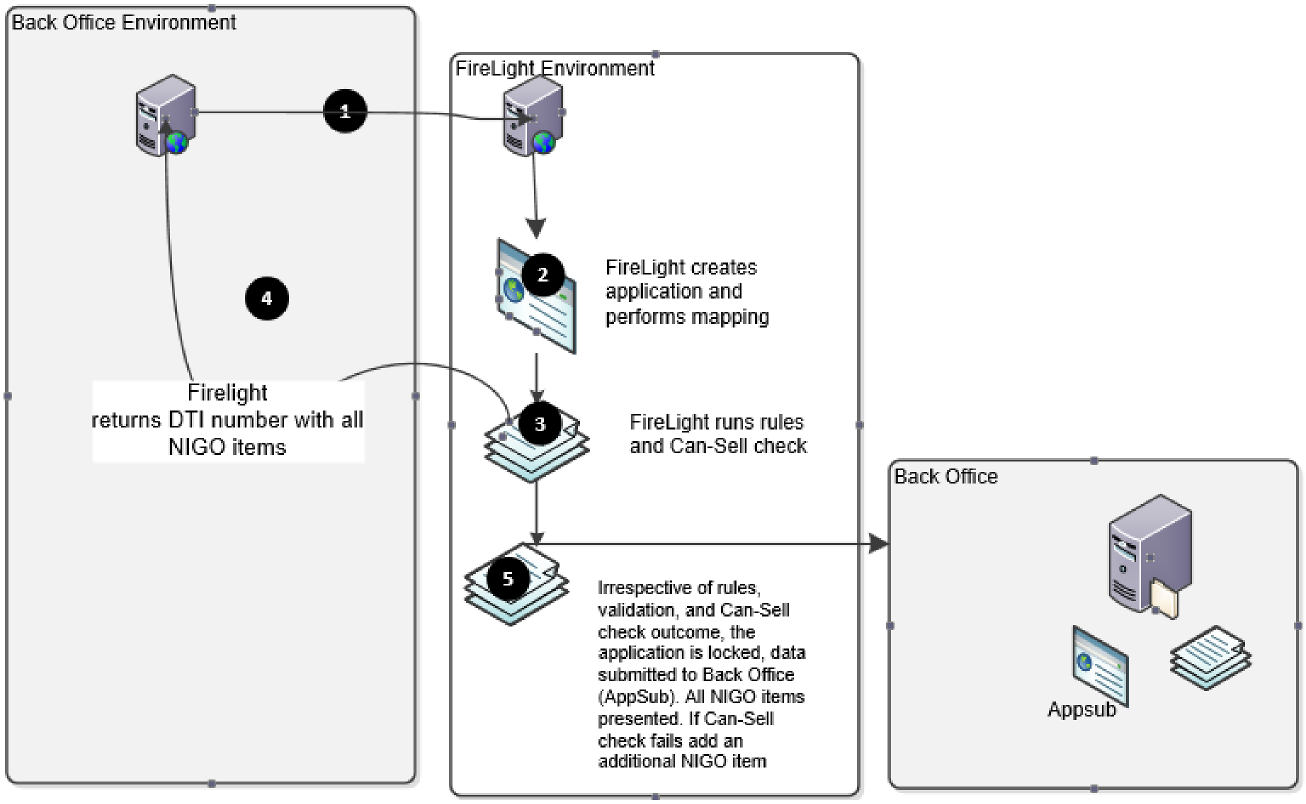
iConnect #153555
Automated Business Service Design Approach

Project Overview

Facilitate base enhancements that provide back office capability to OCR existing paper applications directly into FireLight and return any NIGO responses, and finish normal completion processing of the application. The functionality described below are dependencies of this project. These items are either existing functionality or will be added to Firelight as part of the 2.10 release.

Work Flow


Back Office Environment
FireLight Environment
1
2
FireLight creates application and performs mapping
3
FireLight runs rules and Can-Sell check
4
Firelight returns DTI number with all NIGO items
5
Irrespective of rules, validation, and Can-Sell check outcome, the application is locked, data submitted to Back Office (AppSub). All NIGO items presented. If Can-Sell check fails add an additional NIGO item
Back Office
Appsub

Notes:

Client makes a SOAP request to Firelight with application data in Acord 103 format (with KeyedValue items). FireLight maps (or attempts to map) the data of the 103 to a new application.

Validation of inbound data would occur on Ins Tech side. As part of the SOAP response, FireLight will indicate if the 103 is well formed and passes basic data check, or if there are errors. If the request is not a well-formed payload or if it doesn't pass data check, the system will throw an exception or a relevant validation error (example CUSIP not valid).

Once the data passes the basic checks, FireLight rules will run and look for NIGO's (apply NIGO rules to the data and map data to forms). The client Can-Sell check will also run during this process. IT will pass back all NIGO's including Can-Sell check results to client in SOAP response. The FireLight Provider modules will also add custom data to be sent back (if needed) in the response, so long as it still fits within the Acord 103 structure.

The resulting NIGO messages will be sent back in the response. Then the end processing result specific to the client (regardless of success or not) will be sent.

Features/Requirements

A. Create orders using CUSIP and Solicitation State passed into Firelight via web service as an ACORD103 message

B. Consume Key/Value pair collection passed into Firelight via web service as the Key/Value aggregates of the ACORD103 message

```
<Holding>
  <Policy>
    <ProductCode>XXX DEFINED PRODUCT CODE</ProductCode>
    <CarrierCode>XXX DEFINED CARRIER CODE</CarrierCode>
    <Jurisdiction tc="7">OLI_USA_CO</Jurisdiction>
    <KeyedValue>
      <KeyName>Owner_FirstName</KeyName>
      <KeyValue>Jane</KeyValue>
    </KeyedValue>
    <KeyedValue>
      <KeyName>Owner_MiddleName</KeyName>
      <KeyValue>Doe</KeyValue>
    </KeyedValue>
    <KeyedValue>
      <KeyName>Owner_LastName</KeyName>
      <KeyValue>Smith</KeyValue>
    </KeyedValue>
  </Policy>
</Holding>
```

C. Automatically map data (values, form numbers, form dates, etc.) passed in the Key/Value collection to corresponding tags/controls on forms in Firelight.

D. Orders will be created so they can be viewed in the Firelight UI and available to client employees with appropriate permissions; agent/producer should not be able to see these orders in the Firelight UI

E. Firelight will generate appropriate forms and run established rules (see Supplier Deliverable) against the data mapped from the Key/Value collection. Firelight will generate appropriate forms and run the rules in the background.

F. Firelight will trigger/execute the CanSell check automatically using Agent ID data and handle any errors with incorrect/invalid Agent ID data.

- G. Orders created using this automated process will bypass the Signatures (step 2) and Review (step 3) steps of FireLight and automatically be submitted.
- H. The transmission for the FireLight process will utilize Basic Authentication over HTTPS.
- I. The FireLight web service will respond with a list of errors in three categories: errors related to the application data, errors related to the form composition, and errors related to any CanSell messages.
    - a. Errors related to the application data will be fully determined by the rules written for any of the forms contained within the application. The error messages and corresponding data item ID's will be sent back in the response as NIGO messages.
    - b. Errors related to the CanSell check will be determined by the FireLight Provider modules, which rely on established 3rd party services to serve CanSell messages. CanSell messages will be sent back in the response as NIGO messages.
    - c. Errors related to form composition will be determined through a comparison of the list of forms sent in on the request, and the list of forms generated through the FireLight tool, whether the forms were required (as defined in the FireLight admin tool), or whether the forms were conditionally triggered through the processing of the rules. The errors related to form composition will contain a code to correspond to them, as well as possible additional metadata related to them. The possible errors will be:
        - i. Form Missing in Request XML
            - 1. Code = FormMissing
            - 2. Error will contain reference to the FormNumber & Edition of the form that is missing/expected.
        - ii. Pages Missing from Form as indicated in Request XML
            - 1. Code = PageMissing
            - 2. Error will contain reference to the FormNumber & Edition of the form that is missing a page, and will indicate which page is missing (as was indicated in the Request XML).
        - iii. Form sent in Request XML, but is not present in Application
            - 1. Code = FormUnknown
            - 2. Error will contain reference to the FormNumber & Edition of the form that was sent (but not expected).
        - iv. Form Edition in Request XML does not match Form Edition in FireLight
            - 1. Code = EditionUnknown
            - 2. Error will contain reference to the FormNumber & Edition of the form that has the incorrect Edition, as well as the expected edition.
            - 3. If a form does not have an Edition in FireLight, and additionally was not given one within the Request XML, it will not count as an error.
- J. Firelight will generate and transmit the end processing result for the client including NIGO errors. As a result of possibly incomplete data/NIGO scenarios, FireLight will possibly be sending over an outbound file with incomplete data. Note that FireLight has the option to not send any end processing files.

"K" has been removed as the result of the base team meeting 3/13/17. NOTE: K is still present in the original Work Order for this project.

Use Cases / Workflow Changes

Case A – Request

The XML Request below is an example of the Request structure required for the FireLight web service. Note that the security within the SOAP Header has been taken out of the example, but would otherwise be required.

Notes regarding this example:

- The <TransRefGUID> is a required GUID, and must be unique.
- The Product's CUSIP as defined within the FireLight admin tool must be listed in TXLife/TXLifeRequest/OLifE/Holding/Policy/ProductCode.
- The given carrier's "Carrier Code" as defined by FireLight must be listed in TXLife/TXLifeRequest/OLifE/Holding/Policy/CarrierCode.
- Data items to be contained within the application will be transimitted through the 103 request under TXLife/TXLifeRequest/OLifE/Holding/Policy/KeyedValue.
    - <KeyedValue> is a list, with each element containin a <KeyName>, and a <KeyValue>. The name will be the data item ID, and the value will be the data item's value.
- The <Holding> requires an 'id' attribute with which to identify it.
- The 103 XML requires a <Party> located at TXLife/TXLifeRequest/OLifE/Party with the following requirements:
    - An 'id' attribute with which to identify the party object.
    - Any name can be listed in Party/FullName.
    - The unique ID of the user that serves the given application must be located in Party/Producer/CarrierAppointment/CompanyProducerID.
    - The given carrier's "Carrier Code" as defined by FireLight must be listed in Party/Producer/CarrierAppointment/CarrierCode.
- The 103 XML requires a <Relation> between the Party and Holding.
    - The <Relation> must have its own 'id' attribute, must have an 'OriginatingObjectID' attribute that matches the <Holding> id, and must have a 'RelatedObjectID' attribute that matches the <Party> id.
    - There must be an <OriginatingObjectType> with a tc attribute set to '4'.
    - There must be a <RelatedObjectType> with a tc attribute set to '6'.
    - There must be a <RelationRoleCode> with a tc attribute set to '37'.
- There are a list of <FormInstance> objects – each notes a form from the external calling system. Each <FormInstance> contains:
    - <FormName>, which notates the name of the form. This is optional, and only useful for readability of the xml.
    - <ProviderFormNumber>, which notates the Form Number of the form, as defined within the FireLight admin tool.
    - <FormVersion>, which notates the Form Edition of the form, as defined within the FireLight admin tool.
    - An optional <OLifEExtension>, which contains a list of <MissingPage>. This notates a list of missing pages of the form from the calling external system.
- Within the <FormInstance> list, "Form C Name" (XXXC FormNumber) is listed 3 times. This is to notate that there were three copies of the same form received for the given application. The first form would

have the normal Form Number (as defined within the FireLight admin tool)("XYZ" for example's sake), the subsequent form would have a Form Number of "XYZ2" (appends a '2' to the end of the Form Number), and the third form would have a Form Number of "XYZ3" (appends a '3' to the end of the Form Number).

  - This is only applicable to the forms in FireLight that are set to dynamically trigger copies of themselves. An example would be a Replacement form that has been set to trigger additional copies of itself based on however many replacements are on the contract.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header/>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SubmitAutomatedBusinessApplication
xmlns="http://InsuranceTechnologies.com/Acord/ServiceContracts/V2_23_0/">
      <request>
        <TXLifeRequest xmlns="http://ACORD.org/Standards/Life/2">
          <TransRefGUID>10b41c27-9612-237c-8150-b0bcb79181cc</TransRefGUID>
          <TransType tc="103">OLI_TRANS_NBSUB</TransType>
          <OLifE>
            <Holding id="Holding_1">
              <Policy>
                <ProductCode>XXX DEFINED PRODUCT CODE</ProductCode>
                <CarrierCode>XXX DEFINED CARRIER CODE</CarrierCode>
                <Jurisdiction tc="7">OLI_USA_CO</Jurisdiction>
                <KeyedValue>
                  <KeyName>Owner_FirstName</KeyName>
                  <KeyValue>Jane</KeyValue>
                </KeyedValue>
                <KeyedValue>
                  <KeyName>Owner_MiddleName</KeyName>
                  <KeyValue>Doe</KeyValue>
                </KeyedValue>
                <KeyedValue>
                  <KeyName>Owner_LastName</KeyName>
                  <KeyValue>Smith</KeyValue>
                </KeyedValue>
              </Policy>
            </Holding>
            <Party id="Party_1">
              <FullName>someName</FullName>
              <Producer>
                <CarrierAppointment>
                  <CompanyProducerID>XXX-000000000001</CompanyProducerID>
                  <CarrierCode>XXX</CarrierCode>
                </CarrierAppointment>
              </Producer>
            </Party>
            <Relation id="Relation_1" OriginatingObjectID="Holding_1"
RelatedObjectID="Party_1">
              <OriginatingObjectType tc="4">OLI_HOLDING</OriginatingObjectType>
              <RelatedObjectType tc="6">OLI_PARTY</RelatedObjectType>
              <RelationRoleCode tc="37">OLI_REL_PRIMAGENT</RelationRoleCode>
            </Relation>
            <FormInstance>
              <FormName>Form A Name</FormName>
              <ProviderFormNumber>XXXA FormNumber</ProviderFormNumber>
            </FormInstance>
            <FormInstance>
              <FormName>Form B Name</FormName>
              <ProviderFormNumber>XXXB FormNumber</ProviderFormNumber>
              <FormVersion>XXXB FormEdition</FormVersion>
            </FormInstance>
            <FormInstance>
              <FormName>Form C Name</FormName>
              <ProviderFormNumber>XXXC FormNumber</ProviderFormNumber>
              <FormVersion>XXXC FormEdition</FormVersion>
              <OLifEExtension VendorCode="25">
                <MissingPage>2</MissingPage>
                <MissingPage>4</MissingPage>
              </OLifEExtension>
            </FormInstance>
            <FormInstance>
              <FormName>Form C Name</FormName>
              <ProviderFormNumber>XXXC FormNumber2</ProviderFormNumber>
              <FormVersion>XXXC FormEdition</FormVersion>
            </FormInstance>
            <FormInstance>
              <FormName>Form C Name</FormName>
              <ProviderFormNumber>XXXC FormNumber3</ProviderFormNumber>
              <FormVersion>XXXC FormEdition</FormVersion>
            </FormInstance>
          </OLifE>
        </TXLifeRequest>
      </request>
    </SubmitAutomatedBusinessApplication>
  </s:Body>
</s:Envelope>
```

Case B – Response

The XML Response below is an example of the Response structure to be sent back from the FireLight web service.

Notes regarding this example:

- If there are any NIGO messages, that will set the TXLifeResponse/TransResult/ResultCode to have a tc attribute set to ‘2’.
    - Additionally, there will be a TXLifeResponse/TransResult/ResultInfo/ResultInfoCode with a tc attribute set to ‘3024’.
- All errors related to the application data (generated by the form rules) will be contained in an <OLifEExtension> that has a SystemCode attribute set to “ApplicationErrors”.
    - Inside of this <OLifEExtension> will be a list of <Error> elements. Each will have a ‘dataitemid’ attribute that contains the corresponding data item’s ID, and the value of these <Error> elements will be string messages that are the error message generated by the form rules.

- All errors related to the forms composition will be contained in an <OLifEExtension> that has a SystemCode attribute set to “FormErrors”.
    - Inside of this <OLifEExtension> will be a list <Error> elements. Each will have a ‘code’ attribute that determines what kind of error it is.
        - If the code is “FormMissing”, then this error notates that the given form was expected, but was actually missing on the request payload. The ‘formnumber’ attribute is the Form’s Number as defined in the FireLight admin tool, and the ‘edition’ attribute is the Form’s Edition as defined in the FireLight admin tool.
        - If the code is “PageMissing”, then this error notates that the given form has a page missing in it, as was sent by the calling system (meaning, the request structure says that the given form is missing pages). The ‘formnumber’ attribute is the Form’s Number as defined in the FireLight admin tool, the ‘edition’ attribute is the Form’s Edition as defined in the FireLight admin tool, and the ‘missingpage’ attribute is a number indicating which page is missing, as stated in the request XML.
        - If the code is “FormUnknown”, then this error notates that the given form was sent in the request XML, but is not contained within the application (meaning, after the rules have run on the application, the given form was not present). The ‘formnumber’ attribute is the Form’s Number as defined in the request XML, and the ‘edition’ attribute is the Form’s Edition as defined in the request XML.
        - If the code is “EditionUnknown”, then this error notates that the edition noted in the request XML for the given form was not the same edition that FireLight has, as defined within the FireLight admin tool. The ‘formnumber’ attribute is the Form’s Number as defined in the FireLight admin tool, the ‘edition’ attribute is the Form’s Edition as defined in the request XML, and the ‘expected’ attribute is the Form’s Edition as defined in the FireLight admin tool.
    - Each <Error> element will have a value containing a description text, useful for readability of the XML.
- All errors related to the CanSell check will be contained in an <OLifEExtension> that has a SystemCode attribute set to “CanSellErrors”.
    - Inside of this <OLifEExtension> will be an <Error> element. It will contain the CanSell message as received form the FireLight Provider module, which would originally receive some sort or message or error from its corresponding 3rd party service call.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SubmitAutomatedBusinessApplicationResponse
xmlns="http://InsuranceTechnologies.com/Acord/ServiceContracts/V2_23_0/">
      <SubmitAutomatedBusinessApplicationResult>
        <TXLifeResponse xmlns="http://ACORD.org/Standards/Life/2">
          <TransRefGUID>11b41c37-3358-582c-8150-b1bcb79123cc</TransRefGUID>
          <TransType tc="103"/>
          <TransExeDate>2017-03-15</TransExeDate>
          <TransExeTime>15:07:07.9378636-06:00</TransExeTime>
          <TransResult>
            <ResultCode tc="2">RESULT_SUCCESSINFO</ResultCode>
            <ResultInfo>
              <ResultInfoCode tc="3024">RESULTINFO_FORMNOTGOOD</ResultInfoCode>
            </ResultInfo>
            <OLifEExtension VendorCode="25" SystemCode="ApplicationErrors">
              <Error dataitemid="PRIMARY_BENE_TOTAL">The total primary beneficiary percent
needs to be 100%</Error>
              <Error dataitemid="CONTRIBUTION_TAX_YEAR">Contribution Year is required and must
be either the current tax year or prior tax year.</Error>
              <Error dataitemid="INITIAL_PREMIUM_TOTAL">This is a required field.</Error>
              <Error dataitemid="TOTAL_TRANSFER_1035_AMOUNT">This is a required field.</Error>
              <Error dataitemid="CEDING_COMPANY_PHONE1">The US phone number is
invalid.</Error>
              <Error dataitemid="CEDING_COMPANY_PHONE2">The US phone number is
invalid.</Error>
            </OLifEExtension>
            <OLifEExtension VendorCode="25" SystemCode="FormErrors">
              <Error code="FormMissing" formnumber="155233A" edition="05/05/2016">Given form
is missing but expected.</Error>
              <Error code="PageMissing" formnumber="155233A" edition="05/05/2016"
missingpage="1">Given form has missing page.</Error>
              <Error code="PageMissing" formnumber="155233A" edition="05/05/2016"
       missingpages="3">Given form has missing page.</Error>
              <Error code="PageMissing" formnumber="155233A" edition="05/05/2016"
       missingpages="4">Given form has missing page.</Error>
              <Error code="FormUnknown" formnumber="123123" edition="05/05/2016">Form was
sent, but not expected.</Error>
              <Error code="EditionUnknown" formnumber="155233A" edition="xyz"
expected="05/05/2016">Form was sent, but given Edition was not expected.</Error>
            </OLifEExtension>
            <OLifEExtension VendorCode="25" SystemCode="CanSellErrors">
              <Error>Agent cannot sell.</Error>
            </OLifEExtension>
          </TransResult>
        </TXLifeResponse>
      </SubmitAutomatedBusinessApplicationResult>
    </SubmitAutomatedBusinessApplicationResponse>
  </s:Body>
</s:Envelope>
```

Case C – Error in Processing

If the FireLight web service has an error within its processing, it will send back a response that contains a <ResultCode> with a 'tc' attribute set to '5', indicating failure. The below is an example of this – the request sent in an invalid CarrierCode that FireLight did not recognize. Note that Jurisdiction errors and CUSIP errors, as well as any other xml structural errors (meaning, the required ACORD XML elements were not present) will also generate this error.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SubmitAutomatedBusinessApplicationResponse
xmlns="http://InsuranceTechnologies.com/Acord/ServiceContracts/V2_23_0/">
      <SubmitAutomatedBusinessApplicationResult>
        <TXLifeResponse xmlns="http://ACORD.org/Standards/Life/2">
          <TransResult>
            <ResultCode tc="5">RESULT_FAILURE</ResultCode>
            <ResultInfo>
              <ResultInfoDesc>Invalid organization code 'XYZ'.</ResultInfoDesc>
              <ResultInfoSource>System</ResultInfoSource>
            </ResultInfo>
          </TransResult>
        </TXLifeResponse>
      </SubmitAutomatedBusinessApplicationResult>
    </SubmitAutomatedBusinessApplicationResponse>
  </s:Body>
</s:Envelope>
```

Case D – Successful Transaction

If the transaction successfully processed through FireLight, a success message will be sent back. The following is an example – the ResultCode element will contain a tc attribute of '1', which indicates success. Success is defined as the request being properly received, an application was created based off the data received in the request, no CanSell errors were generated, no forms errors were received, the rules for the application were run, no 'NIGO' errors were present as a result of the rules, and the application transitioned to a state for end-processing.

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SubmitAutomatedBusinessApplicationResponse
xmlns="http://InsuranceTechnologies.com/Acord/ServiceContracts/V2_23_0/">
      <SubmitAutomatedBusinessApplicationResult>
        <TXLifeResponse xmlns="http://ACORD.org/Standards/Life/2">
          <TransRefGUID>23b63c49-9999-367c-8260-b2bcb39182cc</TransRefGUID>
          <TransType tc="103"/>
          <TransExeDate>2017-03-28</TransExeDate>
          <TransExeTime>15:07:56.0501760-06:00</TransExeTime>
          <TransResult>
            <ResultCode tc="1">RESULT_SUCCESS</ResultCode>
          </TransResult>
        </TXLifeResponse>
      </SubmitAutomatedBusinessApplicationResult>
    </SubmitAutomatedBusinessApplicationResponse>
  </s:Body>
</s:Envelope>
```

Integration Changes

Any integration changes needed will be specific to each client. May include:

A. Any custom response pieces/structure – certain examples could include application identifiers such as DTI #'s, Policy Numbers, etc. Basically, anything not previously defined would require the FireLight Provider modules (which handle integration) to customize the response, and would therefore be considered an integration change.

B. Any CanSell checks that require a third-party web service call, that is not already set up, will need to be completed before the response to this web service would contain any CanSell errors.

C. Any application data mapping that should take place outside of the means defined within this document will have to be set up within the FireLight Provider modules. Note that existing mapping can be used.

D. Any outbound data processing that should take place as a result of application completion must be accomplished through the normal process, which is handled by the FireLight Provider modules. Note that existing processes can be used.

Areas Impacted

System Area	Yes	Comment
Admin Tool		
- **Form Library**		
- **Design Forms**		
- **Profile Administration**		
- **Reports**		
- **Deployment**		
FireLight App		
- **New Application**	X	
- **Edit Application**		
- **Signature Process**		
- **Review Queue**		
- **Manual Review**		
- **User Preferences**		
- **Inbound Integration**	X	Doesn't affect existing inbound integration, but still uses it
- **Outbound Integration**	X	Doesn't affect existing outbound integration, but still uses it
- **PDF Generation**		
- **Email System**		
FireLight Console		
- **Windows**		
- **iOS**		
Other Systems		
- **DTCC Integration**		
- **Commission Netting**		
- **Activity Reporting**		