

EDQ FOR SALESFORCE COMMERCE CLOUD

CARTRIDGE GUIDE

EDQ FOR SALESFORCE COMMERCE CLOUD 18.1.0



AUTHOR	JASON WALSTAD
CREATION DATE	DECEMBER 20, 2017
LAST UPDATED	JANUARY 22, 2017
VERSION	VERSION 2

TABLE OF CONTENTS

1. SOLUTION OVERVIEW	4
1.1 INTRODUCTION	4
1.2 SUPPORTED VERSIONS	4
1.3 FUNCTIONALITY	4
REAL TIME ADDRESSSS VALIDATION	4
REAL TIME EMAIL VALIDATION	6
1.4 SUPPORTED TOUCH POINTS	8
1.5 CONFIGURATION OPTIONS	8
2. TECHNICAL OVERVIEW	9
2.1 ARCHITECTURE / DEPLOYMENT CONSIDERATIONS	9
2.2 EDQ PRODUCT MAINTENANCE	9
3. DEPLOYMENT GUIDE	10
3.1 DEPLOYMENT OVERVIEW	10
3.2 PREREQUISITES	10
3.3 SALESFORCE COMMERCE CLOUD ENVIRONMENT ACCESS REQUIRED	10
4. EDQ FOR SALESFORCE COMMERCE CLOUD CARTRIDGE DOWNLOAD	10
5. MAIN INTEGRATION DEPLOYMENT STEPS	10
5.1 EDQ CARTRIDGE INSTALLATION	11
ADD THE CARTRIDGE TO YOUR SITE	11
ENVIRONMENTAL UPDATES	11
5.2 COMMERCE CLOUD BUSINESS MANAGER METADATA IMPORT	12
5.3 SALESFORCE COMMERCE CLOUD BUSINESS MANAGER SERVICES IMPORT	13
5.4 INTEGRATING WITH CONTROLLERS	14
CORE INTEGRATION	14
DISABLE EMAIL VALIDATION	17
DISABLE ADDRESS VALIDATION	17
5.5 INTEGRATING WITH PIPELINES	18
ADDRESS VALIDATION: SHIPPING ADDRESS	18
ADDRESS VALIDATION: BILLING ADDRESS	19
ADDRESS VALIDATION: ADDRESS BOOK	20

6. TESTING YOUR INTEGRATION	26
6.1 ADDRESS VALIDATION	26
6.2 EMAIL VALIDATION	27
7. EDQ MAINTENANCE INFORMATION	29
8. RELEASE NOTES	30
9. EDQ PRODUCT APPENDIX	31
9.1 ADDRESS VALIDATION MATCH TYPES AND EXAMPLES	31
9.2 EMAIL VALIDATION MATCH TYPES AND EXAMPLES	32

1. SOLUTION OVERVIEW

1.1 INTRODUCTION

Experian Data Quality for Salesforce Commerce Cloud verifies customer address and email data during checkout and within the Create/Edit Account and Address Book touchpoints. Most often, customers are unaware that the tool is working in the background. If the tool cannot completely verify and standardize the address or email, the customer will be prompted for missing information.

1.2 SUPPORTED VERSIONS

The EDQ for Commerce Cloud package is certified by Salesforce Commerce Cloud and can be used with the latest version of Commerce Cloud SiteGenesis Storefront.

1.3 FUNCTIONALITY

The Experian Data Quality for Commerce Cloud integration contains both real time address and email validation functionality. The sections below describe each of these validation workflows in more detail.

REAL TIME ADDRESS VALIDATION

EDQ for Commerce Cloud's real time address validation can verify that a given address exists within the postal data from that country's postal authority. Once a full, verified address is captured, the address can be formatted and standardized automatically, ensuring that captured data is both clean and accurate.

EDQ for Commerce Cloud's real time address validation uses our best practices workflow. This workflow is intended for use within external, customer facing websites where customer interaction must be simple and intuitive.

The majority of input postal addresses will be validated automatically without any user interaction required. The remaining addresses will trigger one of five different user interactions that are specific to the issues with the input address. Lists of optional addresses are only shown when necessary and users can override the validation if they are unhappy with the suggested result. This allows the address to be corrected at point of capture with minimal effort by the user or customer.

In the following example, an address is entered on the checkout page's Shipping Address touchpoint:

STEP 1: Shipping
STEP 2: Billing
STEP 3: Place Order

SELECT OR ENTER A SHIPPING ADDRESS • *required*

• First Name

John

• Last Name

Smith

• Address 1

98 De Pasquale Ave

(APO/FPO)

Address 2

• Country

United States ▼

• State

Rhode Island ▼

• City

Providence

• Zip Code

02901

• Phone

6173856890

(Why is this required?)

Example: 333-333-3333

☐ Use this address for Billing

Is this a gift?

☐ Yes ☒ No

SELECT SHIPPING METHOD

☒ USPS: \$19.00 (Details)

CONTINUE >

Upon clicking Continue, EDQ Address Validation will be triggered. In this scenario, we will find that the input address belongs to an apartment building, and that the user did not enter any apartment level details with their address. This results in the following dialog:

Sorry, we think your apartment/suite/unit is missing or wrong.
To proceed, please enter your apartment/suite/unit or use your address as entered.

Confirm your Apartment/Suite/Unit number:

Apt:

[Submit](#)

You Entered: [\[Edit\]](#)

Address 1: 98 De Pasquale Ave

Address 2:

City: Providence

State: RI

Postal Code: 02901

[Use Original](#)

[Display all potential matches](#)

In this screen, the user has the option to update their information to include their apartment details, or, they can accept their original address as it was entered initially. When the correct apartment number is captured, the service returns the complete and accurate address including standardization and the ZIP/+4 code.

REAL TIME EMAIL VALIDATION

EDQ for Commerce Cloud's real time email validation can verify that a given email exists and can currently accept mail. Additionally, our correction logic can catch common typing mistakes and recommend suggestions.

Similar to real time address validation, EDQ for Commerce Cloud's real time email validation also uses our best practices workflow. This workflow is intended for use within external, customer facing websites where customer interaction must be clear and simple.

Many times, email addresses are entered accurately on the first try. The remaining addresses will trigger different user interactions that are specific to the issues with the email. Suggested corrections are only shown when necessary and users can override the validation if they are unhappy with the suggested result. This allows the information to be corrected at point of capture with minimal effort.

In the following example, an email is entered on the Create Account touchpoint:

Create Account

NAME • *required*

• First Name John

• Last Name Smith

EMAIL / LOGIN INFORMATION • *required*

• Email john.smith987@gmail.com

• Confirm Email john.smith987@gmail.com

• Password
8 - 255 characters

• Confirm Password

☐ Please add me to the Demandware's email list.
Demandware does not share or sell personal info. [See Privacy Policy](#)

Apply

Upon clicking Apply, EDQ Email Validation will be triggered. In this scenario, we will find that the input email domain of gmail.com is not valid, and that the user most likely intended to enter gmail.com. This results in the following dialog:

Sorry we could not confirm your e-mail address
To proceed, please confirm your e-mail address below.

Confirm or edit your e-mail address

john.smith987@gmail.com

Did you mean?

john.smith987@gmail.com

Update

Accept Original

1.4 SUPPORTED TOUCH POINTS

EDQ may be used to validate addresses in the following touchpoints:

- Checkout – Billing Address
- Checkout – Shipping Address
- Address Book – Addresses

EDQ may be used to validate email addresses in the following touchpoints:

- Checkout – Billing Address
- Register – Create an Account
- Account – Edit Account

1.5 CONFIGURATION OPTIONS

The EDQ for Commerce Cloud package installs a tab within the Custom Site Preferences for your Commerce Cloud site.

This tab contains the EDQ Credentials for both Address and Email Validation, as well as some EDQ specific options:

[Merchant Tools](#) > [Site Preferences](#) > [Custom Site Preferences](#) > QAS Config

Custom Site Preferences

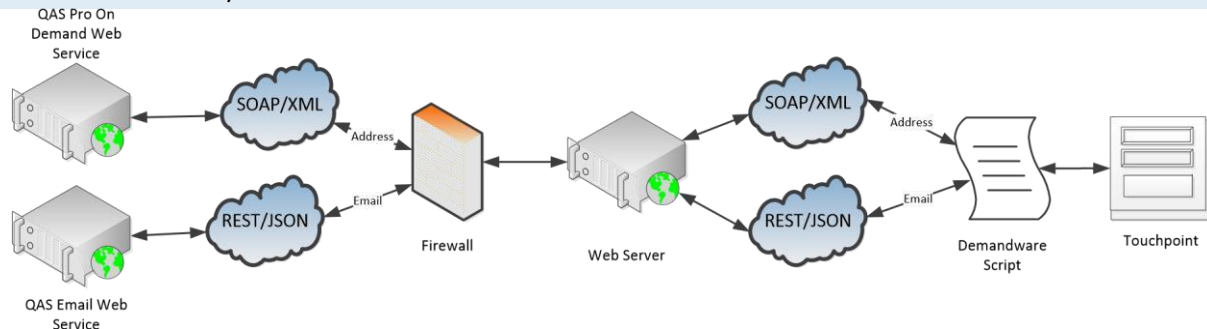
This page allows you to define the custom site preferences of group "QAS Config". For each Demandware instance type, you can enter a different preference value. Select the desired instance type from the list. If no value is defined for a preference, the system uses the defined default value. To save your settings, click **Apply**.

Preference Name	Value	Default Value
QAS On Demand Endpoint:	<input type="text" value="https://ws2.ondemand.qas.com/ProOnDemand/V3/Pro"/>	https://ws2.ondemand.qas.com/ProOnDemand/V3/ProOnD ...
	URL to hit the QAS On Demand Address Verification Service	
QAS On Demand Token:	<input type="text" value=""/>	
	credentials for the QAS On Demand Address Verification Service	
QAS On Demand USA Layout:	<input type="text" value="USA2 TitleCase Retention DPV"/>	USA2 TitleCase Retention DPV
QAS On Demand CAN Layout:	<input type="text" value="CAN2 TitleCase Retention LVR"/>	CAN2 TitleCase Retention LVR
QAS Enable Email Workflow:	<input type="text" value="active"/>	active
	when active, Email verification Service is used (values: active, passive, or inactive)	
QAS Email Token:	<input type="text" value=""/>	
	credentials for the QAS Email Verification Service	

[Apply](#) [Reset](#)

2. TECHNICAL OVERVIEW

2.1 ARCHITECTURE / DEPLOYMENT CONSIDERATIONS



The EDQ for Commerce Cloud integration contains two main components:

First, the Site Preferences and Service configurations for EDQ are imported into the system. A metadata package is imported into the Site to create the Custom Site Preferences options, which are used for storing EDQ credential information. A second package containing the configurations for the EDQ Email and Address verifications web services is also imported into the Site. This fits into the Commerce Cloud Service Framework.

Second, a cartridge is provided which contains pipelines, web services definition files, isml templates, and scripts. These files perform the validation logic.

Once all components are in place, call nodes are added to the pipelines for each touchpoint to enable the EDQ process to be triggered. These call nodes route to Commerce Cloud script files, which set up web service calls to the Address and Email services. Once a response is received, the Commerce Cloud script files handle the response and trigger the appropriate user workflows.

2.2 EDQ PRODUCT MAINTENANCE

The EDQ for Commerce Cloud integration utilizes Experian Data Quality’s SaaS-based address and email services. These services are monitored closely at all times. As these services are hosted, maintenance such as bug fixes and data updates are handled by Experian Data Quality.

Experian Data Quality will provide access details to the address and email services during the integration.

For information related to SLAs of the EDQ hosted services, please refer to our support site below:

http://support.qas.com/qas_pro_on_demand_sla_2079.htm

3. DEPLOYMENT GUIDE

3.1 DEPLOYMENT OVERVIEW

This document provides an overview of integrating and configuring address and email verification within Commerce Cloud. Deployment includes support for both Pipeline and Controller based sites using the int_QAS and int_QASc cartridges accordingly.

3.2 PREREQUISITES

Prior to performing the integration steps below, ensure that you have access to the following:

- QAS Pro On Demand token (only applicable if integrating Address Validation)
- QAS Email token (only applicable if integrating Email Validation)

3.3 SALESFORCE COMMERCE CLOUD ENVIRONMENT ACCESS REQUIRED

This integration process will require access to the following:

- Business Manager
- Commerce Cloud UX Studio

4. EDQ FOR SALESFORCE COMMERCE CLOUD CARTRIDGE DOWNLOAD

To begin, download the EDQ for Commerce Cloud Cartridge from the Commerce Cloud XChange:

<https://xchange.Commerce Cloud.com/docs/DOC-10516>

5. MAIN INTEGRATION DEPLOYMENT STEPS

The EDQ for Commerce Cloud integration contains two main components:

First, the Site Preferences and Service configurations for EDQ are imported into the system. A metadata package is imported into the Site to create the Custom Site Preferences options, which are used for storing EDQ credential information. A second package containing the configurations for the EDQ Email and Address verifications web services is also imported into the Site. This fits into the Salesforce Commerce Cloud Service Framework.

Second, a cartridge is provided which contains pipelines, web services definition files, isml templates, and scripts. These files perform the validation logic.

Once all components are in place, call nodes are added to the pipelines for each touchpoint to enable the EDQ process to be triggered. These call nodes route to Commerce Cloud script files, which set up web service calls to the Address and Email services. Once a response is received, the Commerce Cloud script files handle the response and trigger the appropriate user workflows.

5.1 EDQ CARTRIDGE INSTALLATION

The EDQ for Commerce Cloud main cartridge for Controller-based sites is titled `int_QASc`. This cartridge has controllers, web services definition files, ISML templates, and scripts. The files can be changed to suit the application or client's requirements.

ADD THE CARTRIDGE TO YOUR SITE

After adding the the `int_QASc` cartridge into your Commerce Cloud folder structure and uploading it to the server, you should add the server to your cartridge path for all the storefronts you wish to use EDQ for Commerce Cloud. Go to Administration > Sites > Your Site > Settings, and add `int_QASc` to the front of your cartridge path.

Administration > Sites > Manage Sites > Your Site - Settings

General Settings Cache Site Status

OptarosControllerPort - Settings

Click Apply to save the details. Click Reset to revert to the last saved state.

Instance Type: Sandbox/Development

Deprecated. The preferred way of configuring HTTP and HTTPS hostnames is by using new features of the site aliases configuration ("Site URLs/Aliases Configuration"). The HTTP/HTTPS hostnames values set in this section will be used if no hostnames are defined by aliases configuration and are intended only to support an older configuration style.

HTTP Hostname:

HTTPS Hostname:

Instance Type: All

Cartridges: int_QASc:app_storefront_controllers:app_storefront_pipelines:app_storefront

Effective Cartridge Path: int_QASc:app_storefront_controllers:app_storefront_pipelines:app_storefront_core:plugin_apple_pay:plugin_facebook:plugin_pinterest_commerce:bc_api:core

Apply Reset

<< Back to List

Note: if your site is using Pipelines instead of Controllers, use the "int_QAS" cartridge instead of "int_QASc"

ENVIRONMENTAL UPDATES

The following files can be altered for your environment as needed:

CSS

In `static/default/css/qas.css`, changes can be made to adjust the look and feel of the user interaction pages and to give them the same appearance as the site being integrated into.

USER INTERACTION MESSAGING

In `templates/resources/message.properties` all the dialog displayed on the user interaction pages is stored. All changes to dialog displayed on the page should be made here.

5.2 COMMERCE CLOUD BUSINESS MANAGER METADATA IMPORT

To set up the QAS integration configuration in Commerce Cloud Business Manager, upload and import the QASitePreferencesImport.xml file:

- Upload: Administration > Site Development > Import & Export > Import & Export Files > Upload
- Import: Administration > Site Development > Import & Export > Meta Data > Import

The following attributes will be added (by the meta data) to System object “Site Preferences” under *Administration > Site Development > System Object Definitions*. You can then access / update them under Site > Site Preferences > Custom Site Preferences > QAS Config. Except where noted, you can keep these values as the default.

- QAS Address Verification (enum-of-strings): Set to enable/disable address validation functionality
- QAS On Demand Token (String): credentials for the QAS On Demand Address Verification Service
 - Set to the value supplied to you by your Experian Partner
- QAS On Demand Endpoint (String): the endpoint for the QAS On Demand Address Verification Service
 - Set to <https://ws2.ondemand.qas.com/ProOnDemand/V3/ProOnDemandService.asmx>
- QAS On Demand USA Layout (String)
- QAS On Demand CAN Layout (String)
- QAS Email Verification (enum-of-strings): Set to enable/disable email validation functionality
- QAS Email Token (String): credentials for the QAS Email Verification Service
 - Set to the value supplied to you by your Experian Partner
- QAS Workflow (String : active, passive, inactive) – Enables/disables email validation functionality (Only for use with the pipelines-based int_QAS cartridge)

[Merchant Tools](#) > [Site Preferences](#) > [Custom Site Preferences](#) > QAS Config

Custom Site Preferences

This page allows you to define the custom site preferences of group "QAS Config". For each Demandware instance type, you can enter a different preference value. Select the desired instance type from the list. If no value is defined for a preference, the system uses the defined default value. To save your settings, click **Apply**.

Instance Type: Sandbox/Development

Preference Name	Value	Default Value
QAS On Demand Endpoint:	<input type="text" value="https://ws2.ondemand.qas.com/ProOnDemand/V3/ProD"/>	https://ws2.ondemand.qas.com/ProOnDemand/V3/ProOnD
	URL to hit the QAS On Demand Address Verification Service	
QAS On Demand Token:	<input type="text" value=""/>	
	credentials for the QAS On Demand Address Verification Service	
QAS On Demand USA Layout:	<input type="text" value="USA2 TitleCase Retention DPV"/>	USA2 TitleCase Retention DPV
QAS On Demand CAN Layout:	<input type="text" value="CAN2 TitleCase Retention LVR"/>	CAN2 TitleCase Retention LVR
QAS Enable Email Workflow:	<input type="text" value="active"/>	active
	when active, Email verification Service is used (values: active, passive, or inactive)	
QAS Email Token:	<input type="text" value=""/>	
	credentials for the QAS Email Verification Service	

[Apply](#) [Reset](#)

5.3 SALESFORCE COMMERCE CLOUD BUSINESS MANAGER SERVICES IMPORT

To set up the EDQ Services in Commerce Cloud Business Manager upload and import the QASServicesImport.xml file.

- Upload: Administration > Operations > Import & Export > Import & Export Files > Upload
- Import: Administration > Operations > Import & Export > Services > Import

This will add the all the Services along with their profile / credential configurations to Commerce Cloud. You can see the configurations in Administration > Operations > Services.

[Administration](#) > [Operations](#) > [Services](#)

Services [Profiles](#) [Credentials](#)

Services

Select All	Name	Type	Profile	Credentials	Status
<input type="checkbox"/>	int_qas.address.get	SOAP	int_qas.soap	int_qas.soap	Live
<input type="checkbox"/>	int_qas.address.refine	SOAP	int_qas.soap	int_qas.soap	Live
<input type="checkbox"/>	int_qas.address.search	SOAP	int_qas.soap	int_qas.soap	Live
<input type="checkbox"/>	int_qas.email	HTTP	int_qas.email	int_qas.email	Live

[New](#) [Delete](#)

Note: that the Profile for int_qas.email (Administration > Operations > Services > Service Credentials > int_qas.email) contains the endpoint for the email verification service. It should be set to <https://api.experianmarketingservices.com/sync/queryresult/EmailValidate/1.0/>. The Endpoint for the Address verification service is set in the QAS On Demand Endpoint Site Preference (see above).

[Administration](#) > [Operations](#) > [Services](#) > [Service Credentials](#) > int_qas.email - Details

int_qas.email

Fields with a red asterisk (*) are mandatory. Click **Apply** to save the details. Click **Reset** to revert to the last saved state.

These credentials are used by 1 service.

Name: *	int_qas.email
URL:	https://api.experianmarketingservices.com/sync/query
User:	
Password:	*****

5.4 INTEGRATING WITH CONTROLLERS

Follow the instructions below in Commerce Cloud UX Studio in order to enable the corresponding feature. Your site must be built using JavaScript Controllers and you must be using the “int_QASc” cartridge provided in this distribution to make use of the following capabilities. If you are not using Controllers and your site still uses Pipelines, please reference section 3.5 of this document.

CORE INTEGRATION

The Controller based EDQ distribution handles validation through an AJAX interaction driven by a JavaScript module located at int_QASc/cartridge/js/qas.js. The steps below will add this module to be compiled with app.js through Gulp or Grunt. Once completed, you can enable and disable the use of certain features through commenting/uncommenting segments of the JavaScript module’s trigger conditions.

1. Copy **int_QASc/cartridge/js/qas.js** to your site’s **app_storefront_core/cartridge/js**
2. Modify **app_storefront_core/cartridge/js/app.js** and add the following 2 lines at the corresponding location.

```
app_storefront_core/cartridge/js/app.js

17 17     tooltip = require('./tooltip'),
18 18     util = require('./util'),
19 19     validator = require('./validator'),
20 20 +   qas = require('./qas'),
21 21     tls = require('./tls');
22 22
22 23     // if jQuery has not been loaded, load from google cdn

166 167     tooltip.init();
167 168     minicart.init();
168 169     validator.init();
169 170 +   qas.init();
170 171     rating.init();
171 172     searchplaceholder.init();
172 173     // execute page specific initializations
```

3. Modify **app_storefront_core/cartridge/js/pages/account.js** to add 2 new lines of code and comment out the apply button's on-click hook.

```

app_storefront_core/cartridge/js/pages/account.js

6   6       dialog = require('../dialog'),
7   7       page = require('../page'),
8   8       login = require('../login'),
9   9       - validator = require('../validator');
      + validator = require('../validator'),
      + qas = require('../qas');
10  10
11  11     /**
12  12     * @function

21  22         // $("input>").attr({type:"hidden", name:"format", value:"ajax"}).appendTo(form);
22  23
23  24         $form.on('click', '.apply-button', function (e) {
24  25         + /**
25  26             e.preventDefault();
26  27             if (!$form.valid()) {
27  28                 return false;

49  51         tooltip.init();
50  52     }
51  53     });
52  54     + */
53  55     })
54  56     .on('click', '.cancel-button, .close-button', function (e) {
55  57         e.preventDefault();

81  84     });
82  85
83  86     validator.init();
84  87     + qas.init();
85  88     }
86  89     /**
87  90     * @private

```

4. Modify **app_storefront_core/cartridge/scripts/util/Resource.ds** to add 3 new lines.

```

app_storefront_core/cartridge/scripts/util/Resource.ds

179 179     shippingMethodsJSON      : URLUtils.https('COSHipping-GetApplicableShippingMethodsJSON').toString(),
180 180     shippingMethodsList      : URLUtils.https('COSHipping-UpdateShippingMethodList').toString(),
181 181     selectShippingMethodsList : URLUtils.https('COSHipping-SelectShippingMethod').toString(),
182 182     +
183 183     + qasVerifyAddress         : URLUtils.https('QAS-VerifyAddress').toString(),
184 184     + qasUpdateAddress         : URLUtils.https('QAS-UpdateAddress').toString(),
185 185     + qasVerifyEmail          : URLUtils.https('QAS-VerifyEmail').toString(),
186 186     +
187 187     resetPaymentForms         : URLUtils.url('COBilling-ResetPaymentForms').toString(),
188 188     compareShow               : URLUtils.url('Compare-Show').toString(),
189 189     compareAdd                : URLUtils.url('Compare-AddProduct').toString(),

```

5. If your SiteGenesis controllers cartridge **IS NOT** named **"app_storefront_controllers"**, you must edit the **QAS.js** file located at **int_QASc/cartridge/controllers** and edit lines 10 and 11 to specify the correct name. If you are using the standard naming convention for the Site Genesis cartridges, you do not need to make any changes.
6. Use gulp or grunt, whichever you prefer, in order to compile **app_storefront_core/cartridge/static/default/js/app.js** and upload those changes to your Commerce Cloud Instance.

DISABLE EMAIL VALIDATION

To disable email validation, navigate to:

Merchant Tools > Site Preferences > Custom Preferences > QAS Config

Set the 'QAS Email Verification' value to 'Disabled (false)'

DISABLE ADDRESS VALIDATION

To disable address validation, navigate to:

Merchant Tools > Site Preferences > Custom Preferences > QAS Config

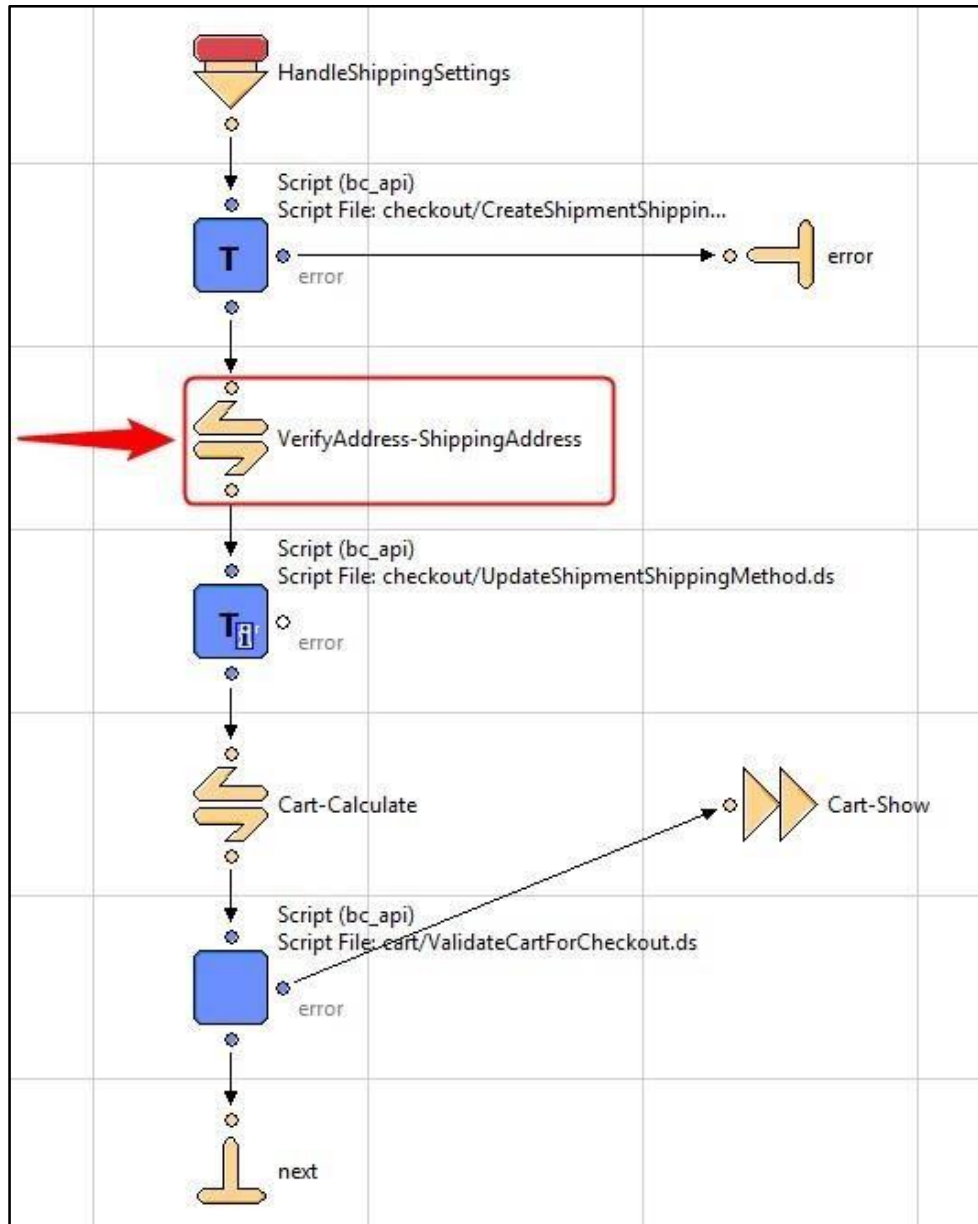
Set the 'QAS Address Verification' value to 'Disabled (false)'

5.5 INTEGRATING WITH PIPELINES

For sites that are still using Pipelines instead of JavaScript Controllers, please make the following changes. These steps can be performed within the Commerce Cloud UX Studio. Ensure that you are using the “int_QAS” cartridge provided in this distribution to integrate with Pipelines. The pipelines described below are not available in the

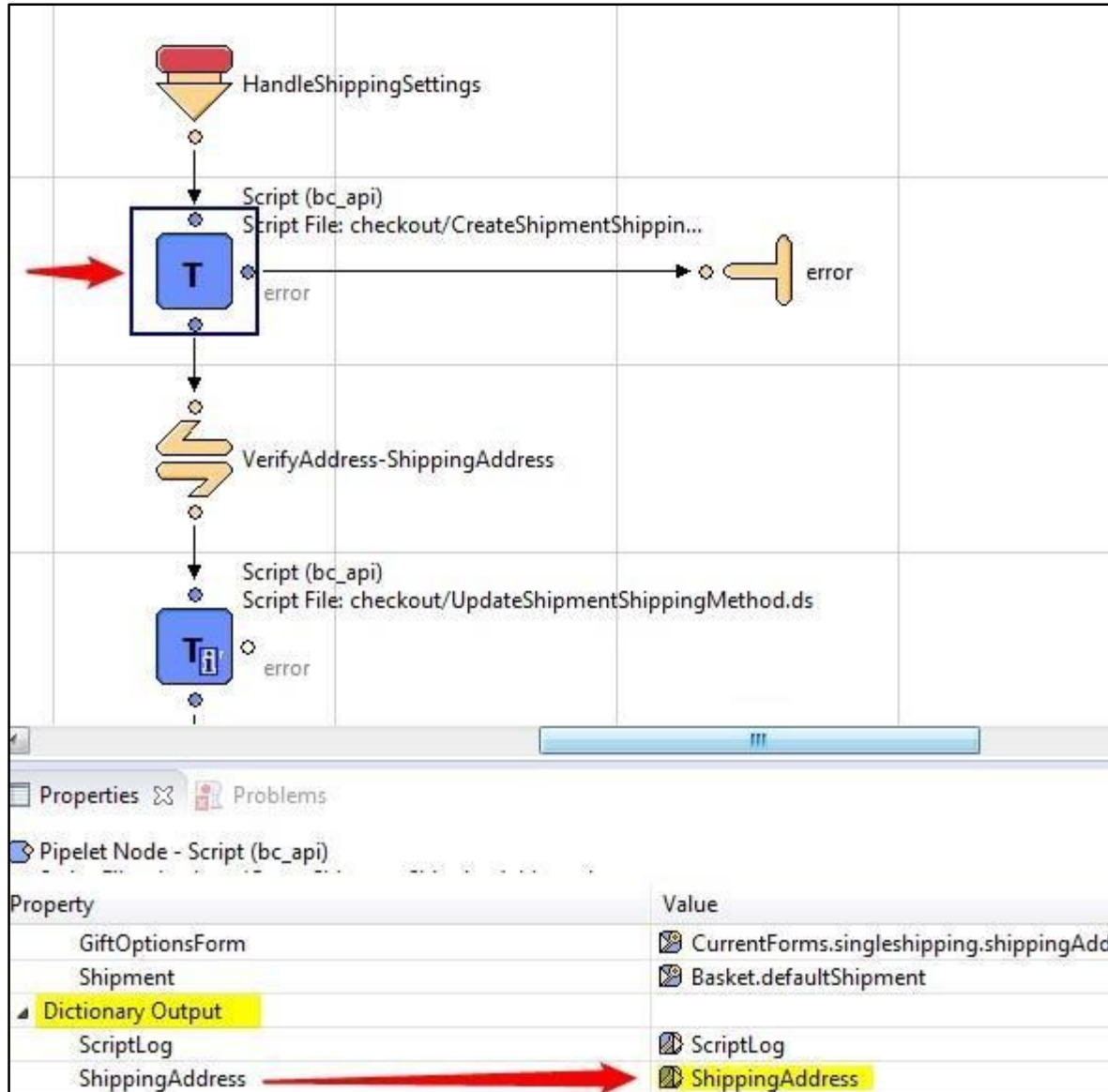
ADDRESS VALIDATION: SHIPPING ADDRESS

Add a call node to “VerifyAddress-ShippingAddress” in the COShipping-HandleShippingSettings pipeline.



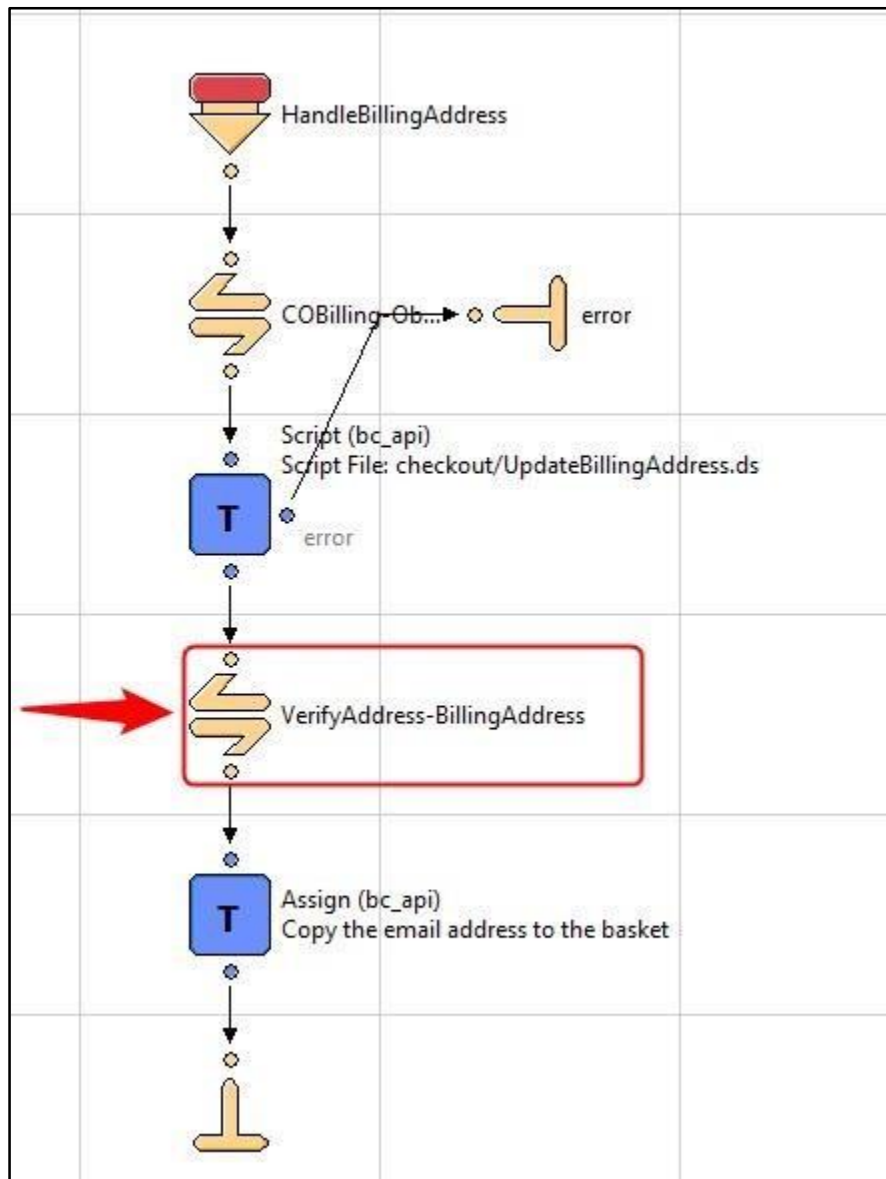
Controller based cartridge.

Ensure that a value of “ShippingAddress” is assigned to the Dictionary Output for ShippingAddress in the Properties of the script pipelette checkout/CreateShipmentShippingAddress.ds.



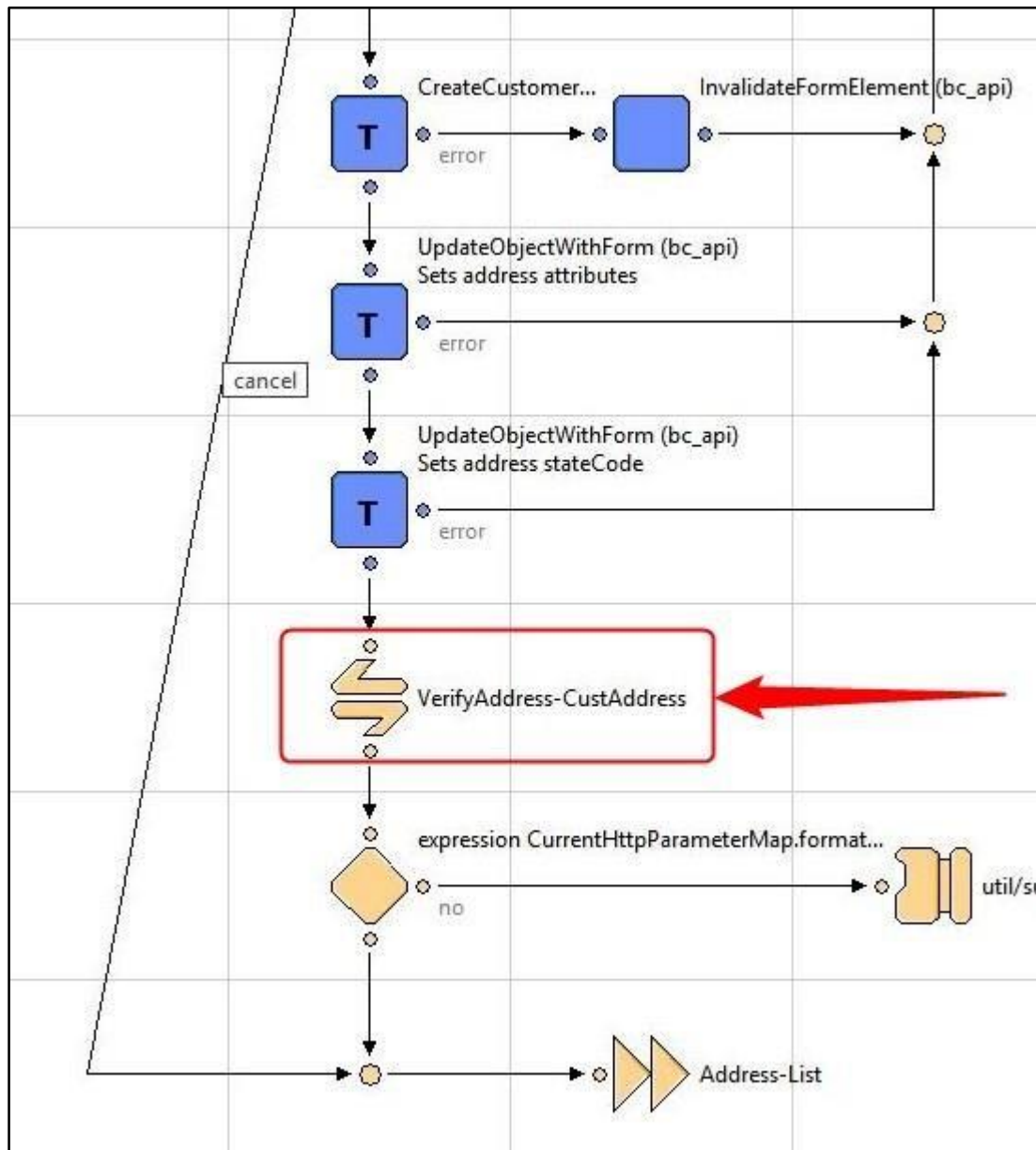
ADDRESS VALIDATION: BILLING ADDRESS

Add a call node to “VerifyAddress-BillingAddress” in the COBilling-HandleBillingAddress pipeline:

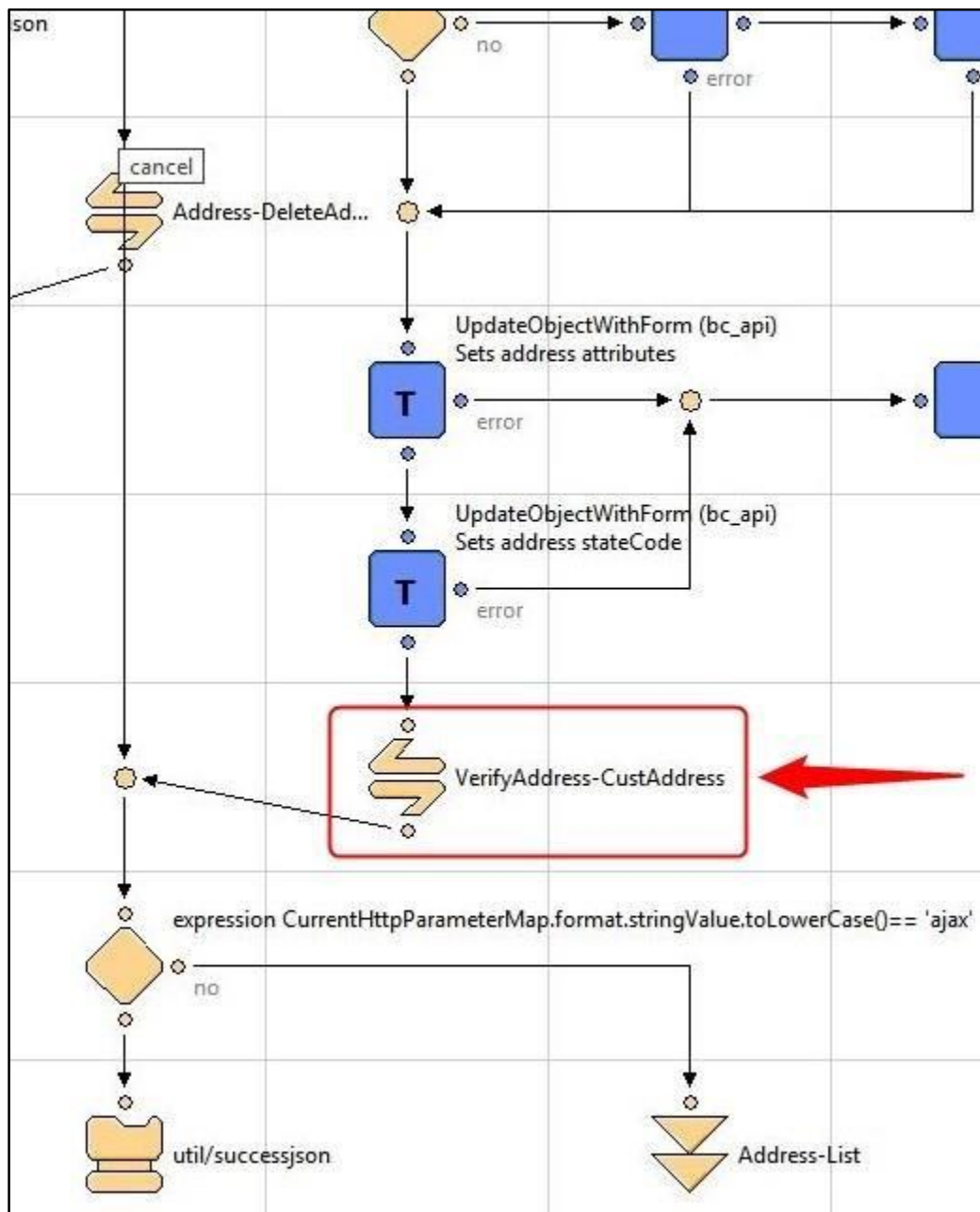


ADDRESS VALIDATION: ADDRESS BOOK

In “Address-Add”, add a Call Node to “VerifyAddress-CustAddress” between the existing nodes as shown below:

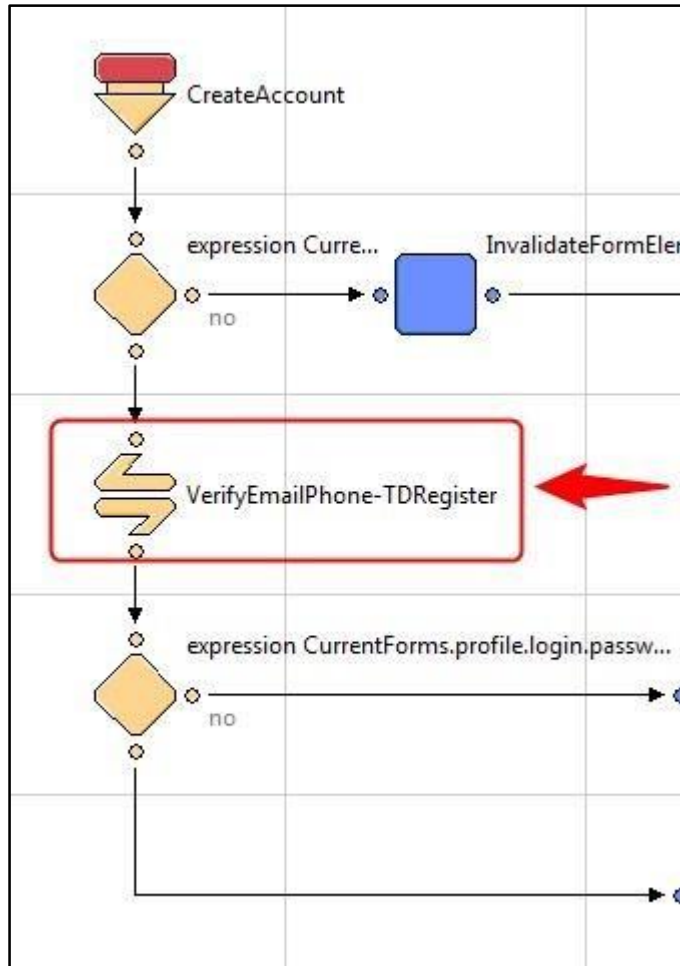


In “Address-Edit”, add a Call Node to “VerifyAddress-custAddress” between the existing nodes as shown below:



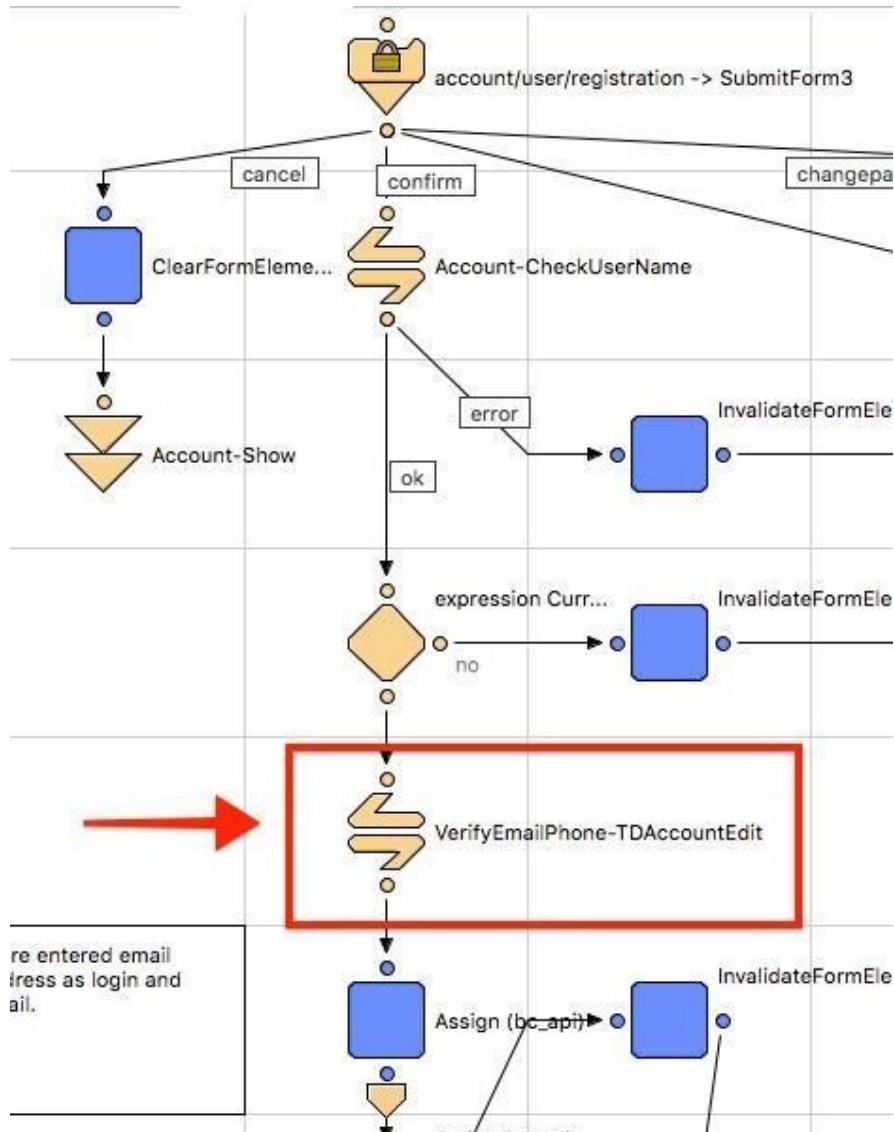
EMAIL VALIDATION: CREATE ACCOUNT

Add a call node to “VerifyEmailPhone-TDRegister” in the Account-CreateAccount pipeline:



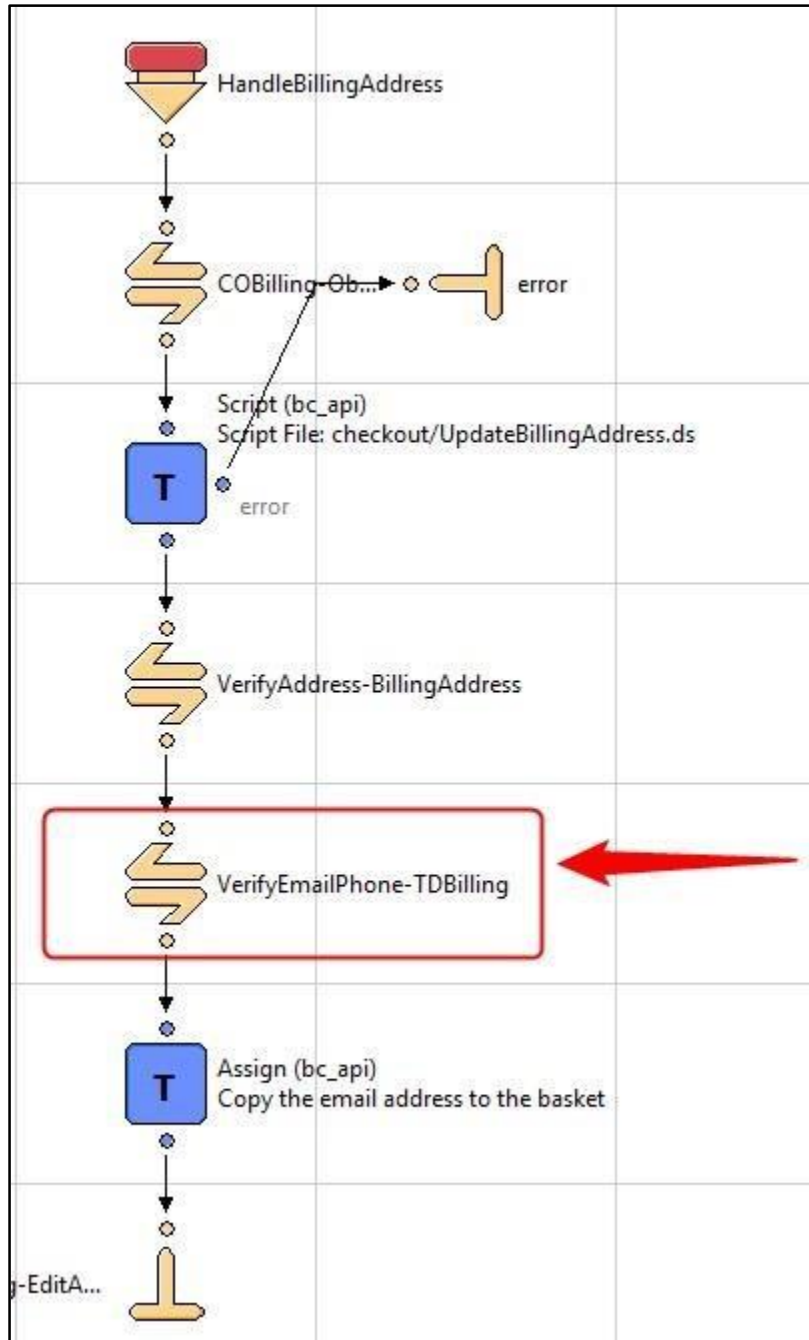
EMAIL VALIDATION: EDIT ACCOUNT

Add a call node to “VerifyEmailPhone-TDAccountEdit” in the Account-EditProfile pipeline:



EMAIL VALIDATION: BILLING ADDRESS

Add a call node to “VerifyEmailPhone-TDBilling” in the COBilling-HandleBillingAddress pipeline:



6. TESTING YOUR INTEGRATION

Once all steps above are complete, the workflow can be tested in the Commerce Cloud site with any touchpoint which was integrated into. In the examples below, we will test address validation within Guest Checkout – Shipping Address, and email validation within Create Account.

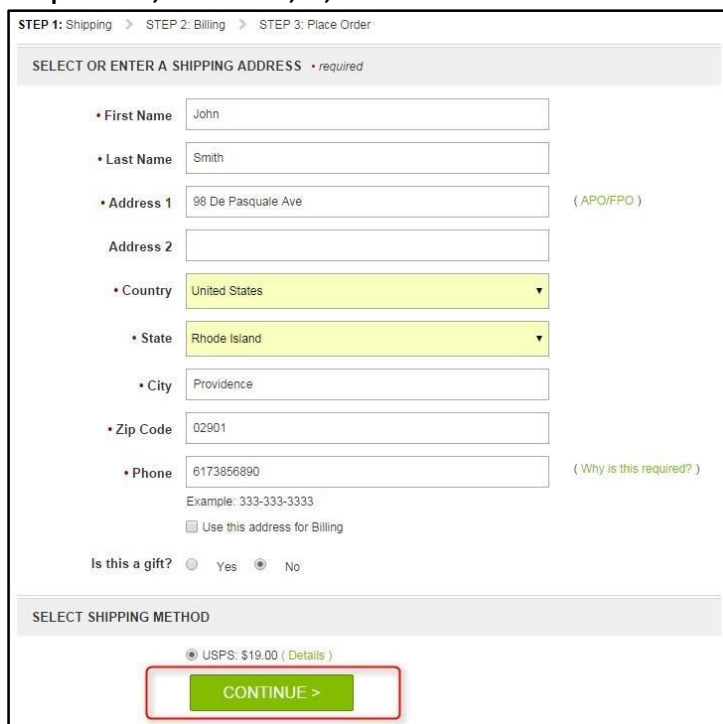
6.1 ADDRESS VALIDATION

EDQ for Commerce Cloud’s real time address validation can verify that a given address exists within the postal data from that country’s postal authority. Once a full, verified address is captured, the address can be formatted and standardized automatically, ensuring that captured data is both clean and accurate.

EDQ for Commerce Cloud’s real time address validation uses our best practices workflow. This workflow is intended for use within external, customer facing websites where customer interaction must be simple and intuitive.

The majority of input postal addresses will be validated automatically without any user interaction required. The remaining addresses will trigger one of five different user interactions that are specific to the issues with the input address. Lists of optional addresses are only shown when necessary and users can override the validation if they are unhappy with the suggested result. This allows the address to be corrected at point of capture with minimal effort by the user or customer.

In the following example, an address is entered on the checkout page’s Shipping Address touchpoint: **98 De Pasquale Ave, Providence, RI, 02901**:



STEP 1: Shipping > STEP 2: Billing > STEP 3: Place Order

SELECT OR ENTER A SHIPPING ADDRESS • required

• First Name John

• Last Name Smith

• Address 1 98 De Pasquale Ave (APO/FPO)

Address 2

• Country United States

• State Rhode Island

• City Providence

• Zip Code 02901

• Phone 6173856890 (Why is this required?)

Example: 333-333-3333

☐ Use this address for Billing

Is this a gift? ☐ Yes ☒ No

SELECT SHIPPING METHOD

☒ USPS: \$19.00 (Details)

CONTINUE >

Upon clicking Continue, EDQ Address Validation will be triggered. In this scenario, we will find that the input address belongs to an apartment building, and that the user did not enter any apartment level details with their address. This results in the following dialog:

Sorry, we think your apartment/suite/unit is missing or wrong.
To proceed, please enter your apartment/suite/unit or use your address as entered.

Confirm your Apartment/Suite/Unit number:

Apt:

[Submit](#)

You Entered: [\[Edit\]](#)

Address 1: 98 De Pasquale Ave

Address 2:

City: Providence

State: RI

Postal Code: 02901

[Use Original](#)

[Display all potential matches](#)

In this screen, the user has the option to update their information to include their apartment details, or, they can accept their original address as it was entered initially. When the correct apartment number is captured, the service returns the complete and accurate address including standardization and the ZIP/+4 code.

6.2 EMAIL VALIDATION

EDQ for Commerce Cloud's real time email validation can verify that a given email exists and can currently accept mail. Additionally, our correction logic can catch common typing mistakes and recommend suggestions.

Similar to real time address validation, EDQ for Commerce Cloud's real time email validation also uses our best practices workflow. This workflow is intended for use within external, customer facing websites where customer interaction must be clear and simple.

Many times, email addresses are entered accurately on the first try. The remaining addresses will trigger different user interactions that are specific to the issues with the email. Suggested corrections are only shown when necessary and users can override the validation if they are unhappy with the suggested result. This allows the information to be corrected at point of capture with minimal effort.

In the following example, an email is entered on the Create Account touchpoint: **john.smith987@gmail.com**

Create Account

NAME • required

• First Name John

• Last Name Smith

EMAIL / LOGIN INFORMATION • required

• Email john.smith987@gmail.com

• Confirm Email john.smith987@gmail.com

• Password

8 - 255 characters

• Confirm Password

☐ Please add me to the Demandware's email list.

Demandware does not share or sell personal info. [See Privacy Policy](#)

Apply

Upon clicking Apply, EDQ Email Validation will be triggered. In this scenario, we will find that the input email domain of gmail.com is not valid, and that the user most likely intended to enter gmail.com. This results in the following dialog:

Sorry we could not confirm your e-mail address
To proceed, please confirm your e-mail address below.

Confirm or edit your e-mail address

john.smith987@gmail.com

Did you mean?

john.smith987@gmail.com

Update Accept Original

Note that the exact error message is hidden from the user, but is visible in the page source for debugging purposes:

```

<div class="enter_email">
  <p>
    <strong>Confirm or edit your e-mail address</strong>
  </p>
  <p class="error_message unreachable">Domain has no reachable mail exchangers.</p>
  <input id="newEmail" type="text" name="Email" value="john.smith987@gmail.com">
  <p>Did you mean?</p>

```

7. EDQ MAINTENANCE INFORMATION

The EDQ for Commerce Cloud integration utilizes Experian Data Quality's SaaS-based address, email, and phone services. These services are monitored closely at all times. As these services are hosted, maintenance such as bug fixes and data updates are handled by Experian Data Quality.

To access these services, an application key will be provided from Experian Data Quality during the integration. This application key will remain the same throughout licensing cycles.

For information related to SLAs of the EDQ hosted services, please refer to our support site below:

http://support.qas.com/qas_pro_on_demand_sla_2079.htm

8. RELEASE NOTES

18.1.0:

- Corrected a bug where the user could not submit a recommended address during the 'Interaction Required' workflow for address validation
- Fixed an issue where checkout could be blocked if a timeout occurred with the EDQ service during address validation
- Added the option to enable/disable cartridge functionality for address and email through SitePreferences

9. EDQ PRODUCT APPENDIX

9.1 ADDRESS VALIDATION MATCH TYPES AND EXAMPLES

EDQ Address Validation can result in six potential outcomes, described below. These outcomes correspond to the quality of validation which occurred with the input address.

Verification Level	Description	Address Example
Verified	Address matched to a high confidence; can include changes such as corrected spelling, added street descriptors or ZIP/+4 codes.	1746 Bay St San Francisco CA 94123
Interaction Required	An address was found, but changes were made requiring confirmation by the user.	19 Cook Portland Or 97212
Premises Partial	The address was found to the premise level but secondary info was wrong or missing. The user is prompted for this information.	1889 Broadway San Francisco Ca 94109
Street Partial	The address was found to the street level, but building number was wrong or missing. The user is prompted for this information.	100 Eliot Rd Arlington MA 02474
Multiple	Multiple addresses were matched to. The user is prompted to select the correct address.	4440 Image Dallas TX
None	No address match was found. The user is prompted to reenter their address	1 Timbuktu Rd Portland OR 97000

9.2 EMAIL VALIDATION MATCH TYPES AND EXAMPLES

EDQ Email Validation can result in six potential outcomes, described below. These outcomes correspond to the quality of validation which occurred with the input email.

Certainty	Description	Example
Verified	Mailbox exists, is reachable, and not known to be illegitimate or disposable.	jeston@gmail.com
Undeliverable	Mailbox or domain does not exist; mailbox is full, suspended or disabled.	nottaken@aol.com
Unreachable	Domain has no reachable mail exchangers.	queen@parliament.uk
Illegitimate	Seed, spamtrap, black hole, technical role account or inactive domain.	postmaster@cnn.com
Disposable	Domain is administered by a disposable email provider (e.g. Mailinator).	john5@mailinator.com
Unknown	We were unable to conclusively verify or invalidate this address. This most is most common when the domain is an Accept all or the request times out.	mark@gas.com

*Note that emails can change certainty levels at any time. Please contact EDQ Support if you have any questions.