA and RI instances of that field

PLEASE NOTE that field default values (both those specified in the TDF and with behaviors) are only applied under the following conditions:

- If you are viewing the page for the first time
- If you have previously viewed but never saved the page

- Immediately after the page was excluded by behavior and subsequently included (in effect, resetting the page and associated data; as if it had never been saved)

In all cases, the page will not be directly accessible in the transaction page menu (not a hyperlink in the list of pages on the left); in other words, for pages 2-n, if you can "jump" directly to the page by clicking on its name in the transaction page menu, no defaults will be applied to any field (OOTB framework functionality).

In the training application, create a new Workers Compensation work item. Create both MA and RI locations. Navigate to the State Rating Factors page. Note that there is no default value for the Experience Mod field for these states. Use a behavior to alter the default value for this field to "1.1" for these specific states by performing the following steps:

1. Open the WEB-INF/workerscomp/pagelibrary/workerscompCommons.xml file.
2. Search for "Experience Mod" in the workerscompCommons.xml file.
3. Note there are numerous instances of this field, inside a pageElement (fieldset or section) for each State, where the pageElement id includes the two-letter State code. Find the ids for the page, pageElement and fieldElement nodes to build up the behavior's <for /> element.
4. Open the WEB-INF/workerscomp/behaviors/workerscompBehavior.xml file.
5. Add the following behavior to that file, substituting the appropriate values for the <page_id> and <pageElement_id> values (Note: You want to affect both (ONLY) the MA and RI instances of the Experience Mod field.)

```
<behavior>
  <!--
      Change default value for Experience Mod for both
      MA and RI instances of that field
   -->
   <do action="alter">
      <property name="defaultValue">1.1</property>
   </do>
   <for>
      */<page_id>/<pageElement_id>/<fieldElement_id>
   </for>
   <for>
      */<page_id>/<pageElement_id>/<fieldElement_id>
   </for>
</behavior>
```

6. Deploy the update and create a new workitem with MA and RI locations; verify that the State Rating Factors page now has a default value of "1.1" for the MA and RI Experience Mod fields.

These fields do not include a uniqueId. The behavior for spec can use either the uniqueId or id. Can you explain some of the benefits of using a uniqueId?

Exercise 4: W3C XPath support

1. Add a precondition to make all Account Record fields on the Billing/Contact Information required when three or more Rating Classifications are selected.

2. Add a precondition to make all Inspection fields on the Billing/Audit Information required when the total of all Rating Classification exposures exceeds $1M.

3. Implement these using W3C XPaths in behavior XML only.

Q: How would these be implemented if W3C XPaths weren't available?

A:

Exercise 5: Behavior Precedence

Add a behavior that uses custom precedence to exclude all "Experience Mod" fields when Employers Liability Limit values all exceed $1M (1000000).

Assumptions:

1. This exercise assumes you have completed the Views lab to properly store the Employers Liability Limit data
2. This exercise assumes you have completed the Option Lists lab to expand the options to include > 1M; if not, use another numeric threshold satisfied by the available list, such as $500K (500000)

Q: How would this be implemented purely in behavior XML if this feature were not available?

A**:**

Exercise 6: Behaviors Targeted by Class

1. Disable the behaviors that control include/exclude of the MA (Massachusetts) section on the

State Rating factors page. Verify the section is always shown regardless of whether a MA location exists in the quote.

2. Add a behavior and update the page/section/field as necessary to include/exclude the MA section on the State Rating factors page by class rather than uniqueId/ field Id.

Q: How does this potentially improve your ability to maintain and update the state-specific rating factor pages/sections/input fields and associated behaviors?

A:

Q: How would a change in requirements, including new state-specific rating factor TDF elements, be addressed under the old vs. new implementation?

A:

Exericse 7: Default / Global PreConditions

In the Personal Auto LOB:

1. Disable all include/exclude behaviors that reference the "IsThereACoApplicant" precondition or "CoApp" fields. Verify the CoApplicant fields are always shown regardless of the answer to the "Is there a co-applicant" input field before moving on to the next step.

2. Add a new behavior file that implements the equivalent of the conditions disabled in the prior step, but instead uses a global IsThereACoApplicant precondition. Test to verify the CoApplicant sections/fields are back to conditionally show based on the answer to the "Is there a co-applicant" input field before moving on to the next step.

3. Add a new precondition in the new behavior file created in the previous step to make the CoApplicant Work Phone Number required when the Controlling State is MA. Test to verify.

Q: What are the potential benefits to using this feature? Are there any potential drawbacks?

A:

If you were also tasked with adding a precondition to make the Applicant Work Phone Number required when the Controlling State is MA, where would this go? Would you consider changing how the

equivalent CoApplicant behavior was implemented a) if both requirements were provided together b) if one of these was added after the first had already been implemented and tested?

Exercise 8: Logical OR'd PreConditions

In the Workers Comp LOB:

On the Policy Information page, make the "Dividend Plan / Safety Group" field required if the Controlling State is MA or RI or the SIC Code indicates "Mining".

Q: How would this be implemented if this feature weren't available?

A: