FireLight Aura Next Disclosure Engine

Integration Guide

AURA NEXT DISCLOSURE ENGINE
Document Version: 2

6/11/2024

The information contained in this document is current as of the date of the publication. Because Insurance Technologies, LLC must respond to changing market conditions and technology advances, Insurance Technologies, LLC cannot guarantee the accuracy of any information presented after the date of publication.

INSURANCE TECHNOLOGIES, LLC MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, IN THIS DOCUMENT AND HEREBY DISCLAIMS ANY AND ALL SUCH WARRANTIES.

The material contained in this document is considered confidential and the intellectual property of Insurance Technologies, LLC. The recipient is given access to this material on the condition that the recipient (1) will keep the information confidential at all times, and (2) will not copy or modify or share the materials, except as expressly authorized by Insurance Technologies, LLC. The recipient should limit its disclosure of the information and materials only to its employees who have a clear business purpose and need to receive such information and materials and who are bound by confidentiality obligations to the recipient that are at least as protective of such information and materials as those contained herein.

Hexure
Two North Cascade Avenue, Suite 200
Colorado Springs, CO 80903
Phone: 719.442.6400
Internet E-Mail: info@hexure.com
Website: hexure.com

Table of Contents

Aura Next Integration Setup

Project Overview

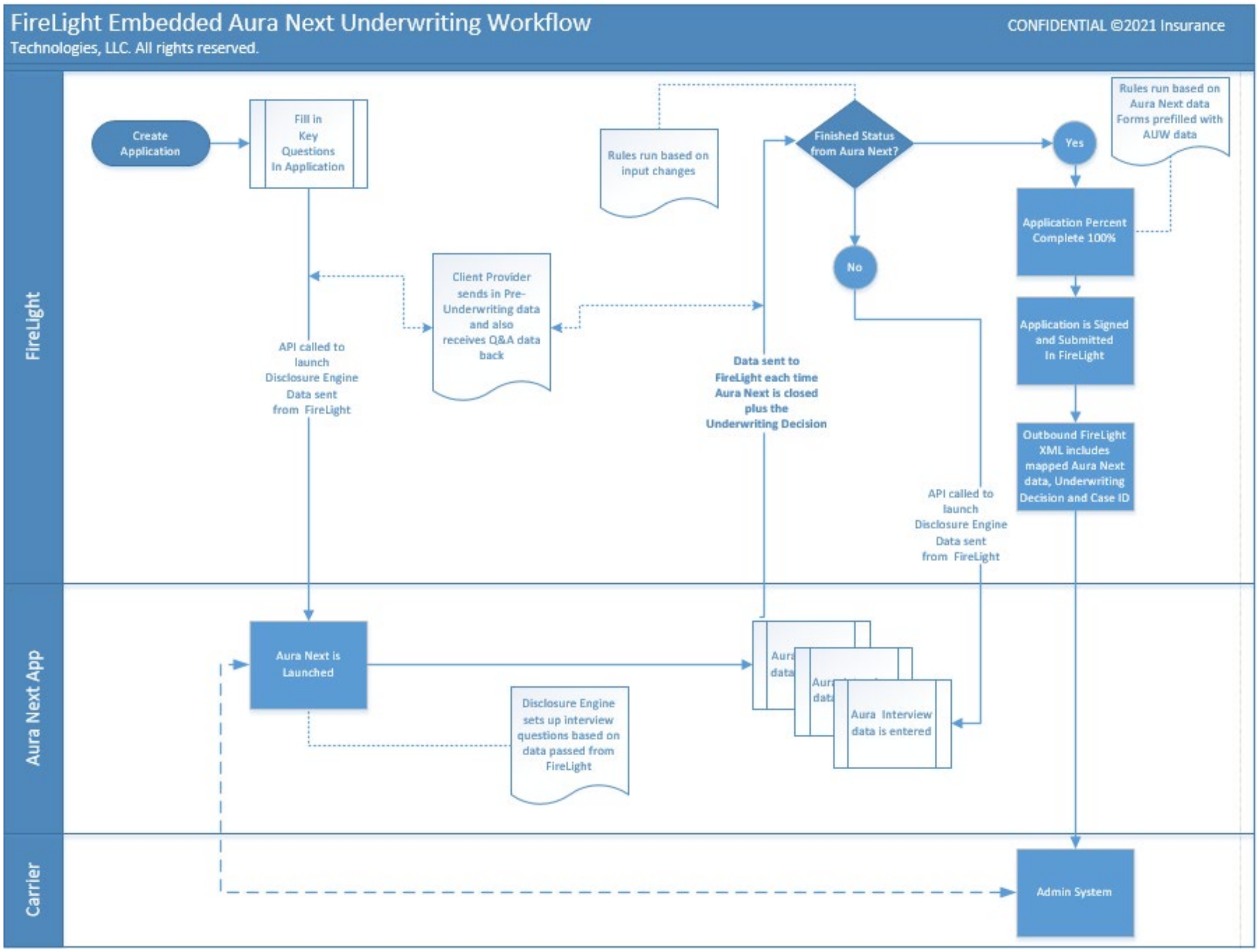
FireLight® has been enhanced to leverage RGA's Aura Next APIs and Disclosure Engine within a FireLight activity to gather underwriting data for life applications.

The APIs are embedded in FireLight and shown in a dialog that opens by clicking on a Custom Action button located in a wizard activity. Data is passed from the Aura Next interview and can be mapped to data items in the activity. FireLight will also retrieve and store the case-level underwriting decision when the interview is complete.

This document contains details regarding the requirements that need to be gathered to integrate Aura Next with FireLight as well as how to set it up through the Admin Tool and the provider.

Note that the Aura Next setup is handled by the carrier, not the distributor. When a distributor runs a carrier's product, the carrier custom action button and provider are called after the relevant distributor components.

FireLight Aura Next Workflow Diagram


FireLight Embedded Aura Next Underwriting Workflow
CONFIDENTIAL ©2021 Insurance
Technologies, LLC. All rights reserved.
FireLight
Aura Next App
Carrier
Create Application
Fill in Key Questions In Application
Rules run based on input changes
Finished Status from Aura Next?
Yes
No
Rules run based on Aura Next data Forms prefilled with AUW data
Application Percent Complete 100%
Client Provider sends in Pre-Underwriting data and also receives Q&A data back
Application is Signed and Submitted In FireLight
API called to launch Disclosure Engine Data sent from FireLight
Data sent to FireLight each time Aura Next is closed plus the Underwriting Decision
Outbound FireLight XML includes mapped Aura Next data, Underwriting Decision and Case ID
API called to launch Disclosure Engine Data sent from FireLight
Aura Next is Launched
Aura data
Aura data
Aura Interview data is entered
Disclosure Engine sets up interview questions based on data passed from FireLight
Admin System

FireLight Aura Next Workflow Steps

The Aura Next integration workflow steps are as follows:

1. Create a new activity.

   Note: All Aura Next data in a copied activity is cleared when it is copied.

2. The user fills out the necessary pre-underwriting questions in the FireLight wizard that Aura Next uses to determine which questions to display in the interview.
3. The user navigates to the Aura Next custom action button in the wizard.
4. The user clicks the custom action button.
5. A new Aura Next interview is created using the data retrieved from the carrier’s provider.
6. The Aura Next dialog opens in FireLight and the interview is rendered.
7. Data is entered in the interview questions.
8. At any point within the interview, the dialog can be closed.

9. Upon closing the Aura Next dialog, the following actions occur:
    a. The Aura Next interview is saved.
    b. The current case status of the interview is retrieved from Aura Next and stored in a data item ({button_dataitemid}_AURA_STATUS). Valid statuses are:
        i. “” – empty string, interview has not been started.
        ii. “INTERVIEW_IN_PROGRESS” – interview is in progress but not yet complete.
        iii. “INTERVIEW_FINISHED” – interview is finished.
    c. The current submission status of the interview is retrieved from Aura Next and stored in a data item ({button_dataitemid}_AURA_SUBMISSION_STATUS). Valid statuses are:
        i. “” – empty string, interview has not been started.
        ii. “INTERVIEW_NOT_SUBMITTED” – interview has not been submitted in Aura Next.
        iii. “INTERVIEW_SUBMITTED” – interview is finished and has been submitted in Aura Next.
    d. The list of answered questions is retrieved from Aura Next. The carrier’s provider may parse/retrieve individual questions/answers and store them in data items.
        i. The interview data is stored so it can be passed in the outbound XML upon application submission.
    e. The list of answered questions is available to the provider so new data items can be added with combinations of questions and answers.
    f. A case-level underwriting decision is retrieved from Aura Next.
        i. The full underwriting decision XML is available to the provider at the same point that question/answer data can be parsed.
        ii. At submission, the underwriting decision XML structure can be passed as part of the outbound XML.
        iii. FireLight stores the overall case underwriting decision in a data item ({button_dataitemid}_AURA_CASE_DECISION). All other underwriting decisions can be retrieved by the provider.

10. At any time while the activity is unlocked, the Aura Next dialog can be re-opened to access and change the data in the interview.
    a. Each time the Aura Next case is opened, it will automatically be updated with the latest pre-underwriting data generated by the provider. If applicable, new interview questions will appear and some may be removed. For example, a user changes the insured age from 59 to 60, which triggers different questions in the interview.

Required Aura Next Data for Integration

To make it possible for FireLight to access the Aura Next APIs and Disclosure Engine, manage the Aura Next case, and map data between FireLight and Aura Next, the following requirements are required:

1. Aura Next Project ID. This is created and managed by the client in the Aura Next portal. It is added in the Wizard Designer to the custom action button properties used to launch Aura Next. This should be the ID of the project containing the interview used in FireLight.
2. Aura Next Interview ID. This is created and managed by the client in the Aura Next portal. It is added in the Wizard Designer to the custom action button properties used to launch Aura Next. This should be the ID of the specific Aura Next interview you'd like to show in FireLight.
3. Aura Next Locale. This is managed by the client in the Aura Next portal. It is added in the Wizard Designer to the custom action button properties used to launch Aura Next. See the Custom Action Button section for a full list of the locales.
4. Provider-created outbound mapping. The provider will be responsible for creating any pre-underwriting data required by the Aura Next interview.
5. The Aura Next API version must be specified. The current version is v1, but this may change in the future as changes are made to the Aura Next API.
6. The Aura Next carrier code. This is dependent on the client and RGA's setup.
    a. ***A carrier may have multiple carrier code options depending on their needs. If this is the case, when entering carrier code in the Custom Action button and in the Third Party Configuration Table, the "Config Group" will need to be used to designate which carrier code is to be used in the API call.***
7. The Aura Next environment. This is dependent on the client and RGA's setup.
8. The Aura Next password is used in combination with the Aura Next username to obtain access to the Aura Next API and to authenticate the Disclosures Engine.

9. Primary Server Base URL of the Aura Next environment. Aura Next has a standard URL schema that is filled in. Everything up to the API version should be included in this URL. For example, https://demo-prod-disclosures.auraservices.cloud.
10. The Aura Next username is used in combination with the Aura Next password to obtain access to the Aura Next API and to authenticate the Disclosures Engine.
11. A list of all data that Aura Next needs for setting up the interview questions. This is obtained from RGA and the client will use it to set up the provider pre-underwriting data.

RGA Aura Next Environment Setup

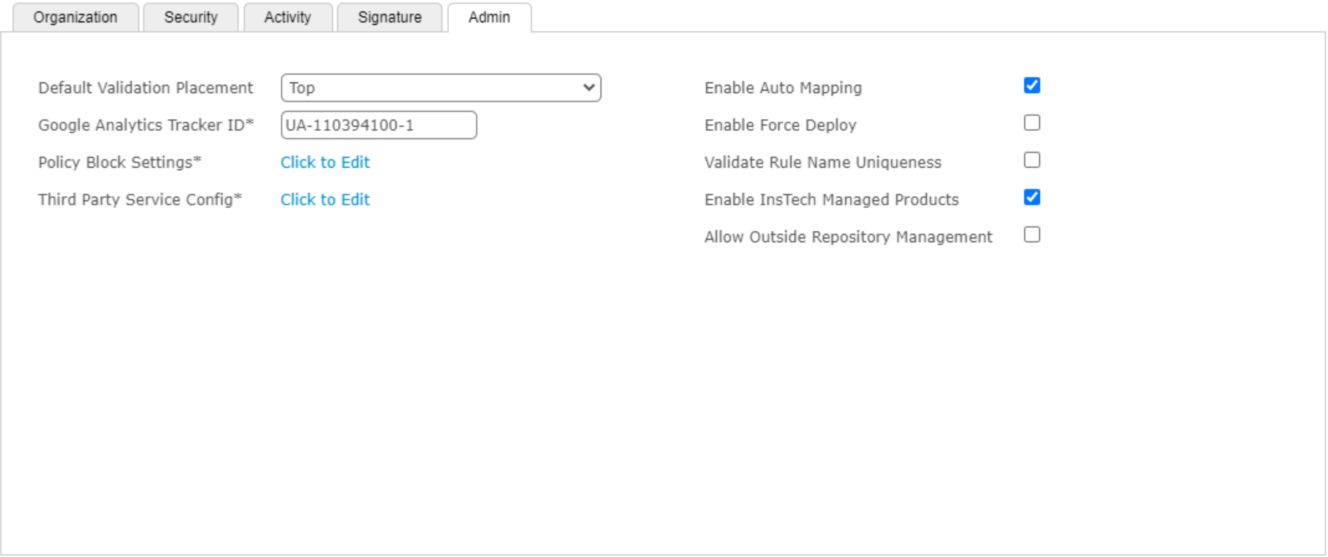
Aura Next uses a whitelist process to determine what environments are allowed to use the JavaScript Disclosure Engine. To use Aura Next in an embedded FireLight environment, RGA must whitelist the hosting parent domain. For example, if you are hosting FireLight embedded under the URL https://isured.com, isured.com must be whitelisted by RGA. Non-embedded FireLight is already whitelisted, so this step is only necessary if you are embedding FireLight in a separate web application.

Setting Up Aura Next Disclosure Engine Integration in Admin Tool

There are several steps to setting up the Aura Next integration in FireLight.

Organization Settings

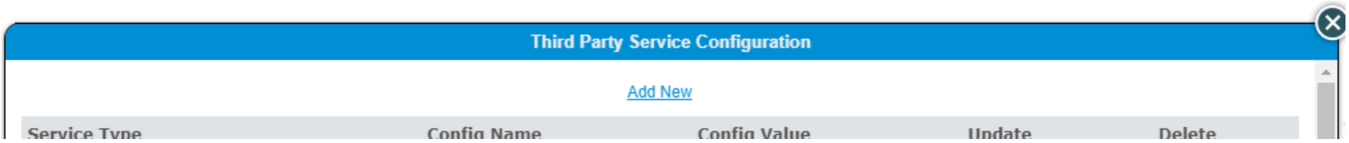
On the Admin tab, click the Third Party Service Config Click to Edit link.



Default Validation Placement	Top	Enable Auto Mapping	☑
Google Analytics Tracker ID*	UA-110394100-1	Enable Force Deploy	☐
Policy Block Settings*	Click to Edit	Validate Rule Name Uniqueness	☐
Third Party Service Config*	Click to Edit	Enable InsTech Managed Products	☑
		Allow Outside Repository Management	☐

Third-Party Configuration

Click the Add New link at the top of the page. Enter the following configuration values.



ApiVersion

The version of the Aura Next Disclosure Engine API to use. As of June 2021, the current version is v1, but this may change in the future. Confirm this information with your RGA Aura Next representative.

CarrierCode

The Aura Next carrier code. This will vary per client. Contact your RGA Aura Next representative for the correct carrier code name***. If a carrier has multiple carrier codes that could be used, an entry for each carrier code must be made and a the "Config Group" field will need to be utilized in the Third Party Configuration Table. The designation entered in the Config Group field in the Third Party Configuration Table must also be utilized in the Config Group field in the Custom Action button properties***.

Environment

The Aura Next environment. This will vary per client and per FireLight environment. Aura Next has multiple environments available. If desired, you may use an Aura Next test environment for FireLight's test environments and an Aura Next production environment for FireLight's production environment. Contact your RGA Aura Next representative for the correct environment name.

Password

The Aura Next OAuth2 password is used for both API access and the JavaScript Disclosure Engine authentication. Contact your RGA Aura Next representative for the correct password.

PrimaryServer

This is the base URL of the Aura Next cloud environment hosted by RGA. Aura Next has a standard URL schema that is filled in. Everything up through the version of the Aura Next is used. This should be everything between and including https:// *and* .cloud. For example, https://demo-prod-disclosures.auraservices.cloud. Everything after that point is automatically filled in by FireLight.

Username

The Aura Next OAuth2 username is used for both API access and JavaScript Disclosure Engine authentication. Contact your RGA Aura Next representative for the correct username.

Provider Setup

To use the Aura Next underwriting disclosure engine service, a provider extension must be defined for the organization that builds the pre-underwriting required data, which varies based on the interview setup in Aura Next. Once the list of required data is defined, the carrier will work with the FireLight integration team to set up the provider to build the appropriate data from the FireLight activity.

Wizard Designer

Create Wizard

Set up the product inputs in the Wizard Designer. The wizard will contain the pre-underwriting questions based on the client's business requirements, such as jurisdiction, age, and occupation, which will determine the interview questions presented in the Aura Next Disclosure Engine.

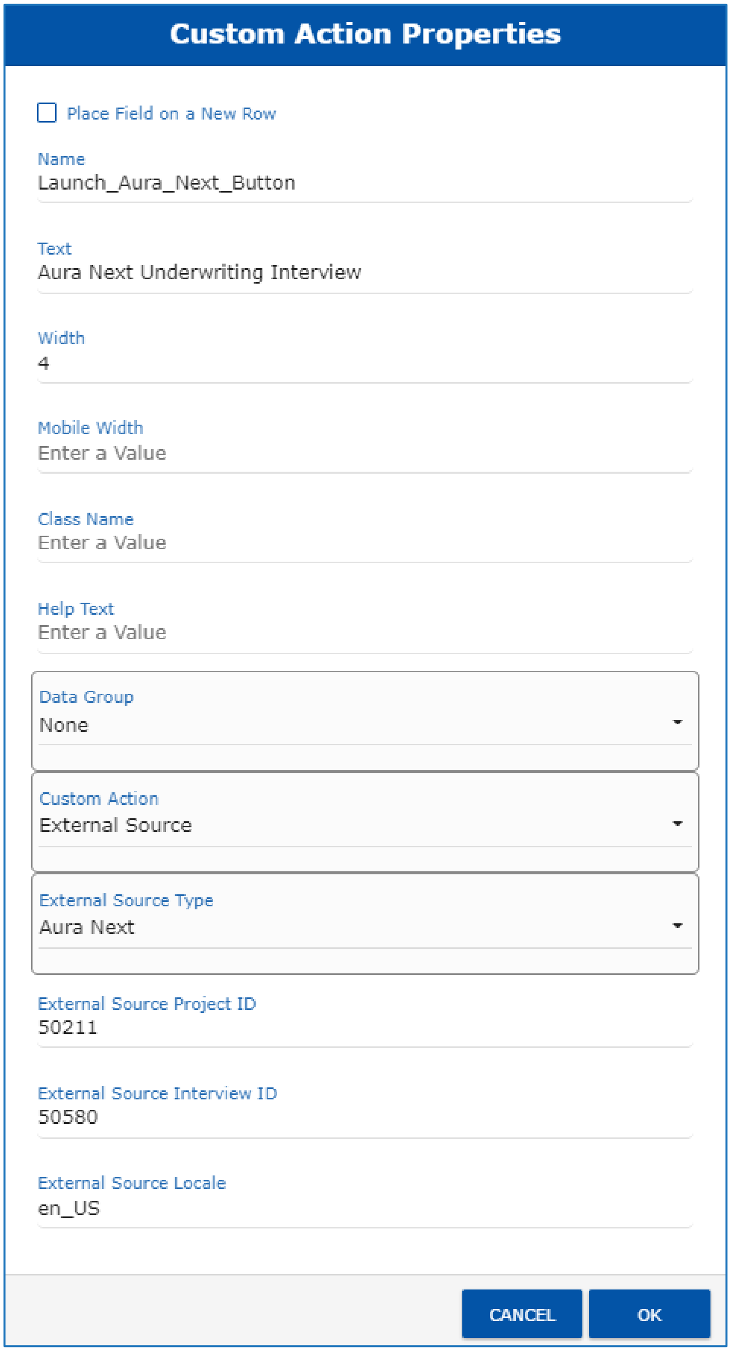
Refer to the Wizard Designer section in the help for details.

Note: The Aura Next Disclosure Engine integration is not supported in legacy forms.

Custom Action Button

Create a Custom Action button on the desired wizard page to launch the Aura Next Disclosure Engine. If a carrier has multiple Carrier Code options that could be used in a call, a specific Custom Action Button will need to be configured for each Carrier Code that could be called.

Optionally add rules to control when the custom action button is available.



Custom Action Properties

□ Place Field on a New Row

Name
Launch_Aura_Next_Button

Text
Aura Next Underwriting Interview

Width
4

Mobile Width
Enter a Value

Class Name
Enter a Value

Help Text
Enter a Value

Data Group
None

Custom Action
External Source

External Source Type
Aura Next

External Source Project ID
50211

External Source Interview ID
50580

External Source Locale
en_US

Refer to the Custom Action section in the help for details about adding the custom action button.

- Set the Custom Action to External Source.
- The External Source Type will appear. Set this to ***Aura Next Disclosure Engine***.
- Enter the External Source Project ID of the Aura Next interview. This can be retrieved from your Aura Next configuration portal.
- Enter the External Source Interview ID of the interview. This can be retrieved from your Aura Next configuration portal.
- Enter the External Source Locale of the interview. This can be retrieved from your Aura Next configuration portal. This should match one of the available locales for the Aura Next interview shown in FireLight. If the locale is not defined in the custom action button, it defaults to en_US. If en_US is not available for the Aura Next interview, an error will be displayed and the interview cannot be launched.
- ***If the carrier has multiple Carrier Code options in the Third Party Configuration table, the Config Group value in the Custom Action button will need to match the Config Group value listed for the desired Carrier Code in the Third Party Configuration table.***

  Valid locale options are English (US): en_US, Czech: cs_CZ, French (Canada): fr_CA, French (France): fr_FR, German (Germany): de_DE, Italian (Italy): it_IT, Japanese: ja_JP, Portuguese (Portugal): pt_PT, Slovak: sk_SK, Spanish (Mexico): es_MX, Spanish (Spain): es_ES

Rules

Set up the rules that apply to the wizard, which are associated with the Aura Next integration. Common rule examples and Aura Next data items are outlined in this section.

All rule examples in this section assume you are using a button with a data item ID of “AURANEXT_APPLICANT.”

Leverage Aura Next Status to Control Activity Percentage Complete

Set up a rule to check the status of the Aura Next interview. If it is not complete, do not allow the FireLight activity to get to 100% complete. An error message will appear on the Aura Next custom action button in this scenario.

```
<if>
  <condition>
   <compare op="==">
    <diget dataitemid="AURANEXT_APPLICANT_AURA_STATUS" />
    <const value="INTERVIEW_FINISHED" type="String" />
   </compare>
  </condition>
  <removemessage dataitemid="AURANEXT_APPLICANT" />
  <else />
  <postmessage dataitemid="AURANEXT_APPLICANT">
   <const value="Aura Next underwriting must be completed before continuing." />
  </postmessage>
</if>
```

Control User Changes in Activity with Finished Interview

Carriers will want to be aware of when a user changes any wizard pre-underwriting questions on a finished interview because it may affect the Aura Next interview questions. This example assumes you want to prohibit changes to pre-underwriting questions after the Aura Next case is complete. To support this scenario, follow these steps:

1. Add a rule to require that the Aura Next interview is finished before continuing (see 5.3.1).
2. Set up a rule to disable all the pre-underwriting questions if the Aura Next status is finished (“INTERVIEW_FINISHED”). Only a subset of the possible fields is shown in the example below.

   ```
   <block>
    <set name="isReadonly">
     <compare op="==">
      <diget dataitemid="AURANEXT_APPLICANT_AURA_STATUS" />
       <const value="INTERVIEW_FINISHED" />
      </compare>
    </set>
    <setreadonly dataitemid="Insured_DOB">
         <get name="isReadonly" />
    </setreadonly>
    <setreadonly dataitemid="Insured_Gender">
         <get name="isReadonly" />
    </setreadonly>
    <setreadonly dataitemid="FinInfo_Income_AnnualIncome">
   ```

```
          <get name="isReadonly" />
  </setreadonly>
  <setreadonly dataitemid="Insured_EmpDetails_Occupation">
          <get name="isReadonly" />
  </setreadonly>
</block>
```

3. Add an increment button to the appropriate wizard page. The below example assumes a data item ID of “RESET_PRE_UNDERWRITING.”
    a. Show the button only when the interview is finished. Prior to the interview being finished, the pre-underwriting questions may still be changed.

       ```
       <setvisibility dataitemid="RESET_PRE_UNDERWRITING">
         <compare op="==">
           <diget dataitemid="AURANEXT_APPLICANT_AURA_STATUS"></diget>
           <const value="INTERVIEW_FINISHED" type="String" />
         </compare>
       </setvisibility>
       ```

    b. When the button is clicked, change the interview status back to “INTERVIEW_IN_PROGRESS” and reset the button back to an unclicked status with a click count of 0. This will guarantee that the Aura Next interview will be opened again prior to continuing through the activity to signatures. When this rule is run, the pre-underwriting fields will automatically unlock when the above rule runs (see #2 in 5.3.2).

       ```
       <if>
         <condition>
           <compare op="&gt;">
             <diget dataitemid="RESET_PRE_UNDERWRITING"></diget>
             <const value="0" type="Int32" />
           </compare>
         </condition>
         <diset dataitemid="AURANEXT_APPLICANT_AURA_STATUS">
           <const value="INTERVIEW_IN_PROGRESS" type="String" />
         </diset>
         <diset dataitemid="RESET_PRE_UNDERWRITING">
           <const value="0" type="Int32" />
         </diset>
         <else />
       ```

```
</if>
```

Aura Next Data Items to Be Used in Rules

The following is a list of Aura Next-specific data items that can be used in rules and the provider. Note that the custom action button ID needs to be prepended to all data items. For example, ({button_dataitemid}_AURA_CASE_ID).

- AURA_INTERVIEW_ID: Contains the ID of the Aura Next interview. Populated after the case is created by clicking the custom action button for the first time in an activity.
- AURA_PROJECT_ID: The project ID that the Aura Next interview falls within. Populated after the case is created by clicking the custom action button for the first time in an activity.
- AURA_LOCALE: Contains the locale of the Aura Next interview. Populated after the case is created by clicking the custom action button for the first time in an activity.
- AURA_CASE_ID: Contains the Aura Next case ID of the interview associated with the custom action button that is clicked. Populated after the case is created by clicking the custom action button for the first time in an activity.
- AURA_STATUS: Contains the current Aura Next interview status attached to the activity in FireLight. Valid responses are blank, INTERVIEW_IN_PROGRESS, and INTERVIEW_FINISHED. The status is updated every time the Aura Next dialog is closed.
- AURA_SUBMISSION_STATUS: Contains the Aura Next interview submitted status. Submission happens automatically when the dialog is closed and all required questions have been answered. Valid responses are blank, INTERVIEW_NOT_SUBMITTED, and INTERVIEW_SUBMITTED. The status is updated every time the Aura Next dialog is closed.
- AURA_CASE_DECISION: Contains the overall underwriting case-level decision based on the interview data.

Provider Extensions

The following provider hooks are available to extend the FireLight base implementation of Aura Next.

Send Outbound Case Data

This provider extension sends data from FireLight to Aura Next and is ***required***. It is used to create the pre-underwriting data that Aura Next needs to create and display an interview. This data typically includes inputs from a FireLight wizard, such as date of birth, gender, salary, etc. The format of this data varies based on the interview setup in your Aura Next portal. Aura Next uses an extended Acord XML format for pre-underwriting data.

Work with your RGA Aura Next contact and the FireLight integration team to set up the necessary data to integrate Aura Next in FireLight.

Receive Aura Case Data

This provider extension receives all case data when the Aura Next interview is closed/saved. The payload that the provider receives includes:

- Questions and answers from the interview.
- Underwriting Decisions, including the case-level decision and all sub-level decisions.
- Status information.

The provider is able to retrieve any of the data from the payload and insert it into any FireLight data items. All questions can be retrieved from the list of questions by their text values or by the index of page number and question number. More complex data parsing and formatting can occur if you need to build a review page to show a list of all answered questions.

Work with your FireLight integration team to retrieve and format the necessary data from Aura Next in FireLight.