hexure.

Distributor to Carrier Mapping Guide

firelight.

hexure.

Hexure
Two South Cascade Avenue
Colorado Springs, CO 80903
Phone: 719.442.6400

Internet E-Mail: info@hexure.com
Website: hexure.com

Table of Contents

1   Distributor to Carrier Mappings

Insurance Technologies is improving the way we implement Distributor to Carrier Mapping for speed and elimination of redundant mapping. The master Distributor to Carrier Rule Set and a Master Mapping Form will now live and be maintained in the carrier environment and deployed to distributor environments. Insurance Technologies team is creating and testing the initial Master Mapping RuleSet with the Master Mapping Form for each carrier at no cost to the carriers. All new distributor implementations will be using this same process with custom distributor values in a distributor RuleSet.

The benefits to adding the Distributor to Carrier ruleset are:

- One RuleSet to set up and maintain

- Reuse of RuleSet via deployment for all new distributors

- Eliminates redundant mapping per distributor

- One spot for breakage, increasing speed for corrections

- Eliminates the need for a distributor implementation to map to carrier forms

- Quicker distributor implementations

- Maintenance of mapping is greatly reduced

- Carriers can deploy updates to distributors on their own

- No cost to the carrier to implement the initial Distributor to Carrier Mapping

New and Improved Process

Distributor to Carrier Mapping will now reside in the carrier package and deployed to distributors.
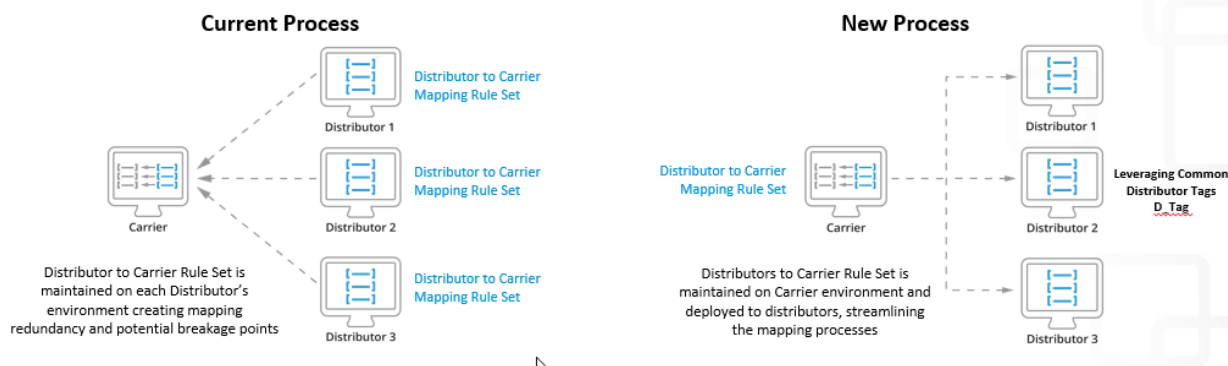
2 Common Tags

All new clients should be using the common tags and list values that are loaded into the Admin Help. This allows for more efficiency during the development timeframe.

See also the following Spotlight videos about Common Tags, which is also available from the Admin Help.

- Common Tags: https://vimeo.com/333372962/a2b1765093

- Common Tag Conversion Process: https://vimeo.com/353873388/cce4001558

NOTE: To find the common tag spreadsheets, refer to the "Common Tags" topic within the FireLight Admin Help. This can be found under the Help button.

On the Admin Help dashboard, click the **Designer** tab, then select **Common Tags**.

FireLight®

Getting Started Product Config Designer Organization Config Utilities Support Features Release History

Common Tags
Differences between Combined Wizard+Forms UI and Legacy Forms UI
Form Designer »
Form Library »
Wizard Designer »
Wizard Library »

Welcome to Firel

Search

FireLight Admin Help

Release History

Getting Started

Spotlight Feature Videos

PDF Guides

Admin User Guide | Platform User Guide | Configurable Pre-Sale Guide | Console for Windows Guide | eApp™ Guide for iPad

E-Delivery Guide | E-Signature Guide | How FireLight Meets Legal Requirements | Helpful Implementation Options | Review Queue Guide

---

FireLight®

Getting Started Product Config Designer Organization Config Utilities Support Features Release History

Search

You are here: **Designer** > *Common Tags*

Common Tags

NOTE: For a Spotlight Feature Video about common tags, click the link: FireLight Common Tags

FireLight provides a list of common tags delineated by group (e.g. Owner, Agent) to be used in Form and Wizard fields. The use of common tags ensures data is mapped correctly throughout the system, for example, inbound data, input controls, CRMs, etc. It also saves you from having to re-enter data when creating a new activity from another activity or when switching products.

There are two ways to locate a common tag: using the autocomplete function while editing a field's properties or by searching through the CommonTags.xlsx file below.

Click the CommonTags.xlsx link to view common tags. The file is organized by group name (e.g. Owner, Agent). Click the CommonTags_ListName link to view common tag list name values for options within a list (e.g., state and country codes on the Address tab).

CommonTags.xlsx

CommonTags_ListName.xlsx

In support of the common tags, there is an autocomplete function available in the Field Property "Name" in both forms and wizards. It is a "contains" search so you can type any portion of the common tag name and a list containing the typed text will appear. Continue typing to refine the list. **NOTE:** If the list does not appear, make sure your spelling is correct, you may also consider searching for a different portion of the tag name.

Click the image below for an example of adding Agent_FirstName in the Name property.

Designer
Common Tags
Differences between Combined Wizard+Forms UI and Legacy Forms UI
Form Designer
Form Library
Wizard Designer
Wizard Library

hexure

On the Common Tags page, two hyperlink options are available which show all of the allowed common tags and list profiles that are associated with the common tags.

- CommonTags.xlsx hyperlink opens a Microsoft Excel spreadsheet with all common tags that FireLight uses throughout the system. The spreadsheet contains all common tags and is broken out by FireLight Base, Illustration, Annuity, Life, and roles in the tabs at the bottom of the spreadsheet.

- CommonTags_ListName.xlsx hyperlink opens an Excel spreadsheet with all common list values that are available for the associated common tags. The spreadsheet contains the common list values by Party, Address, Product, Annuity and Life.

3   Download Tag List

Below is the process for generating a Download Tag List, followed by the Generate Mapping Ruleset process, which are used for Common Tag mappings. These processes can be found in the FireLight Admin Help on the "Utilities" tab.

> **NOTE:** Be sure to enable the Download Tag List utility. Go to the **Organization Configuration/Groups/Admin** tab and select the **Enable View Utilities** check box. This enables all utilities for your organization.

Download Tag List is the first of four utilities for converting existing data item tags in forms, wizards, rule sets, packages and the provider to new tags (provider conversion is done only by the Insurance Technologies FireLight team). This utility is especially useful in converting existing tags to common tags.

Additional Information about Common Tag Conversion

See the "Common Tag Conversion Workflow" topic in the Admin Help for details about common tag conversion workflow including initial QE/UAT development and testing, Staging and Production deployment and testing, and after-deployment scenarios for create new activity, existing active activity, and existing locked activity.

3.1   Step 1. Download Tag List

1.   On the **Home** page or navigation bar, click **Utilities/Download Tag List** to go to the Download Tag List page.

2.   In **Package**, select a package option from the list. (This is a list of Packages that are on the Product Configuration/Packages page.)

> **TIP:** You could add all forms, wizards and rule sets to a "tag conversion" package to convert all your forms, wizards and rule sets in one run. Then move the new converted forms, wizards and rule sets to the applicable packages after verifying the converted tags. This would produce a very large spreadsheet for mapping data items, but would ensure that all form, wizard and rule set tags were converted at the same time.

3.   (Optional) Select the **Generate Non-Matching Common Tags** check box to generate a list to include only the data items and export values that are not common tags in FireLight. You are still able to pull all values simply by not selecting the check box.

NOTE: If you have common tag names with export values that you want to change, do not select this check box so the existing common tag name and export values will be included in the spreadsheet.

4. Click the **GENERATE PACKAGE TAG LIST** button.

    FireLight searches through all forms, wizards, and rule sets within the selected package and generates a spreadsheet in .csv format with all found data items and export values.

5. On the information bar at the bottom of the screen, choose an option to **Open** or select a **Save** option. Be sure to save the mapping .csv spreadsheet. You will need it for the next step in the tag conversion process.

3.2   Step 2. Create the Mapping Spreadsheet

1. Open the mapping spreadsheet created in step 1.

    Rows 1 – 4 contain general information about the product (e.g., Carrier, Package, Product Type, Product). **NOTE:** Product Type and Product fields are currently not supported for tag conversion.

    Row 5 contains the column titles.

2. Add new tags and List Values to the mapping spreadsheet using the following as a guide and save the mapping spreadsheet.

    Group – the Group column is used to separate the Rule Mapping xml into multiple sections. It is not necessary to separate tags into different groups, but it can make things cleaner.

    From Dataitem – current data item tag name. This column is filled in when the .csv spreadsheet is created.

    From Value – current name and export value (List Value) in the field properties for the list option if the data item has options (dropdown list, radio buttons, etc.).

    NOTE: To assign multiple dropdown value mappings, add a duplicate row for each dropdown item, using the same From Dataitem and To Dataitem for each row, then populate the From Value and To Dataitem Value columns with the corresponding dropdown values. The To Dataitem Value contains the option name followed by a pipe character (|) List Value (export value). See the CommonTags_ListNames.xlsx in the Admin Help, "Common Tags" topic.

Page 9 of 40

To Dataitem – new data item tag name. See the CommonTags.xlsx in Common Tags.

To Dataitem Value – new name and export value for the list option if the data item has options (dropdown list, radio buttons, etc.).

info – additional information

4   Generate Mapping Ruleset

Below are the steps to generate the mapping ruleset.

1.   On the navigation bar, click **Utilities/Generate Mapping Rule Set**.

2.   Under **Upload Mapping Spreadsheet**, drop the mapping spreadsheet or click in the box and select the mapping spreadsheet.

3.   The mapping spreadsheet opens below the box. Click the **PREVIOUS** and **NEXT** buttons to move through the pages of the spreadsheet to verify the mappings from old to new tags.

4.   Below the mapping spreadsheet under **Rule Mapping Options**, select the applicable check boxes.

   Use getcontrolstate check – Checks if a data item is enabled or disabled.

   Use isnullorwhitespace check – Checks if a data item is null or contains just whitespace characters.

   Set the rule scope to "Property Change" (instead of "All")

5.   Click **BUILD MAPPING**.

6.   The rule mappings open below. Scroll down the rules to view and verify the mapping rules.

7.   In **Enter Rule Set Name** enter a mapping rule set name for the rule set and click **SAVE RULE SET**.

8.   Click **OK** on the confirmation Message dialog box.

   FireLight adds the mapping rule set to the grid on the Product Configuration > Rule Sets page.

9. On the navigation bar, click **Product Config > Rule Sets**. Select the approved states in the **State Approved** grid.

10. Add the mapping rule set to the corresponding product package on the **Product Configuration > Packages** page.

5 Common Tags for Distributors

The new distributors will all be built on the common tags with "D_" value added in front of each data item. This D_ is put in place on all Distributor common tags in the event both distributor and carrier have like rules. All distributor implementations will be tagged with the D_ in front of the existing common tag to prevent any rule issues. Example: D_Owner_FirstName.

5.1 List Values Grouping

All list values found in the CommonTags_ListName.xlsx in the FireLight Admin Help are required to be used when mapping from Distributor to Carrier. To locate the proper list value, access the **CommonTags.xlsx** spreadsheet found in the FireLight Admin Help.

The groups for the List values can be found in column D as highlighted below.

Once the proper group name is located, refer to the **CommonTags_ListName.xlsx** spreadsheet in the FireLight Admin Help.

	A	B	C	D	E	F	G
1	ListName	ACORDListName	ListValue	CodeValue	Caption	Abbreviation	Description
2	List_Gender	OLI_LU_GENDER	1	OLI_GENDER_MALE	Male	M	
3	List_Gender	OLI_LU_GENDER	2	OLI_GENDER_FEMALE	Female	F	
4	List_Gender	OLI_LU_GENDER	3	OLI_GENDER_UNISEX	Unisex	U	
5	List_Gender	OLI_LU_GENDER	5	OLI_GENDER_INTERSEX	Intersex		
6	List_Gender		0		None		
7	List_Gender		-1000		Other		
8	List_MaritalStatus	OLI_LU_MARSTAT	1	OLI_MARITAL_MARRIED	Married		
9	List_MaritalStatus	OLI_LU_MARSTAT	2	OLI_MARITAL_SINGLE	Single		
10	List_MaritalStatus	OLI_LU_MARSTAT	3	OLI_MARITAL_DIVORCED	Divorced		
11	List_MaritalStatus	OLI_LU_MARSTAT	4	OLI_MARITAL_WIDOWED	Widowed		
12	List_MaritalStatus	OLI_LU_MARSTAT	5	OLI_MARITAL_SEPARATED	Legally Separated		
13	List_MaritalStatus	OLI_LU_MARSTAT	6	OLI_MARITAL_COMMONLAW	Common Law		
14	List_MaritalStatus	OLI_LU_MARSTAT	7	OLI_MARITAL_ESTRANGED	Estranged		
15	List_MaritalStatus	OLI_LU_MARSTAT	10	OLI_MARITAL_DOMESTICPARTNI	Domestic Partner		
16	List_MaritalStatus	OLI_LU_MARSTAT	11	OLI_MARITAL_CIVILUNION	Civil Union		
17	List_MaritalStatus		0		None		
18	List_MaritalStatus		-1000		Other		
19	List_EmploymentStatus	OLI_LU_EMPSTATTYPECODES	1	OLI_EMPSTAT_ACTIVE	Active		
20	List_EmploymentStatus	OLI_LU_EMPSTATTYPECODES	2	OLI_EMPSTAT_NOTACTIVE	Not Active		
21	List_EmploymentStatus	OLI_LU_EMPSTATTYPECODES	3	OLI_EMPSTAT_DISABLED	Disabled		
22	List_EmploymentStatus	OLI_LU_EMPSTATTYPECODES	4	OLI_EMPSTAT_LEAVEOFABSENCE	Leave of Absence		

5.2 List Names

Column A, titled ListName, shows the group names of the list values. Column C shows the List Value that needs to be used in the mapping rule for that specific Common Tag. If a client needs to view the List Value caption or abbreviation, a pipe is added to the List Value in the field and the rules of that Common Tag.

Example: Owner_ResidentialAddress_State Common tag is used for the state of AL. The proper List Value would show as AL|1 as 1 is the proper List Value for AL.

6 Master Form for Testing

A master form with Distributor Common Tags will be used for testing. This form will act as the distributor and allow the ability to test all Distributor to Carrier Mappings.

The Master Form that can be used for testing will be in the shared drive:

P:\Engineering\Firelight\Firelight Forms\Master mappings.

Example of a Master Mapping Form:

To use the Master Mapping form effectively, the user will add the master mapping form and Distributor to Carrier ruleset to a selected package within the Carrier's environment.
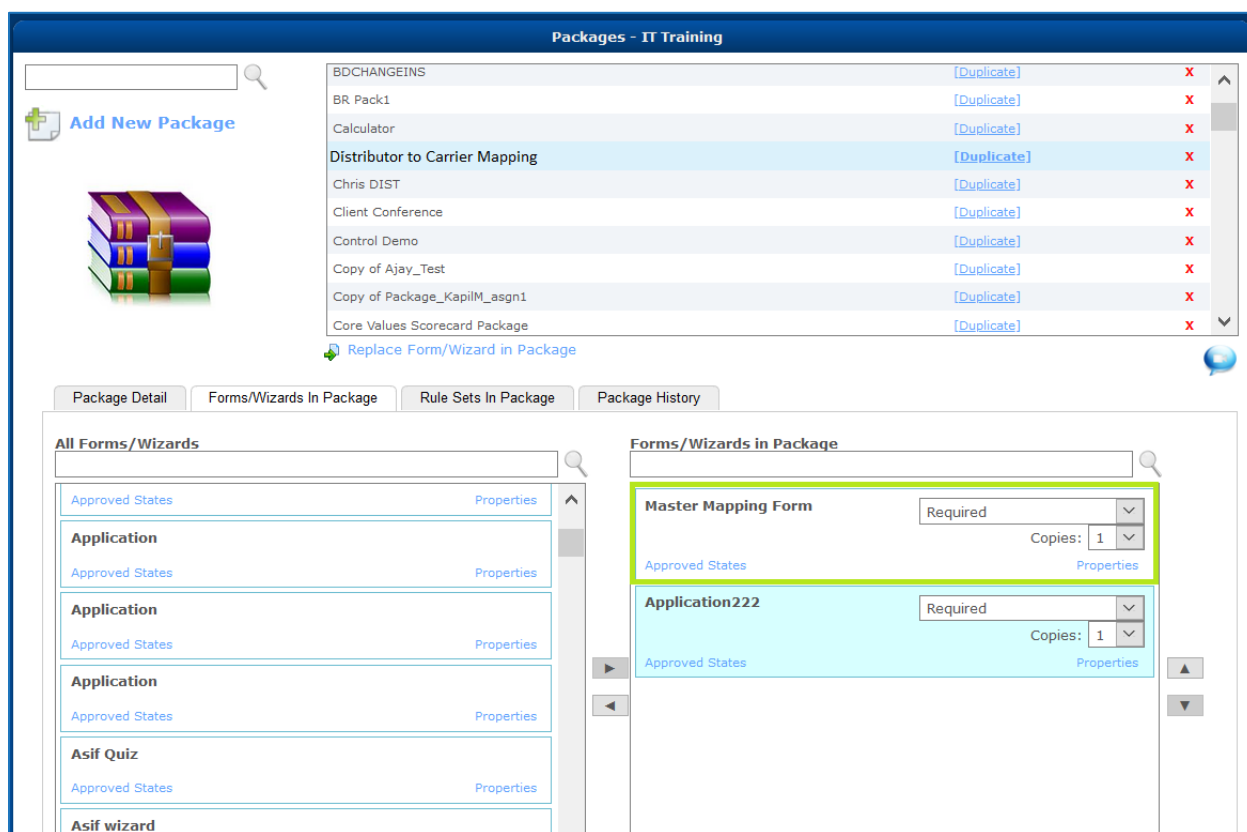
The Master Mapping Form acts as the Distributor in testing, and allows developers to create Distributor to Carrier Mapping rules and test without deploying to an active distributor.

7   Individual Rules in One Master Rule Set

All mapping rules will be broken out individually. This is to ensure that errors within rules that have conditions and do not exist in the carrier, do not break the rule that is generated. These rules will be added into one rule set called Carrier Distributor Mapping.

The process to convert Mapping rules into individual rules are as follows:

1.   Generate the download tag list by clicking the **Generate Package Tag List** button in the Download Tag List screen:

Select Package to generate Common Tags from

Package
Chris Common Tag Training

☐
Generate Non-Matching Common Tags

GENERATE PACKAGE TAG LIST

Firelight enables an Excel spreadsheet with all download tags from the selected package.

2.   The highlighted Group in column A is where the user can group the rules that will be generated by the conversion tool. Each Group name must be unique for FireLight to separate the rules for that field correctly as shown in the example below:

	A	B	C	D	E	F	G
1	Carrier	IT Training					
2	Package	Chris DIST					
3	Product Type						
4	Product						
5	Group	From Dataitem	From Value	To Dataitem	To Dataitem Value	info	
6	Owner First	D_Owner_FirstName		Owner_FirstName			
7	Owner SSN	D_Owner_SSN		Owner_ResidentialAddress1			
8	Owner Phone	D_Owner_Home_PhoneNumber		Owner_SSN			
9	Owner DOB	D_Owner_DOB		Owner_DOB			
10	Annuitant SSN	D_Owner_Annuitant_SSN		Owner_Annuitant_SSN			
11	Owner Gender	D_Owner_Gender	M|1	Owner_Gender	1		
12	Owner Gender	D_Owner_Gender	F|2	Owner_Gender	2		
13							
14							

3.   Once the desired Group names are in place along with the "From Dataitem" and "To Dataitem" tags, the user saves the Excel file and proceeds to the mapping.

4.   The user will then select the appropriate option from the utilities button. Generate Mapping ruleset will be selected.

Generate mapping ruleset maps the From Dataitems in column B to the To Dataitems in column D in the Excel file.

5.   In the Generate Mapping Ruleset screen, the user uploads the corrected Excel file. Under **Rule Mapping Options**, the user selects the check boxes for **Use getcontrolstate** check and **Set the rule scope to Property Change (instead of All)**.

Although the mapping rule is generated as one rule, the rule itself is coded to show as separate rules. The user will be able to see the rules broken out by identifying the <ScriptCode> and </ScriptCode>. This signifies the beginning of one rule and the end of that rule. Below are highlighted examples of broken out rules.

Rules broken out example:

<ArrayOfRuleDef xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <RuleDef>

   <ScriptCode>

    <if>

     <condition>

      <compare op="==">

       <const value="IT Training" />

       <diget dataitemid="FLI_PRODUCT_CARRIER_NAME" />

      </compare>

     </condition>

    <if>

     <condition>

      <getcontrolstate dataitemid="Owner_FirstName" />

```xml
        </condition>
        <diset dataitemid="Owner_FirstName">
          <diget dataitemid="D_Owner_FirstName" />
        </diset>
      </if>
    </if>
  </ScriptCode>
  <Version>2.17.195</Version>
  <RuleName>IT Training: Owner First</RuleName>
  <Class>PropertyChange</Class>
  <LastUpdate>6/2/2020 2:37 PM</LastUpdate>
</RuleDef>
<RuleDef>
  <ScriptCode>
    <if>
      <condition>
        <compare op="==">
          <const value="IT Training" />
          <diget dataitemid="FLI_PRODUCT_CARRIER_NAME" />
        </compare>
      </condition>
      <if>
        <condition>
          <getcontrolstate dataitemid="Owner_ResidentialAddress1" />
        </condition>
```

```
        <diset dataitemid="Owner_ResidentialAddress1">

          <diget dataitemid="D_ResidentialAddress1_SSN" />

        </diset>

      </if>

    </if>

  </ScriptCode>

  <Version>2.17.195</Version>

  <RuleName>IT Training: Owner SSN</RuleName>

  <Class>PropertyChange</Class>

  <LastUpdate>6/2/2020 2:37 PM</LastUpdate>

</RuleDef>

<RuleDef>

  <ScriptCode>

    <if>

      <condition>

        <compare op="==">

          <const value="IT Training" />

          <diget dataitemid="FLI_PRODUCT_CARRIER_NAME" />

        </compare>

      </condition>

      <if>

        <condition>

          <getcontrolstate dataitemid="Owner_SSN" />

        </condition>

        <diset dataitemid="Owner_SSN">
```

```xml
          <diget dataitemid="D_Owner_SSN" />

        </diset>

      </if>

    </if>

  </ScriptCode>

  <Version>2.17.195</Version>

  <RuleName>IT Training: Owner Phone</RuleName>

  <Class>PropertyChange</Class>

  <LastUpdate>6/2/2020 2:37 PM</LastUpdate>

</RuleDef>

<RuleDef>

  <ScriptCode>

    <if>

      <condition>

        <compare op="==">

          <const value="IT Training" />

          <diget dataitemid="FLI_PRODUCT_CARRIER_NAME" />

        </compare>

      </condition>

      <if>

        <condition>

          <getcontrolstate dataitemid="Owner_DOB" />

        </condition>

        <diset dataitemid="Owner_DOB">

          <diget dataitemid="D_Owner_DOB" />
```
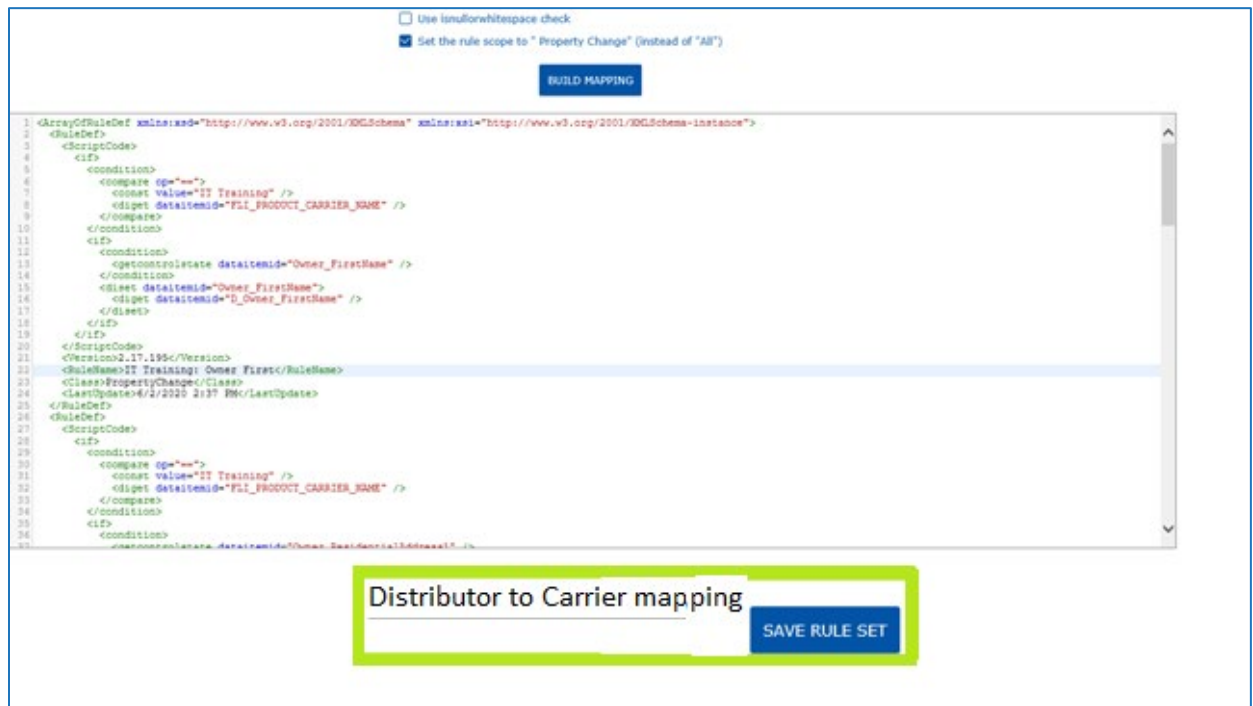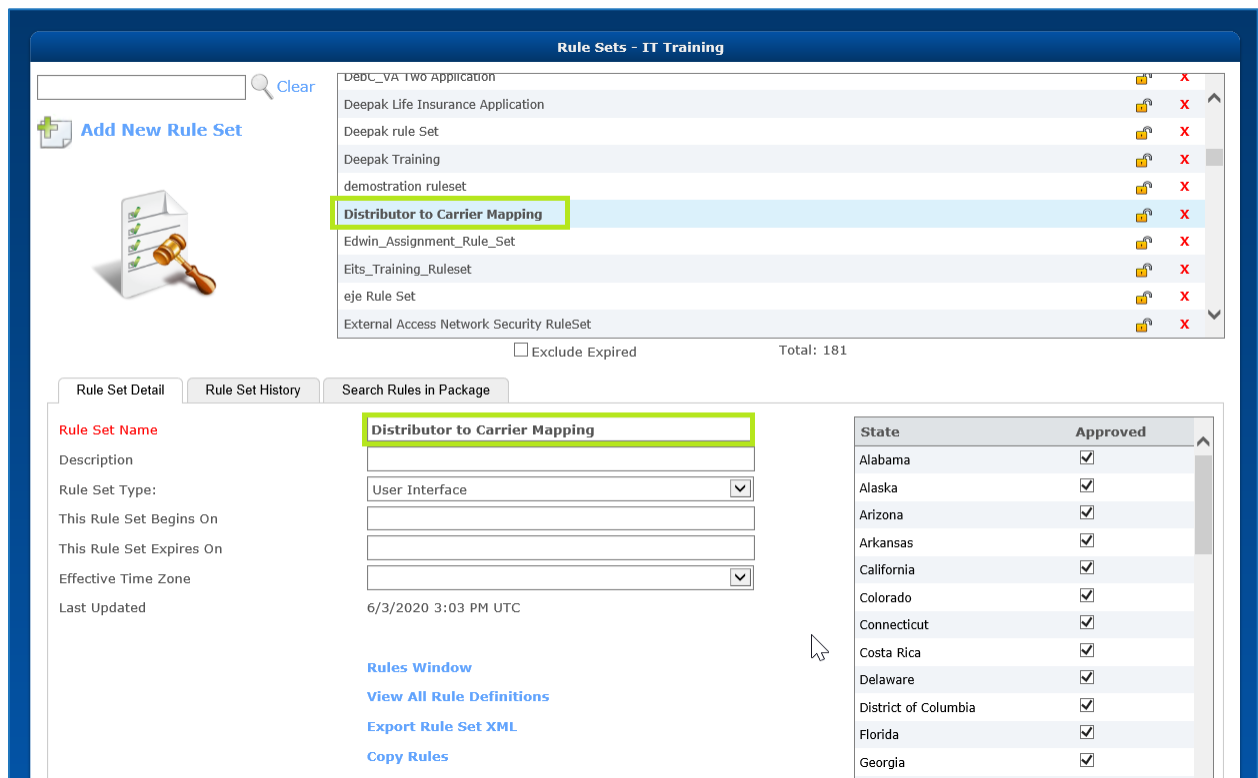
```xml
      </diset>
    </if>
  </if>
</ScriptCode>
<Version>2.17.195</Version>
<RuleName>IT Training: Owner DOB</RuleName>
<Class>PropertyChange</Class>
<LastUpdate>6/2/2020 2:37 PM</LastUpdate>
</RuleDef>
<RuleDef>
<ScriptCode>
  <if>
    <condition>
      <compare op="==">
        <const value="IT Training" />
        <diget dataitemid="FLI_PRODUCT_CARRIER_NAME" />
      </compare>
    </condition>
    <if>
      <condition>
        <getcontrolstate dataitemid="Owner_Annuitant_SSN" />
      </condition>
      <diset dataitemid="Owner_Annuitant_SSN">
        <diget dataitemid="D_Owner_Annuitant_SSN" />
      </diset>
```

```
      </if>
    </if>
   </ScriptCode>
   <Version>2.17.195</Version>
   <RuleName>IT Training: Annuitant SSN</RuleName>
   <Class>PropertyChange</Class>
   <LastUpdate>6/2/2020 2:37 PM</LastUpdate>
 </RuleDef>
 <RuleDef>
  <ScriptCode>
   <if>
    <condition>
     <compare op="==">
      <const value="IT Training" />
      <diget dataitemid="FLI_PRODUCT_CARRIER_NAME" />
     </compare>
    </condition>
    <if>
     <condition>
      <getcontrolstate dataitemid="Owner_Gender" />
     </condition>
     <if>
      <condition>
       <compare op="==">
        <diget dataitemid="D_Owner_Gender" />
```

```xml
          <const value="M|1" />
        </compare>
      </condition>
      <diset dataitemid="Owner_Gender">
        <const value="1" />
      </diset>
      <elseif>
        <compare op="==">
          <diget dataitemid="D_Owner_Gender" />
          <const value="F|2" />
        </compare>
      </elseif>
      <diset dataitemid="Owner_Gender">
        <const value="2" />
      </diset>
      </if>
    </if>
  </if>
  </ScriptCode>
  <Version>2.17.195</Version>
  <RuleName>IT Training: Owner Gender</RuleName>
  <Class>PropertyChange</Class>
  <LastUpdate>6/2/2020 2:37 PM</LastUpdate>
  </RuleDef>
</ArrayOfRuleDef>
```

6.   The user names the ruleset and saves it by clicking **SAVE RULE SET**.

Ruleset as seen in Rule Sets page:

7.   The user selects the ruleset on the left and is able to view all rules broken out by the grouping that was applied.

This ruleset is complete and available to be added to the package that is designated as mapping from Distributor to Carrier.

Flag to Control When to Use the Distributor to Carrier Master Form Mapping

■ In order to eliminate breaking existing distributor mappings we have created a Flag that will be set to indicate when to utilize the Distributor to Carrier Mapping RuleSet.
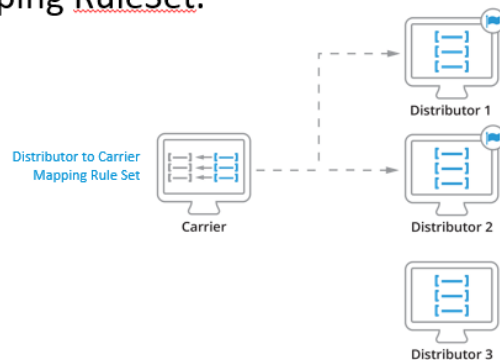
NOTE: USE_CARRIER_MAPPING needs to be added to all mapping rules manually. This data item will act as a trigger for all Distributor to Carrier mapping rules. If the USE_CARRIER_MAPPING is True than the Distributor to Carrier mapping ruleset will enable. If USE_CARRIER_MAPPING is False the Distributor to Carrier mapping will NOT enable.

Following are the steps to implement and test the USE_CARRIER_MAPPING data item.

1. Add a Textbox on the master mapping form with tag "USE_CARRIER_MAPPING" as follows.

2. <mark>Add <diget dataitemid="USE_CARRIER_MAPPING" type="Boolean" /></mark> line to all mapping rules.

Example:

<if>

  <condition>

  <getcontrolstate dataitemid="Annuitant_FirstName" />

    <mark><diget dataitemid="UseCarrierMapping" type="Boolean" /></mark>

```
</condition>

<diset dataitemid="Annuitant_FirstName">

  <diget dataitemid="D_Annuitant_FirstName" />

</diset>

</if>
```

3.   Add the rule below to enable carrier mapping in distributor.

```
<diset dataitemid=" USE_CARRIER_MAPPING ">

        <const value="true" type="String" />

</diset>
```

This rule needs to be added to your Distributor to Carrier ruleset manually.

Example:
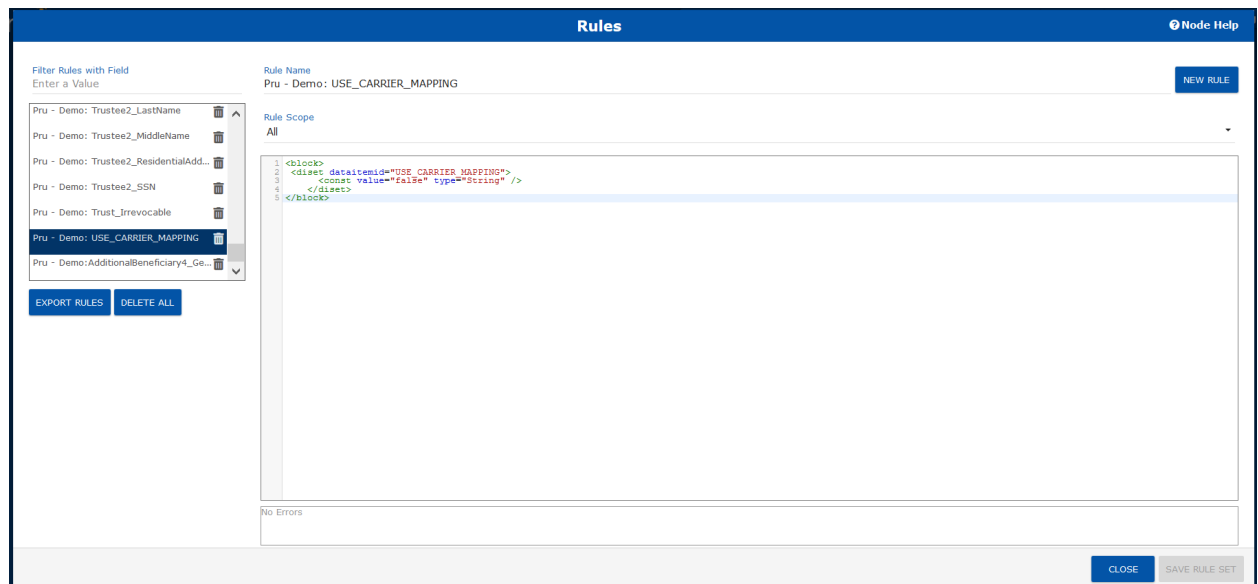
In real life, the rule mentioned in step 3 should add in distributor environment for which we want to run mapping.

When adding the USE_CARRIER_MAPPING flag rule the user will need to be aware of several variations shown below:

•   Removal of the flag rule:

If the flag rule is not present, the Distributor to Carrier mapping ruleset will not enable the ruleset.

- Add False Value to USE_CARRIER_MAPPING flag rule:

```
<diset dataitemid=" USE_CARRIER_MAPPING ">
        <const value="false" type="String" />
</diset>
```

USE_CARRIER_MAPPING rule is set to false and will not correctly map Distributor to Carrier Mapping.

- Add True Value to USE_CARRIER_MAPPING flag rule:

```
<diset dataitemid=" USE_CARRIER_MAPPING ">
        <const value="true" type="String" />
 </diset>
```

USE_CARRIER_MAPPING rule is set to true and will correctly map Distributor to Carrier Mapping.

- Add USE_CARRIER_MAPPING flag rule to distributor:

In a live Distributor to Carrier Mapping, the USE_CARRIER_MAPPING rule will be included within the Distributor ruleset.

8   Rule Set Resides with Carrier

The Rule Set with the mapping will reside in the Carrier package with the name Distributor to Carrier Mapping. That package will be added to the sales agreements and deployed to the distributor. Doing this will eliminate the redundant mapping when bringing up a new distributor.

1.   Once the Distributor to Carrier ruleset is complete, the user adds this ruleset to the desired Forms/Wizard Package.

Example:
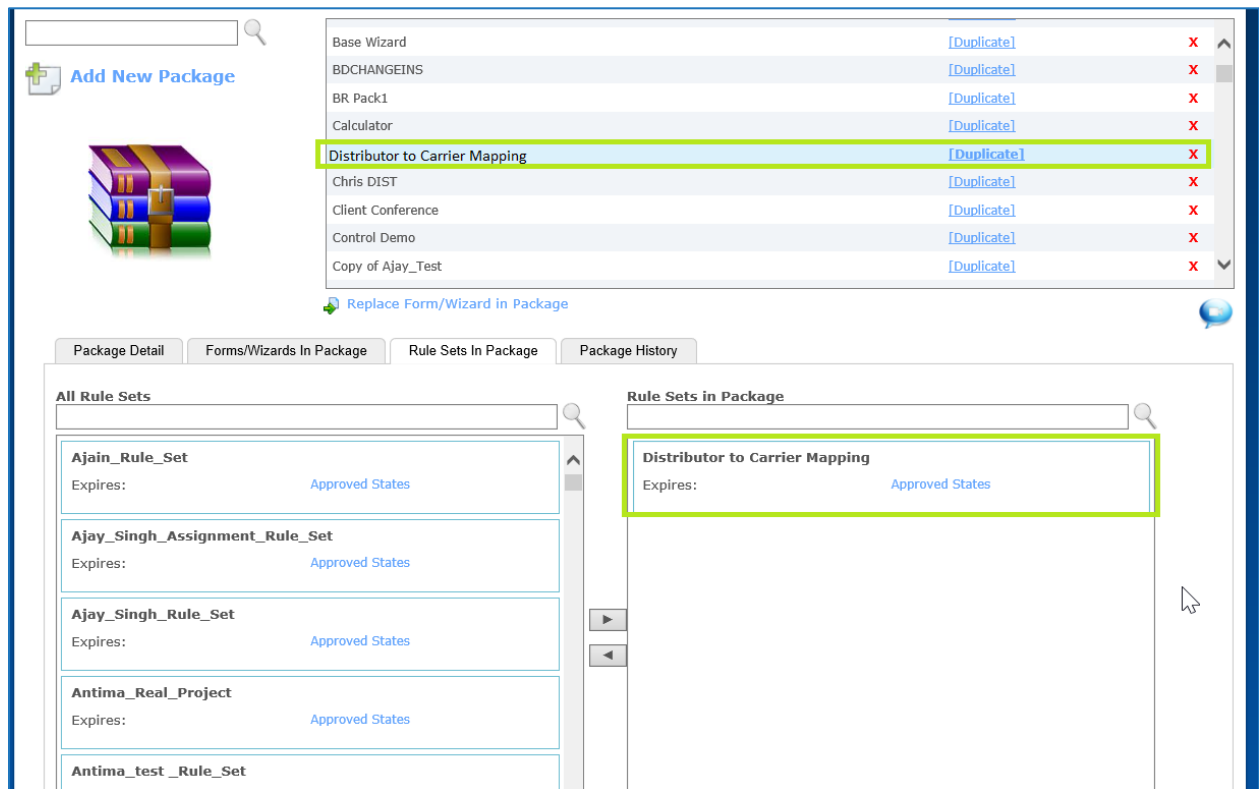
2.   Once the package is complete, the user then adds the completed package to the desired Distributor Sales Agreement as shown below.

NOTE: "Distributor to Carrier Mapping ruleset" will trigger only if "UseCarrierMapping" tag = "True". We will set it "True" in distributor instance only for new distributors.

Example:

Sales Agreement of "IT Training" for product "Chris Common tag Training"

3.   Once the desired package is saved, a deployment needs to occur in order to send the completed package to the Distributor. As with all deployments, the desired synchronization scope must be selected. In this example, Distributor All Configuration is selected in order to deploy all necessary changes. Verify that the correct package is with the proper distributor to deploy to.

4. To verify that the deployment is successful, the user enters the distributor environment and goes to the Sales Agreement page.

5. In the Sales Agreement Page, the user verifies the deployed Carrier package is present and adds the completed package to the desired Distributor package. The Distributor package must be first in the Assigned Packages list as the user is mapping from Distributor to Carrier.

6. Testing in the Distributor Environment can now occur. The user enters the FireLight application and begins to fill out all fields that have been mapped in the Distributor to Carrier Ruleset.

Distributor Form:

Carrier Form:

The information that was entered in the Distributor form is now mapped into the Carrier form.

*Important Note from the test shown above.

When data is entered in the Distributor forms and correctly maps, a user will see the following screens.

Distributor:

Carrier:

This mapping is correct as all data from the distributor mapped to the carrier correctly.

However, in the situation that a user changes the carrier data, will the data change to the distributor?

Carrier change:

The value from first name has changed from "Bob" to "Chris" in the carrier form.

Distributor form updated:

**Index Annuity
Insurance Authorization Form**
Page 1 of 3

Chase Insurance Agency, Inc.
Insurance and Annuities made available through Chase Insurance Agency, Inc. or affiliated agencies.

1. Primary Contract Owner (or annuitant if trust)

Citizenship Status (select one)
☐ USA
☐ Resident Alien (provide Country of Citizenship) [▾]

Name
| Bob | W | Johnson |

Social Security or Tax ID Number | Date of Birth (mm/dd/yyyy) | # Dependents

Gender (select one): ☐ Male ☐ Female
Marital Status (select one): ☐ Single ☐ Married

Primary Identification Type

Identification Number | ID Issue Date (mm/dd/yyyy)

ID State/Country | ID Expiration Date (mm/dd/yyyy)

Legal/Residential Address (no P.O. Box or "In Care Of" address)

City | State | Zip Code

2. Joint Contract Owner

Citizenship Status (select one)
☐ USA
☐ Resident Alien (provide Country of Citizenship) [▾]

Name

Social Security or Tax ID Number | Date of Birth (mm/dd/yyyy) | # Dependents

Gender (select one): ☐ Male ☐ Female
Marital Status (select one): ☐ Single ☐ Married

Primary Identification Type

Identification Number | ID Issue Date (mm/dd/yyyy)

ID State/Country | ID Expiration Date (mm/dd/yyyy)

Legal/Residential Address (no P.O. Box or "In Care Of" address)

City | State | Zip Code

Although the value for Owner first name changed on the carrier form the original value remains on the distributor form; therefore, the answer in "No". Carrier values will not affect distributor values if changed. This was an important note during our testing for Distributor to Carrier.

8.1   Test Cases

To effectively test the Distributor to Carrier Mappings we created an XML test plan spreadsheet. This will assist testers with testing the Distributor to Carrier mappings effectively.

1. The Master RuleSet and master hidden form will live in the carrier packages and will be deployed to distribution.

2. In the distributor, if UseCarrierMapping=true, then the below tests should pass if the distributor has these tags. (this flag will only be set for new distributors using the common tags and common list values. This will ensure that old mapping is not impacted by the new mapping.)

3. UseCarrierMapping=false or null, then the below tests should fail, regardless of the distributor tags.

4. Test the below mapping with the implementation specs for the carrier and distributor to ensure the carrier and distributor rules are still honored.

Group	From Dataitem	From Value	To Dataitem	To Dataitem Value	Pass/Fail	Comment
Owner_NonNaturalName	D_Owner_NonNaturalName		PRIMARY_OWNER_ENTITY_NAME		Pass	
Owner_ResidentialAddress1	D_Owner_ResidentialAddress1		PRIMARY_OWNER_ADDRESS_LINE1		Pass	
Owner_ResidentialAddress_City	D_Owner_ResidentialAddress_City		PRIMARY_OWNER_ADDRESS_CITY		Pass	
Owner_ResidentialAddress_State	D_Owner_ResidentialAddress_State		PRIMARY_OWNER_ADDRESS_STATE		Pass	
Owner_ResidentialAddress_Zipcode	D_Owner_ResidentialAddress_Zipcode		PRIMARY_OWNER_ADDRESS_POSTAL_CODE		Pass	
Owner_Home_PhoneNumber	D_Owner_Primary_PhoneNumber		PRIMARY_OWNER_PHONE_FORMATTED		Pass	
Owner_Personal_EmailAddress	D_Owner_Personal_EmailAddress		PRIMARY_OWNER_EMAIL		Pass	
Owner_DOB	D_Owner_DOB		PRIMARY_OWNER_DATE_FORMATTED		Pass	
Owner_Citizen_NonUS_Country	D_Owner_Citizen_NonUS_Country		PRIMARY_OWNER_RESIDENCY_RESIDENT_TEXT		Pass	
Owner_SSN	D_Owner_SSN		PRIMARY_OWNER_TAX_ID		Pass	
Owner_ResidentialAddress2	D_Owner_ResidentialAddress2		PRIMARY_OWNER_ADDRESS_LINE2		Pass	
AdditionalAgent2_FullName	D_AdditionalAgent2_FullName		AGENT_2_FULL_NAME_APP		Pass	
AdditionalAgent3_FullName	D_AdditionalAgent3_FullName		AGENT_3_FULL_NAME_APP		Pass	
AdditionalAgent4_FullName	D_AdditionalAgent4_FullName		AGENT_4_FULL_NAME_APP		Pass	
AdditionalAgent5_FullName	D_AdditionalAgent5_FullName		AGENT_5_FULL_NAME_APP		Pass	
Agent_FullName	D_Agent_FullName		AGENT_OF_RECORD_FULL_NAME_APP		Pass	
JointOwner_ResidentialAddress_City	D_JointOwner_ResidentialAddress_City		JOINT_OWNER_ADDRESS_CITY		Pass	
JointOwner_ResidentialAddress1	D_JointOwner_ResidentialAddress1		JOINT_OWNER_ADDRESS_LINE1		Pass	
JointOwner_ResidentialAddress_Zipcode	D_JointOwner_ResidentialAddress_Zipcode		JOINT_OWNER_ADDRESS_POSTAL_CODE		Pass	

As shown above, this test plan will show the following:

1. The Master RuleSet and master hidden form live in the carrier packages and are deployed to distribution.

2. In the distributor, if UseCarrierMapping=true, then the below tests should pass if the distributor has these tags. (This flag will only be set for new distributors using the common tags and common list values. This will ensure that old mapping is not impacted by the new mapping.)

3. UseCarrierMapping=false or null, then the below tests should fail, regardless of the distributor tags.

4. Test the below mapping with the implementation specs for the carrier and distributor to ensure the carrier and distributor rules are still honored.

   Group-The name of the group selected by the user to separate mapping rules

   From Dataitem- The Distributor dataitem

   From Value- The Distributor List Value

   To Dataitem- The Carrier dataitem

   To Dataitem value- The Carrier List Value

   Pass/Fail- Did this mapping Pass or Fail the test

   Comment- Any comments that were noted from the tester

8.2 Developer Notes

- A "common hidden mapping form" should be added in all products/packages in all carriers.

- A "common mapping ruleset" should be added in all products/packages in all carriers.

- Mapping from "common hidden mapping form" to "Carriers forms" should be tested for all products/packages in all carriers.

- "Common mapping ruleset" will trigger only if "UseCarrierMapping" tag = "True"—we will set it "True" in distributor instance only for new distributors.

- When products/packages from carrier will deploy to a distributor, "common hidden mapping form" and "common mapping ruleset" will automatically deploy into the distributor.

- Form team will need to do nothing for mapping from Distributor to carrier excect set "UseCarrierMapping = True".

- We have noticed in our testing that the Generate Mapping Ruleset will give incorrect rules in cases that the data item is the same, but the list value is different. For example, D_Owner_Gender is found twice on the download tag list. Once for Male and once for Female. Currently the conversion tool is not separating the values and gives an incorrect rule. This issue will need to be adjusted manually. (NEED TO TEST TO VERIFY THIS IS STILL AN ISSUE.)

Example of an Incorrect Rule:

```
161    </RuleDef>
162    <RuleDef>
163        <ScriptCode>
164            <if>
165                <condition>
166                    <compare op="==">
167                        <const value="IT Training" />
168                        <diget dataitemid="FLI_PRODUCT_CARRIER_NAME" />
169                    </compare>
170                </condition>
171                <if>
172                    <condition>
173                        <getcontrolstate dataitemid="PRIMARY_OWNER_SEX_MALE_X" />
174                    </condition>
175                    <if>
176                        <condition>
177                            <compare op="==">
178                                <diget dataitemid="D_Owner_Gender" />
179                                <const value="1" />
180                            </compare>
181                        </condition>
182                        <diset dataitemid="PRIMARY_OWNER_SEX_MALE_X">
183                            <const value="TRUE" />
184                        </diset>
185                        <elseif>
186                            <compare op="==">
187                                <diget dataitemid="D_Owner_Gender" />
188                                <const value="2" />
189                            </compare>
190                        </elseif>
191                        <diset dataitemid="PRIMARY_OWNER_SEX_MALE_X">
192                            <const value="TRUE" />
193                        </diset>
194                    </if>
195                </if>
196            </if>
197        </ScriptCode>
```

Example of a Good Rule to Use for Owner_Gender:

<block>

<if>

  <condition>

   <compare op="==">

     <diget dataitemid="D_JointOwner_Gender" />

     <const value="1" type="String" />

   </compare>

   <getcontrolstate dataitemid="JOINT_OWNER_SEX_MALE_X" />

   <diget dataitemid="USE_CARRIER_MAPPING" type="Boolean" />

  </condition>

  <diset dataitemid="JOINT_OWNER_SEX_MALE_X">

```
        <const value="true" type="String" />

    </diset>

<elseif>

    <compare op="==">

      <diget dataitemid="D_JointOwner_Gender" />

      <const value="2" type="String" />

    </compare>

    <getcontrolstate dataitemid="JOINT_OWNER_SEX_FEMALE_X" />

  <diget dataitemid="USE_CARRIER_MAPPING" type="Boolean" />

 </elseif>

  <diset dataitemid="JOINT_OWNER_SEX_FEMALE_X">

    <const value="true" type="String" />

  </diset>

  </if>

</block>
```

Mapping from Combo Box to Text Box:

When mapping from a combo box to a text box a custom rule will need to be used in order for the List values to map correctly. Below is an example of a Country mapping rule from combo box to text box:

MapCountry.txt