

iConnect 189466 Design Approach - EFT Validation using GIACT's gVerify and gAuthenticate

Project Overview

This project will integrate FireLight with GIACT's gVerify and gAuthenticate web services to provide bank account verification and authentication. These services can be used for one time and reoccurring ACH payments that users collect within Firelight.

gVerify is GIACT's bank account verification service that verifies the current status of bank accounts in real-time and reports any negative history associated with the account. Inquiries on currently assigned routing numbers will return the bank name assigned to the routing number.

The gVerify service returns an account response code indicating:

- Open and valid account
- Account in NSF status or excessive returns
- Closed or frozen account
- Account not located
- Non-Demand Deposit account that does not allow posting of ACH debits
- Special routing numbers like Government institutions
- American Express Travelers Cheque accounts
- Stop Payment Orders on a specific check number or check number range

gAuthenticate is GIACT's bank account authentication service takes the gVerify one step further to authenticate the customer data against the account. This service works with both personal and business bank accounts.

The gAuthenticate service will authenticate:

- Person name or business name
- Tax ID Number (SSN/ITIN/EIN)
- Date of Birth
- Address
- Phone Number

The information returned from GIACT's web services will be available to be leveraged in the FireLight rules engine for additional configuration of the service. For example, clients will be able to write rules to limit the number of attempts to validate the account before enforcing another form of payment. Clients can configure that after three attempts, present an error message that requires the user to select a different funding source.

Clients will contract with GIACT to utilize their services. There will be no additional cost within FireLight. Time and materials for implementation specific rules still apply.

Requirements / User Stories

Admin: Add GIACT to Third Party Service Configuration > Service Type dropdown

Add GIACT to the Third Party Service Configuration menu in the Service Type dropdown for a FireLight admin to setup the GIACT web services.

When a user adds a service type of GIACT in the third party service configuration and enters a username and password applicable to GIACT it should apply to the GIACT web service.

If distributor organization does not have GIACT credentials, look at carrier organization for credentials and use them if applicable.

Acceptance Criteria

- GIACT appears in the Service Type dropdown in the admin.

Admin: Add GIACT to "Custom Action Type" dropdown in button field properties.

Add GIACT to "Custom Action Type" dropdown in the button field properties for selection by the form designer.

When button in Activity is clicked the GIACT service should be called.

Custom Action Set-up Instructions:

Add a Custom Action Button to your form or wizard

The Data item name is only needed for rules. Name this whatever you want. (Example: GIACT1)

Set the Data Group field to EFT Validation 1 (1-5 are supported, enumerate according to # of Accounts being validated)

Set the Custom Action field to "Custom Action"

Set the Custom Action Type to "GIACT" (This is a drop-down)

Set the Custom Action Parameter to "GIACT" (You must type this value.)

Custom Action Properties

☐ Place Field on a New Row

Name

GIACT1

Text

Validate ACH Information

Width

6

Mobile Width

Enter a Value

Class Name

Enter a Value

Help Text

Enter a Value

Data Group

EFT Validation 1

Data Group Property

None

Custom Action

Custom Action

Custom Action Type

GIACT

Custom Action Parameter

GIACT

CANCEL

OK

Set the Data Group and Data Group Properties for all of the required fields

The Data Group should be set to the same value defined as the Custom Action Data Group (EFT Validation 1 - 5)

The Data Group Properties required are: First name and Last name or Business Name; Bank Name, Account Number; Routing Number

Text Box Properties

Name AccountHolder_RoutingNumber
Text Routing Number
Width 5
Mobile Width Enter a Value
Class Name Enter a Value
Help Text Enter a Value
Input Type None ▼
Text Mask None ▼
SSN Format None ▼
Data Group EFT Validation 1 ▼
Data Group Property Routing Number ▼
Maximum Length 100
Number Of Lines 1

CANCEL
OK

Acceptance Criteria

- GIACT is shown in Custom Action Type dropdown for button field properties.
- Button click should call the server

Admin: Add EFT Validation 1 - 5 data groups in field properties

Add EFT Validation 1-5 data groups in field properties for utilizing the GIACT web services.

Acceptance Criteria

- EFT Validation 1, EFT Validation 2, EFT Validation 3, EFT Validation 4, and EFT Validation 5 are available data groups in field properties.

Admin: Make data group properties dynamic based on data group selected

Make the data group properties dynamic by the data group that is selected. When the user selects the "GIACT #" data group, only the following data group properties should show:

- Routing Number
- Account Number
- Check Number
- Amount
- Name Prefix
- First Name
- Middle Name
- Last Name
- Name Suffix
- Business Name
- Address Line 1
- Address Line 2
- City
- State
- Zip Code
- Home Phone Number
- Work Phone Number
- Tax ID
- Date Of Birth
- ID Type
- DL Number
- DL State

- Email Address
- Bank Account Type
- Country
- gVerify Account Response Code
- gVerify Verification Response
- Bank Name
- Bank Account Added Date
- Bank Account Last Updated Date
- Bank Account Closed Date
- gAuthenticate Customer Response Code
- gAuthenticate Verification Response

Acceptance Criteria

- The following data group properties are available in the data group properties selection of field properties:
 - Routing Number
 - Account Number
 - Check Number
 - Amount
 - Name Prefix
 - First Name
 - Middle Name
 - Last Name
 - Name Suffix
 - Business Name
 - Address Line 1
 - Address Line 2
 - City
 - State
 - Zip Code
 - Home Phone Number
 - Work Phone Number
 - Tax ID
 - Date Of Birth
 - ID Type
 - DL Number
 - DL State

- Email Address
- Bank Account Type
- Country
- gVerify Account Response Code
- gVerify Verification Response
- Bank Name
- Bank Account Added Date
- Bank Account Last Updated Date
- Bank Account Closed Date
- gAuthenticate Customer Response Code
- gAuthenticate Verification Response

Integrate with gVerify web service to verify account is active/in good standing

Implement the ability to connect to the GIACT gVerify web service to verify the account is active and in good standing. Within FireLight we will utilize data group and data group properties for the data to pull for the API.

Acceptance Criteria

- Able to connect to the gVerify web service successfully.

Store gVerify information as data group properties to be used for validation

Store the following information sent back from the gVerify web service in data group properties for rule use:

- gVerify Account Response Code
- gVerify Verification Response
- Bank Name
- Bank Account Added Date
- Bank Account Last Updated Date
- Bank Account Closed Date

Acceptance Criteria

- gVerify Account Response Code can be seen on forms if form has a textbox with the gVerify Account Response Code data group property.
- Rules utilizing the gVerify Account Response Code data group property work as expected.
- gVerify Verification Response can be seen on forms if form has a textbox with the gVerify Verification Response data group property.
- Rules utilizing the gVerify Verification Response data group property work as expected.

- Bank Name can be seen on forms if form has a textbox with the Bank Name data group property.
- Rules utilizing the Bank Name data group property work as expected.
- Bank Account Added Date can be seen on forms if form has a textbox with the Bank Account Added Date data group property.
- Rules utilizing the Bank Account Added Date data group property work as expected.
- Bank Account Last Updated Date can be seen on forms if form has a textbox with the Bank Account Last Updated Date data group property.
- Rules utilizing the Bank Account Last Updated Date data group property work as expected.
- Bank Account Closed Date can be seen on forms if form has a textbox with the Bank Account Closed Date data group property.
- Rules utilizing the Bank Account Closed Date data group property work as expected.

Integrate with gAuthenticate web service to authenticate customer data

Implement the ability to connect to the GIACT gAuthenticate web service to verify customer data is valid against account. Within FireLight we will utilize data group and data group properties for the data to pull for the API.

Acceptance Criteria

- Able to connect to the gAuthenticate web service successfully.
- Only call this service when needed.

Store gAuthenticate information as data group properties to be used for validation

Store the following information sent back from the gAuthenticate web service in data group properties for rule use:

- gAuthenticate Customer Response Code
- gAuthenticate Verification Response
-

Acceptance Criteria

- gAuthenticate Customer Response Code can be seen on forms if form has a textbox with the gAuthenticate Customer Response Code data group property.
- Rules utilizing the gAuthenticate Customer Response Code data group property work as expected.
- gAuthenticate Verification Response can be seen on forms if form has a textbox with the gAuthenticate Verification Response data group property.
- Rules utilizing the gAuthenticate Verification Response data group property work as expected.

Add detailed activity report on volume of transactions of GIACT web service(s) executed

Implement a detailed activity report that can be sent daily, weekly, or monthly that lists all transactions that have an account response code (and customer response code if applicable) which will provide the volume of transactions utilizing the GIACT web services.

FireLight will report for each individual GIACT call out, even if there are multiple call outs in one transaction. There could be multiple records for the same transaction ID.

Fields in Report:

- Transaction Type (eApp, eDelivery, etc.)
- Agent ID
- Distributor Transaction ID (AppID)
- Distributor/Carrier
 - Date
 - Account Response Code (gVerify)
 - Customer Response Code (gAuthenticate)

Acceptance Criteria

- GIACT Web Service detailed activity report is available and contains the following columns:
 - Transaction Type (eApp, eDelivery, etc.)
 - Agent ID
 - Distributor Transaction ID (AppID)
 - Distributor/Carrier
 - Date
 - Account Response Code (gVerify)
 - Customer Response Code (gAuthenticate)
- Reports all cases that have an account response code (have utilized the gVerify web service)

Support the ability to require a button to be selected through a rule

This enhancement will introduce the ability to require a button be used through a rule and keep the user from progressing to signatures if they have not yet interacted with it.

Have a counter or indicator that will relay if a button has been clicked yet or not (i.e. dataitem that has a 1 or 0. If 1, the button has been clicked and is no longer required. Rules could be written so that if the user changes certain information, the dataitem could be changed back to 0 to require that they interact with the button again.)

Use case: Once required information is filled out on a form to do a call out, a button becomes enabled and required. If the user continues without clicking the button they will be able to get the application to 99%, but not proceed until they go back and click the button.

Acceptance Criteria

- If there is a required rule written for a button, will show as red.
- Required button will be accounted for in % complete.
- Required button will work for all button functionality (CRM, Custom Action, Print)

Build rule template to enable EFT button when required information has been entered

Build a rule that will keep the EFT button disabled until all required info is entered for both gVerify and gAuthenticate (Routing Number, Account Number, and First Name/Last Name or Business Name) to ensure that FireLight has gathered appropriate information before initiating the call.

Sample Rule:

```
<setcontrolstate dataitemid="GIACT!">
<and>
  <compare op="!=">
    <const value="" type="String" />
    <diget dataitemid="giactAccountNum" />
  </compare>
  <compare op="!=">
    <const value="" type="String" />
    <diget dataitemid="giactRoutingNum" />
  </compare>
</or>
<and>
  <compare op="!=">
    <const value="" type="String" />
    <diget dataitemid="FirstName" />
  </compare>
  <compare op="!=">
```

```

    <const value="" type="String" />
    <diget dataitemid="LastName" />
  </compare>
</and>
<compare op="!=">
  <const value="" type="String" />
  <diget dataitemid="BusinessName" />
</compare>
</or>
</and>
</setcontrolstate>

```

Acceptance Criteria

- EFT button is disabled on forms until Routing Number, Account Number, and First Name/Last Name or Business Name is entered.
- EFT button enables and can be used once Routing Number, Account Number, and First Name/Last Name or Business Name is entered.

Build rule template to display generic message to user if there is a failure response code

No message will be displayed to the user if the web service validation(s) are successful. However, if there are any failures than we will present a generic message to the user "We are unable to process this account at this time. Please provide alternate account or select a different funding source." As some of the direct GIACT messages may pose issues displaying to advisors.

The user will not be able to get the application to 100% unless they receive a pass code from gVerify (and gAuthenticate if applicable).

Sample Rules:

If you want to ensure GIACT is selected at least 1 time, then set it to Required:

Sample Rule:

```

<if>
  <condition>
    <not>
      <isnullorwhitespace>

```

```

    <diget dataitemid="GIACT!" />
  </isnullorwhitespace>
</not>
</condition>
<removemessage dataitemid="GIACT!" />
<else />
<postmessage dataitemid="GIACT!">
  <const value="You must run the GIACT service at least once." />
</postmessage>
</if>

```

If you want to force another payment selection when the Account is not found, here is a sample Rule.

```

<if>
  <condition>
    <compare op="==">
      <const value="ND00" type="String" />
      <diget dataitemid="Client_gVerifyAccountResponseCode" />
    </compare>
  </condition>
  <postmessage dataitemid="GIACT_Button">
    <const value="We are unable to find this account at this time. Please provide alternate account or
select a different funding source." />
  </postmessage>
<else />
  <removemessage dataitemid="GIACT_Button" />
</if>

```

Acceptance Criteria

- Generic error message is displayed on data entry when a failure is passed back from the gVerify/gAuthenticate web service(s).
- Button is required if there is no code (web service has not been ran yet).

Build rule to limit x attempts to run EFT service for security purposes

Build a rule to limit x attempts to run EFT service for security purposes. If x failures have occurred, throw an error on the application requiring the user to select a different funding source.

For example, a client wants to allow the agent 3 attempts to run the web service. If it fails after 3 attempts, lock down the EFT button.

Example rule (assumes a dropdown to select funding type):

```
<block>
  <setcontrolstate dataitemid="GetFunding">
    <and>
      <compare op="==">
        <const value="EFT" type="String" />
        <diget dataitemid="FundingSource" />
      </compare>
      <compare op="!=">
        <const value="3" type="String" />
        <diget dataitemid="GetFunding" />
      </compare>
    </and>
  </setcontrolstate>
</if>
<condition>
  <and>
    <compare op="==">
      <const value="EFT" type="String" />
      <diget dataitemid="FundingSource" />
    </compare>
    <compare op="==">
      <const value="3" type="String" />
      <diget dataitemid="GetFunding" />
    </compare>
  </and>
</condition>
```

```

</condition>
<postmessage dataitemid="FundingSource">
  <const value="Maximum EFT attempts reached. Please select a different funding source." />
</postmessage>
<else />
  <removemessage dataitemid="FundingSource" />
</if>
</block>

```

Simple rule (button-only):

```

<setcontrolstate dataitemid="GetFunding">
  <compare op="!=">
    <const value="3" type="String" />
    <diget dataitemid="GetFunding" />
  </compare>
</setcontrolstate>

```

Acceptance Criteria

- If the web service fails # of times defined in the rule, present error to user and do not allow them to use ACH funding.

Build rule that requires EFT button if information on required fields has changed

Build a rule that requires the EFT button be clicked again if information on required fields has changed (Routing Number, Account Number, and First Name/Last Name or Business Name).

Sample Rule to clear the original GIACCT responses if it is being rerun.

```

<if>
  <condition>
    <and>
      <or>
        <compare op="!=">
          <const value="" type="String" />
          <diget version="Current" dataitemid="TextBox_12" />

```

```
</compare>
<compare op="!=">
  <const value="" type="String" />
  <diget version="Current" dataitemid="TextBox_13" />
</compare>
</or>
<or>
  <compare op="!=">
    <diget version="Original" dataitemid="TextBox_12" />
    <diget version="Current" dataitemid="TextBox_12" />
  </compare>
  <compare op="!=">
    <diget version="Original" dataitemid="TextBox_13" />
    <diget version="Current" dataitemid="TextBox_13" />
  </compare>
</or>
</and>
</condition>
<removemessage dataitemid="GIACT!" />
<setcontrolstate dataitemid="GIACT!">
  <const value="false" type="Boolean" />
</setcontrolstate>
</if>
```

If you want to require the button to be re-clicked if the data is changed, here is a sample rule:

```
<if>
  <condition>
    <or>
      <compare op="!=">
        <diget version="Original" dataitemid="giactAccountNum" />
```

```

    <diget version="Current" dataitemid="giactAccountNum" />
  </compare>
  <compare op="!=">
    <diget version="Original" dataitemid="giactRoutingNum" />
    <diget version="Current" dataitemid="giactRoutingNum" />
  </compare>
  <compare op="!=">
    <diget version="Original" dataitemid="FirstName" />
    <diget version="Current" dataitemid="FirstName" />
  </compare>
  <compare op="!=">
    <diget version="Original" dataitemid="LastName" />
    <diget version="Current" dataitemid="LastName" />
  </compare>
  <compare op="!=">
    <diget version="Original" dataitemid="BusinessName" />
    <diget version="Current" dataitemid="BusinessName" />
  </compare>
</or>
</condition>
<postmessage dataitemid="GIACT!">
  <const value="Changing routing number, account number, first name, last name, or business name
requires the EFT button be selected to run the call again." />
</postmessage>
<setcontrolstate dataitemid="GIACT!">
  <const value="true" type="Boolean" />
</setcontrolstate>
<else />
<removemessage dataitemid="GIACT!" />
<setcontrolstate dataitemid="GIACT!">
  <const value="false" type="Boolean" />

```


</setcontrolstate>

</if>

Acceptance Criteria

- Changing routing number, account number, first name, last name, or business name requires the EFT button be selected to run the call again.