

Address Validation Documentation

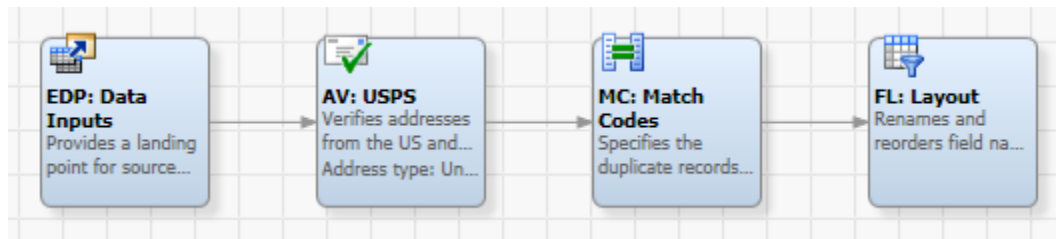
Introduction

The “Address Validation” job ... [boilerplate goes here](#) [ssu test edit]

The current scope of the pilot focuses solely on domestic addresses, so it is worth keeping in mind that additional work is required to bring this to full production mode with validation/standardization options for both domestic and international addresses.

```
Name: addressvalidation.ddf
WSDL: http://54.244.240.85:21036/datasvc/AddressValidation.ddf?wsdl
```

High-Level Overview



The DataFlux job has four nodes flowing from left to right.

External Data Input Node

This node provides a landing point for source data that is external to the current job. It receives raw address data from the CommIT Registration web form via a SOAP WS. The current attributes include:

- address1
- address2
- address3
- city
- state
- postal_code

Future iterations should probably include county, perhaps as iso_country_code, but for the scope of this pilot all addresses are assumed to be from the US.

Address Validation

This node verifies addresses from the US via the USPS Database. It maps the raw address data into fields which are consumed by the USPS address validation process. The outputs of this node include what is commonly referred to as the “cleansed” address, as well as the “original” address from the web form. This node also includes USPS metadata, of which we are utilizing the USPS Result Code and USPS Numeric Result Code.

At this stage, the cleansed address will include:

- Some typical standardization (e.g., “Street” becomes “St”; “Avenue” becomes “Ave”, etc.)
- 2 Address Lines (i.e., the job takes in 3 Address Lines and converts the output to 2)
- Zip Code, if address is found in the USPS Database, will be in 99999-9999 format.
- The “US Preferred City Name,” which in some cases may be different from what the user originally entered.
- The “US Result Code” and “US Numeric Result Code,” which simply indicate whether the address was found or not. (This is a very simple validation, for now).

Text Result Code	Numeric Result Code	Description
OK	0	Address was verified successfully.
PARSE	11	Error parsing address. Components of the address might be missing.
CITY	12	Could not locate city/state or zip in the USPS database. At least (city and state) or ZIP must be present in the input.
MULTI	13	Ambiguous address. There were two or more possible matches for this address with different data.
NOMATCH	14	No matching address found in the USPS data.
OVER	15	One or more input strings is too long (maximum 100 characters).

Future iterations should consider deeper levels of address validation, such as Delivery Point Validation (“DPV”) and use of other nodes to handle non-US addresses.

Match Codes

This node generates proprietary match codes for both the original and cleansed versions of "Address1" and "City." These are the match codes which the match logic will check for person/identity matches.

- "address1" uses the DataFlux Address (v22) Definition at 85 Sensitivity.
- "city" uses the DataFlux City Definition at 85 Sensitivity.

Field Layout

Reorganizes the various data fields into a practical layout for the output. For clarity, the original data fields are returned with an `o_` prefix (e.g., `o_city`) and the cleansed data fields are returned with a `c_` prefix (e.g., `c_city`).

The full list of returned attributes include:

- `o_address1`
- `o_address2`
- `o_address3`
- `o_city`
- `o_state`
- `o_postal_code`
- `o_address_matchcode` (44 characters)
- `o_city_matchcode` (15 characters)
- `c_address1`
- `c_address2`
- `c_city`
- `c_state`
- `c_postal_code`
- `c_address_matchcode` (44 characters)
- `c_city_matchcode` (15 characters)
- `c_usps_result_code`
- `c_usps_numeric_result_code`