

Exercise: APXPaths

To see an example of how the workitem XML document is updated using APXPaths on the back end as data is entered on the front end (in the browser), while creating a new workitem, for each page of data, we'll retrieve the XML via the debug console and save it to a file then compare them.

Although we won't be looking at the APXPaths directly during this exercise, we'll see the net result of their use when writing workitem data. We'll work directly with APXPaths- not just their results- in later modules.

Lab

Prerequisite – create a new folder in your Eclipse portal project for the page-by-page data

Right-click on the root of the project, choose New > Folder, and name it "APXPathsLab"

Every time you transition to a new data page, you will be viewing the XML in the debug console and saving it to a file.

For the filenames, use a sequential number starting with 0, then a dash, then the page name (feel free to abbreviate, but try to be consistent [you'll be visiting some pages multiple times]).

We start with zero to reflect the initial state; the first time we save the XML to file will be immediately after creating the new workitem, *before* saving any data on the first page.

How to save the XML from the AgencyPortal (debug) Console

Immediately after you get to a new page (before changing data and choosing Continue or Save)

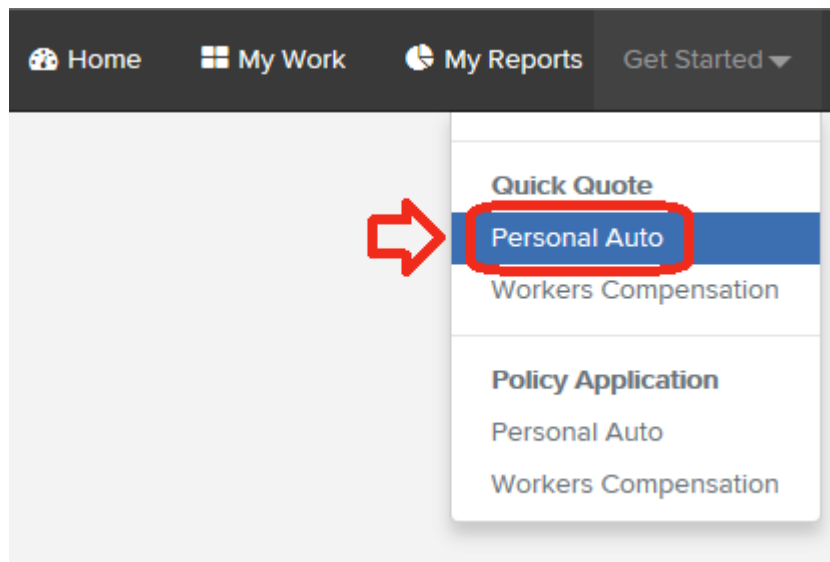
- Go to the debug console "AgencyPortal Console" browser window
- choose "Current ... XML " in the dropdown
- Right-click and choose Select All; Right-click and choose Copy
- In your Eclipse project, Right-click on the APXPathsLab folder and choose New > File
- Enter the filename as previously instructed: <sequential#>-<page>.xml
- The file will automatically open in an editor tab...
- Paste in the XML acquired from the debug console and save the new file content

- Depending on which browser you use, there may be extraneous characters included when you copy from the debug console; this is okay, we're not so concerned about valid XML for this exercise, just the overall structure of the document
- Compare each saved file to the previous one
 - Choose the current file in the project tree
 - Ctrl-Click on the previous file
 - Right-Click and choose Compare With > Each Other

Your instructor will walk you thru this process for the first few pages...

Create a new workitem

Use the AgencyPortal to create a new **Personal Auto Quick Quote Transaction**:



For the first "pass", only fill out the required fields.

Unless otherwise specified below, the data values don't matter...

- On the Agency/Applicant Information page, choose "No" for Is there a Co-Aplicant
- On the Drivers page
 - Enter only one driver
 - Leave the Student over 100 Miles, Good Student, and Driver Training fields set to No

- Don't enter any Incidents
- On the Vehicles page
 - o Change all the optional coverages (below Vehicle Usage) to a value other than No Coverage

After you get all the way to the Summary page, go back and make the following changes, saving a new xml file after each change (after entering/updating the data and hitting Continue):

- On the Agency/Applicant Information page, choose "Yes" for Is there a Co-Applicant and enter the associated data
- On the Policy Coverages page, switch to any other Liability Coverage type (first field) and enter the associated data
- On the Drivers page
 - o Edit the existing Driver and change each of Student over 100 Miles, Good Student, and Driver Training fields to Yes; Save
- On the Vehicles page,
 - o For all the optional coverages (below Vehicle Usage) switch back to No Coverage

Review

See if you can answer the following questions

- Why isn't the XML document initially empty (before you've entered any data)
 - o What can you infer about the schema from this?
- Why is some of the data inserted (in the middle of the document) rather than appended (added at the end)?
- Why doesn't the order of the data in the document match the order of the fields in the browser?
- How is the association of driver and vehicle (as defined on the Driver Vehicle Assignments page) stored?
- Bonus Question: Why is it important that the entire Coverage node is removed after you switch the vehicle back to minimal coverage?



duckcreek
technologies

- That is, it doesn't remove *just* the Deductible or Limit values
- Hint: this is as important from an insurance business rules perspective as it is from a data perspective.