

# UPS Address Validation in Syteline

---



Customization & Documentation by Ryan Dumais

Revision A – 2018-3-14

# Table of Contents

Introduction .....	3
Purpose .....	3
What is the Customization? .....	3
How Does the Customization Work? .....	3
Limitations .....	4
Installation .....	4
Requirements .....	4
Program Components .....	4
IDOs .....	4
IDO Projects .....	4
Tables & Views .....	4
Global Scripts .....	4
Forms .....	4
Installation Procedure .....	5
Implementation Procedure .....	6
Event Handler Parameters Explained .....	6
Rollback Procedure .....	7
Using the Customization .....	7
LSC_AddressValidationParameters form .....	7
Validation Prompt Examples .....	8
Valid Address Prompt .....	8
Invalid Address Prompt .....	10
Ambiguous Address Prompt .....	10
Resources .....	12
Contact the Author .....	12

# Introduction

## Purpose

Linemaster's purpose for this customization was to reduce shipping errors that occur before product shipment. We often ran into problems where the customer's address was never validated which then caused late shipments and frustration. Linemaster decided it would be best to find problems with addresses and validate them immediately before they become an issue at the end of the road.

## What is the Customization?

The customization gives you the ability to validate addresses in Syteline through UPS's web service. It consists of a global script that is designed to be modular and can be used just about any form that has an address. By using the customization you will be able to know if the address is valid, invalid, or ambiguous. The user is also given the option to format the address to UPS standards.

It is important to know that validating addresses with UPS does not increase the speed of delivery. However, having a valid address will reduce errors which could prevent shipment delays.

## How Does the Customization Work?

The core of the customization utilizes UPS's API for street level validation. By using the Mongoose Framework we are able to use VB.NET as global scripts that make HTTP requests to UPS's API. The global script handles the request and the response from UPS to validate parameters that have been passed in by the Syteline form.

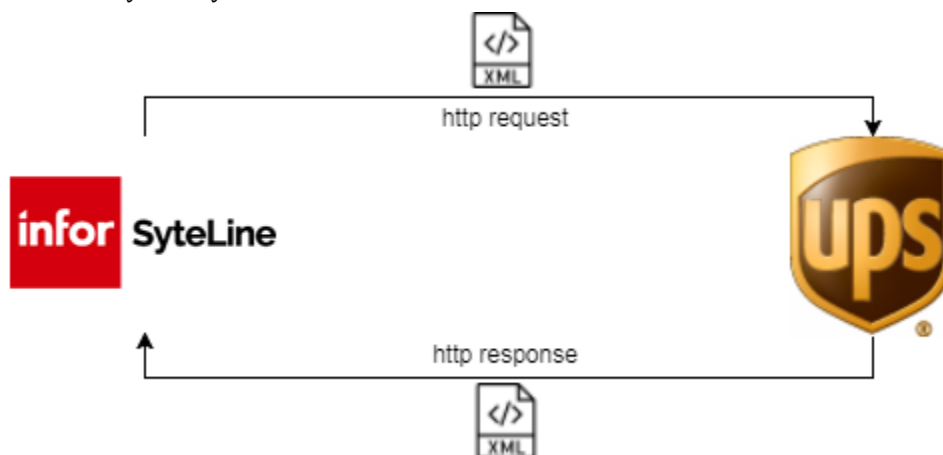


Figure 1 - Diagram for http request and response

## Limitations

Unfortunately UPS has some limitations with its API:

- UPS only validates addresses located in the United States and Puerto Rico.
- UPS does not validate suites/apartments/floors/etc.

## Installation

### Requirements

- On-premise Syteline v9.01.01 or higher
- Microsoft SQL Server 2012 or higher
- Backup of forms being modified
- [UPS Developer kit access](#)

### Program Components

#### IDOs

- LSC\_UPSAddrValidations

#### IDO Projects

- LSC

#### Tables & Views

- LSC\_UPSAddrValidation\_mst (table)
- LSC\_UPSAddrValidation (view)

#### Global Scripts

- LSC\_UPSAddressValidation

#### Forms

- LSC\_AddressValidationParameters

## Installation Procedure

Before starting installation have your UPS developer's kit username, password, and API key ready. You can view instructions in the link provided in the 'Requirements' section of this documentation. It is crucial that this procedure is followed in order.

1. Close any instances of Syteline.
2. In the SQL Scripts folder, run 'TABLE\_LSC\_UPSAddrValidation\_mst.sql' then 'VIEW\_LSC\_UPSAddrValidation.sql' against your application database.
3. Open 'INSERT\_LSC\_UPSAddrValidation\_mst.sql' in a text editor. Replace 'USERNAMEHERE', 'PASSWORDHERE', and 'APIKEYHERE' with your actual credentials.
4. Run the edited 'INSERT\_LSC\_UPSAddrValidation\_mst.sql' against your application database.
5. On your utility server run 'App Metadata Transport' and import 'LSC\_UPSAddrValidation.xml' from your IDOs folder. Select the configuration and only import selected IDOs. There are no replication documents to import. Finish the wizard. You now have the IDO and the IDO's project in your Syteline configuration.
6. In the Forms folder, run 'FGO\_LSC\_AddressValidationParameters.sql' against your forms database.
7. Run 'LIC\_LSC\_AddressValidationParameters.sql' against your application database. You now have the required forms in Syteline.
8. Open Syteline and unload global objects (Ctrl+U)
9. Enter design mode (Ctrl+E) and navigate Edit -> Script. Click the 'New' button, name your script 'LSC\_UPSAddressValidation' with Visual Basic as the language. Copy the contents from 'LSC\_UPSAddressValidation.vb' in your Global Script folder and replace all the text in the Script editor. Click OK > Done.
10. Sign out of Syteline, save the script modifications.

## Implementation Procedure

Now that we have the customization in Syteline we are now able add this customization on forms that have address fields. I recommend starting with 'Customer Ship-Tos' or 'Customers' form.

1. Enter design mode on the desired form.
2. Add a button onto the form. Name and style the button.
3. Add an event handler to the button with the following settings:
  - a. Type: Run Script
  - b. Parameters: Add all text from the EventParameters.txt file in the Event Handlers folder.
4. Click OK on the Event Handler Properties and Event Handlers windows.
5. Regenerate the form by pressing Ctrl+G.
6. Close the form and reopen.
7. Test!

## Event Handler Parameters Explained

The global script requires 15 parameters to be passed in. The text in the EventParameters.txt file will work on most vendor version forms, but if any component names differ than you will have to make adjustments. It is important to know that the parameter list is zero-based. Below is a table on the parameters:

Table 1 - Parameters for the global script

Parameter #	Default Value	Value Meaning
0	C(Addr1_Edit)	Component value for the first address line
1	C(Addr2_Edit)	Component value for the second address line
2	C(Addr3_Edit)	Component value for the third address line
3	C(Addr4_Edit)	Component value for the fourth address line
4	C(CityEdit)	Component value for the city
5	C(StateEdit)	Component value for the state
6	C(ZipEdit)	Component value for the zip code
7	C(CountryEdit)	Component value for the country
8	Addr_1Edit	Component name for the first address line
9	Addr_2Edit	Component name for the second address line
10	Addr_3Edit	Component name for the third address line
11	Addr_4Edit	Component name for the fourth address line
12	CityEdit	Component name for the city field
13	StateEdit	Component name for the state field
14	ZipEdit	Component name for the zip code field

## Rollback Procedure

Follow this procedure to remove the customization from your Syteline environment.

1. In Syteline, open the form 'LSC\_AddressValidationParameters' in design mode. In the menu bar, Form > Definitions > Delete...
2. Open the 'IDOs' form and check out the 'LSC\_UPSAddrValidations' IDO. Delete the IDO and save.
3. Open the 'IDO Projects' form, delete the 'LSC' project and save.
4. Close Syteline.
5. Delete the 'LSC\_UPSAddrValidation\_mst' table and 'LSC\_UPSAddrValidation' view from your application database.
6. Run the production form backups you have taken of the form you chose to customize against your forms database.
7. Open Syteline and unload global objects (Ctrl+U)

## Using the Customization

It is ultimately up to you where the customization is used. Once the customization is added to a form you simply click the validate button when there is an address record available on the form. You will receive a prompt indicating the validity of the address. If formatting is available you will be prompted the option to. See the 'Validation Prompt Examples' section for more details on the prompts.

### **LSC\_AddressValidationParameters form**

This form is added from the customization. The purpose of this form is to allow you to change your UPS developer account's username, password, and API key as needed. Once you save the record it will be stored in your table. After installation you will be able to see the username, password, and API key that you set in 'INSERT\_LSC\_UPSAddrValidation\_mst.sql'.

This form is required by the global script as it extracts information from this form to send the request to UPS. The form has two additional fields named 'XMLTemplate Header' and 'XMLTemplate Body' which holds the barebones template for the UPS request. The global script populates the template's XML tags with the parameter values on the form we are running the script from before sending to UPS.



The form contains three input fields on the left: Username (placeholder: USERNAMEHERE), Password (placeholder: PASSWORDHERE), and API Key (placeholder: APIKEYHERE). On the right, there are two XML template editors. The 'XMLTemplate Header' editor contains the following XML:

```
<?xml version="1.0" ?>
<AccessRequest xml:lang='en-US'>
  <AccessLicenseNumber></AccessLicenseNumber>
  <UserId></UserId>
  <Password></Password>
</AccessRequest>
```

The 'XMLTemplate Body' editor contains the following XML:

```
<?xml version="1.0" ?>
<AddressValidationRequest xml:lang='en-US'>
  <Request>
    <RequestAction>XAV</RequestAction>
    <RequestOption>1</RequestOption>
  </Request>
  <AddressKeyFormat>
    <AddressLine></AddressLine>
    <Region></Region>
    <PoliticalDivision2></PoliticalDivision2>
    <PoliticalDivision1></PoliticalDivision1>
    <PostcodePrimaryLow></PostcodePrimaryLow>
    <CountryCode></CountryCode>
  </AddressKeyFormat>
</AddressValidationRequest>
```

Figure 2 - LSC\_AddressValidationParameters form

## Validation Prompt Examples

### Valid Address Prompt

There are two types of valid address prompts: one with the option to format the text to UPS standards and other is a prompt to let you know the address is valid. You will only be given the option to format the address if the component fields are enabled or not read-only.

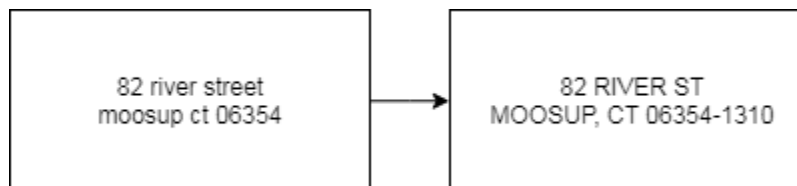


Figure 3 - Example of formatting provided by UPS



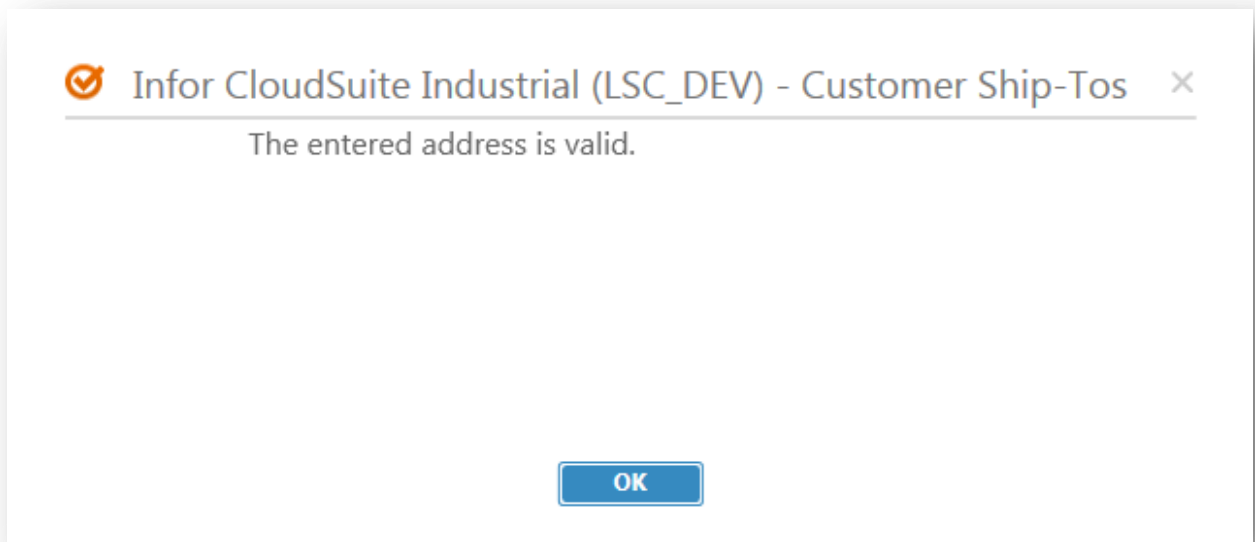


Figure 4 - Valid address without format prompt

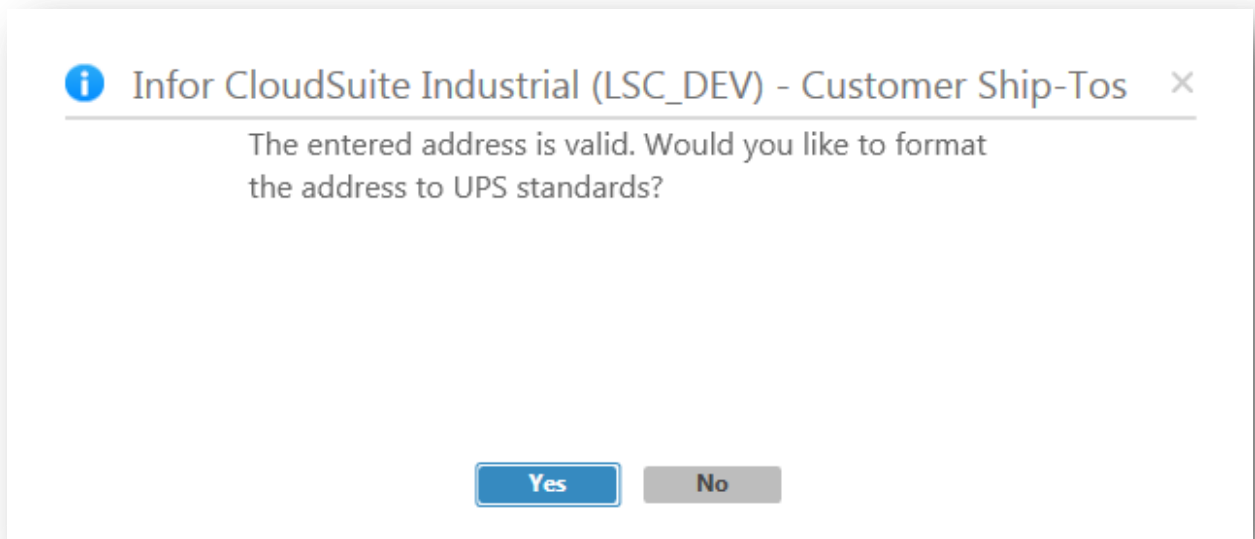


Figure 5 - Valid address with format prompt

### Invalid Address Prompt

If UPS is unable to validate an address you will receive a prompt. Addresses will not be formatted in invalid responses.

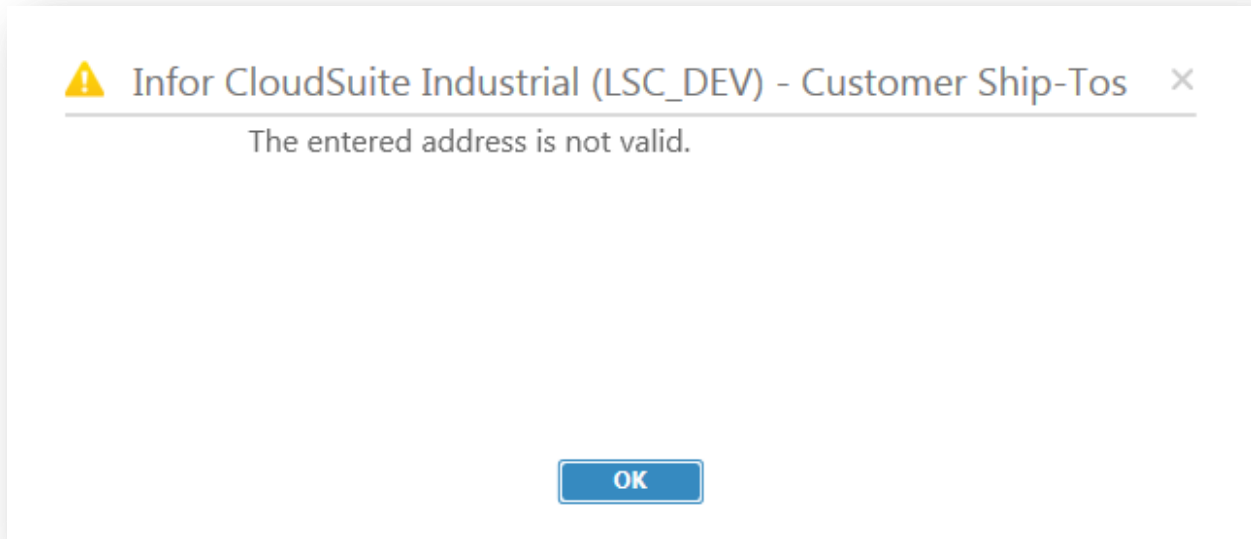


Figure 6 - Invalid address prompt

### Ambiguous Address Prompt

If the address is ambiguous you will receive a prompt. This happens when the address may have been worded differently than what UPS was expecting. Although the address may be deliverable, it does not mean that the address is validated through UPS. The prompt will let you know which fields were not consistent with UPS' database.

When this message appears you don't necessarily have to do anything with it. All it is trying to tell you is that the following fields (in this case, the address line and Zip Code) are the only part of the address that came back questionable, according to UPS. The address will not be formatted in ambiguous responses.

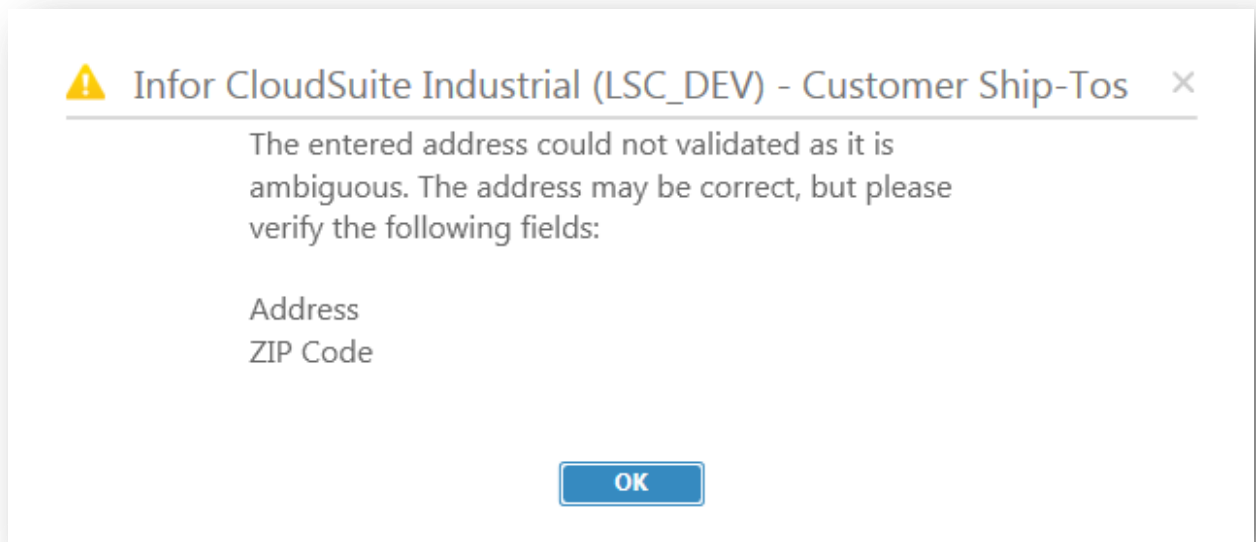


Figure 7 - Ambiguous address prompt

## Resources

[UPS Standard Address Guideline](#) – Useful for knowing the specification of the UPS standard on addresses.

### Contact the Author

<b>Name</b>	Ryan Dumais
<b>Email</b>	<a href="mailto:RDumais@linemaster.com">RDumais@linemaster.com</a>
<b>Portfolio Website</b>	<a href="https://ryanduma.is">https://ryanduma.is</a>
<b>GitHub</b>	<a href="https://github.com/RDumais">https://github.com/RDumais</a>