

## Technical Best Practices for Summer EBT Mailing Address Validation

### Introduction

Summer EBT (S-EBT) provides food assistance over the summer to eligible children via the use of an electronic benefit transfer (EBT) card. S-EBT benefits are time-limited—they expire four months after issuance—so timely and accurate delivery of S-EBT cards is crucial to ensure that eligible families are able to use their benefits during the summer. However, incomplete or inaccurate mailing address data often hinders cards from being delivered. This is frustrating for families who need benefits, and it is also expensive and inefficient for S-EBT administrators who have to resend or reissue EBT cards within a tight time window.

Some states co-load S-EBT benefits onto existing EBT cards, which can simplify benefits delivery to streamlined certified families who automatically qualify for benefits (e.g., those who are already enrolled in the Supplemental Nutrition Assistance Program (SNAP)). However, all states that are administering S-EBT need to issue and send new EBT cards specifically for S-EBT to at least some portion of approved participants. **Implementing strategies for successful card delivery is therefore essential to contain program costs and increase program effectiveness.**

This document provides technical guidance to S-EBT administrators on how to implement **address validation technology** to improve S-EBT card deliverability, including detailed discussion of APIs and code libraries. It also acts as a companion to the accompanying technical resources that are freely available and usable by any S-EBT agency: a [demo application](#) of address validation in action and [sample code](#).

Some key questions for S-EBT administrators to consider are:

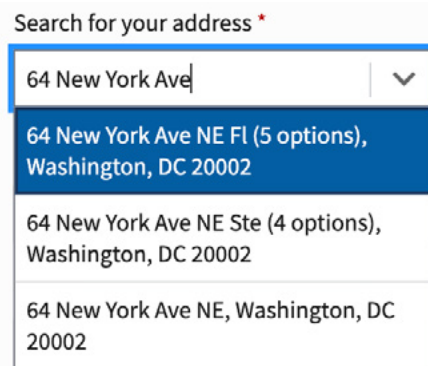
- 1. When collecting a new or updated address, how can you ensure it is valid and deliverable by the USPS?** When participants provide a mailing address on their S-EBT Application form via a call with a customer service agent or update their address on file via a [self-service portal](#), states have an important opportunity to ensure address data quality. However, human data entry mistakes are easy to make when entering an address in a form. An interactive web form using address validation technology provides a simple way to catch and correct errors when they are detected.

- 2. How can you maximize the likelihood of having a correct and current address for the person who should receive a S-EBT card?** People may move often and don't always update their address with government agencies. When determining eligibility via streamlined certification, it is possible that addresses sourced from SNAP, TANF, FDPIR, and/or Medicaid program data may be outdated, incomplete, and/or incorrect. Bulk validation of addresses sourced from these systems can be used to proactively determine when outreach to the client for an updated address is necessary.

### Step 1: Improve address quality at the point of data entry

When collecting an address from a client in a web form (i.e. the S-EBT application or in a self-service portal) or when agency staff are updating a case file, we recommend using an [autocomplete/typeahead](#) user interface element. As people type their address into the combo box, a connected web service provides suggested addresses as they type, which are displayed for selection as shown in Figure 1. This type of user interface allows the client to choose from a list of suggested addresses that are known to be valid, reducing the likelihood of errors introduced by typos or other data entry mistakes, such as an incorrect ZIP code or a missing apartment/unit number.

**Figure 1:** Autocomplete suggestions are displayed in response to entered text



The image shows a web form with a search bar labeled "Search for your address \*". The search bar contains the text "64 New York Ave". Below the search bar, a dropdown menu is open, displaying three suggestions:

- 64 New York Ave NE Fl (5 options), Washington, DC 20002
- 64 New York Ave NE Ste (4 options), Washington, DC 20002
- 64 New York Ave NE, Washington, DC 20002

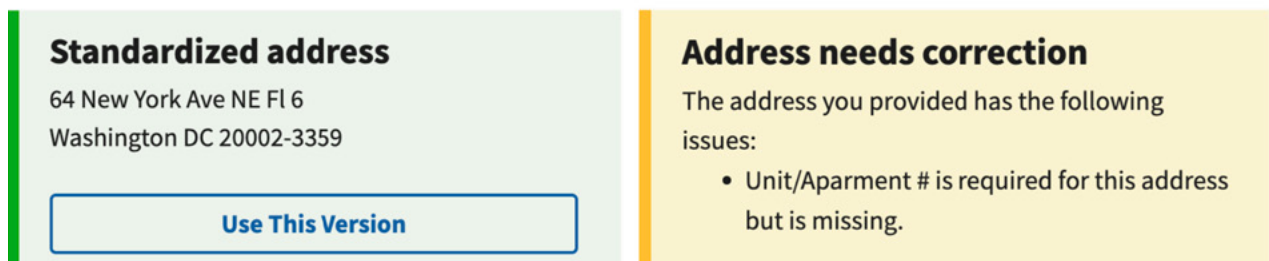
As a best practice, the points of manual address entry where an [autocomplete/typeahead](#) user interface element should be implemented are:

- The S-EBT online application
- Any forms or case management systems being used by customer service agents or agency staff to:
  - Take client address information over the phone and enter it into a data system
  - Digitize client address information based on a paper S-EBT application
- Any self-service customer support tool where a S-EBT client can independently enter or correct their address information (this might be a self-service portal, via a chatbot or live chat tool)

## Step 2: Confirm address deliverability with USPS

Whether a person selects an address from an autocomplete suggestion or enters it manually, the address should be checked against the USPS [delivery point](#) database for validity. The user interface should provide feedback on the validation to the user as seen in Figure 2.

**Figure 2:** Alerts displaying a valid, standardized address and prompting a user for a correction



Various APIs exist for this purpose, including the official, free-of-charge [USPS Address Validation API](#), as well as several commercial options, such as those from [Smarty](#) and [Google Maps](#), which both include [CASS™-certified](#) USPS validation capabilities. When selecting a commercial address validation service, it is crucial to choose one that is CASS™-certified and provides USPS validation results. If you plan to retain detailed validation data in your databases, be sure to review the service’s terms of use to ensure there are no restrictions on data retention.

Key data points from the USPS validation result are the **DPV Confirmation** and **DPV Footnotes**. The confirmation code provides a top-level indicator of validity and will be either “Y”, “D”, “S”, “N”, or blank:

- **Y:** Yes, this address is valid and recognized by the USPS, including apartment or unit numbers if applicable. To confirm deliverability, ensure that the vacancy indicator is **N** and the **DPV Footnotes** do not contain **R7** (address matched to a physical address that does not receive delivery from the USPS). You may also consider checking the no-stat indicator (if available) to ensure a value of **N**, but this field is infrequently updated by the USPS, and relying on it could result in withholding mail from an address that is actually deliverable.
- **D:** The street address is valid according to the USPS, but it’s missing an apartment or unit number. The person should be prompted to enter that information.
- **S:** The street address is valid, but the apartment or unit number that was entered is not valid or unnecessary. The person should be asked to correct it.
- **N or blank:** The USPS could not confirm the address in the delivery point database.

**DPV Footnotes** and **Footnotes** provide additional, more detailed information about the deliverability of the address, which can be used to guide mailing decisions and/or prompt the client for more/corrected information. See Section 2.3 in the [USPS documentation](#) or the **dpv\_footnotes** and **footnotes** fields in

the [Smarty documentation](#) for more detail. Sometimes additional fields may be useful for ensuring deliverability, such as indicators of whether the address is vacant or if it is classified as residential or commercial. See the sample code for our demo application for parsing the [DPV footnotes](#) and [footnotes](#) and transforming them into [convenient descriptions](#) for the user interface.

### Step 3: Validate database addresses in bulk

When issuing benefits on new EBT cards to people whose eligibility was determined via streamlined certification, it is important to check the validity of the addresses in the streamlined certification dataset. The quality of these addresses may vary widely depending on their provenance and age.

**Offline pre-processing:** We recommend pre-processing addresses offline before performing online validation with a web service to avoid API usage for addresses that could never be resolved due to severe data quality problems (e.g., lacking a street number entirely). This will also prevent wasteful API usage costs if you are using a commercial validation API.

How to do it: One useful tool to apply for both parsing/normalizing and fuzzy matching tasks is the open-source [libpostal](#) library. This library is written in C but has bindings for Python and several other popular programming languages, and it is relatively easy to create bindings for other languages.

**Bulk validation:** After completing offline pre-processing steps, the web services mentioned above can all be used for validating large datasets. Some of the commercial offerings, such as Smarty, even include [bulk validation tools](#) for speed, ease of use, and improved performance. When selecting a commercial bulk validation option, be sure to review the terms of use. Some commercial address validation services have terms that significantly restrict data retention when used in backend processing where no client interaction occurs. The points discussed above also apply to bulk address validation, but the actions required in case of validation issues differ:

- **Okay to mail:** With a DPV Confirmation of **Y**, it is safe to proceed with mailing, even if minor changes were made to the address for consistent formatting or correcting a zip code.
- **Not okay to mail:** When a DPV Confirmation of **N or blank** is returned, outreach to the client via phone, email, or [text message](#) to request an updated address is necessary in order to ensure successful delivery.
- **Maybe okay to mail:** When a DPV Confirmation of **D or S** is returned, outreach to the client is preferable. However, it may be possible to proceed with mailing if you know the number of apartments/units at the address is low and feel confident that a mail carrier can route the mailpiece to the correct person by name. The USPS does not provide data indicating the number of units in a building, but it is available in open data sets for some locations (e.g., the [Address Residential Units](#) data set for Washington, D.C., made available by [Open Data DC](#)), as well as via some commercial address validation products like [Smarty](#).

## Technical Resources for S-EBT Agencies

[Sample address web form using Smarty for autocomplete and validation](#) A sample application, demonstrating usage of the Smarty JavaScript SDK for autocomplete and address validation.

[USPS Address Validation API Documentation](#) Official documentation from the USPS for the Address Validation API. This resource provides detailed descriptions of request and response data, including definitions for various codes like DPV Confirmation and DPV Footnotes.

Smarty [Address Validation](#) and [Autocomplete](#) APIs Official documentation for the Smarty Address Validation and Autocomplete APIs, including detailed description of request and response data.

Google [Address Validation](#) and [Autocomplete](#) APIs Official documentation for the Google Maps Address Validation and Autocomplete APIs, including detailed description of request and response data.

[Libpostal open-source library](#) The GitHub repository for the libpostal open-source code library useful for the pre-processing step of bulk address validation. The repository's README contains detailed instructions for installation, sample code in C and Python, and links to official and unofficial bindings in other programming languages.