

Immunization Information System (IIS)
Implementation Guidance for a
**Shared Address Cleansing
and Geocoding Service**

May 2017



AIRA
AMERICAN IMMUNIZATION
REGISTRY ASSOCIATION

Table of Contents

Acknowledgments	2
Chapter 1. Introduction	3
Chapter 2. Benefits of Address Cleansing and Geocoding	4
Chapter 3. Getting Started	5
Chapter 4. Implementation Support	7
Chapter 5. Connecting to SmartyStreets	9
Manual Batch Processing (Command-Line Tool)	9
Integrating the Address Cleansing API (HTTP Request/HTTP Response)	11
Automated Batch Processing	11
HL7 Real-Time Solution	12
User Interface API (Direct Data Entry)	13
HTTP Request/HTTP Response Technical Guidance	14
Chapter 6. Final Considerations	16



Acknowledgments

The American Immunization Registry Association (AIRA) would like to acknowledge and thank the following individuals and organizations for their support and assistance with this important project:

The primary researcher and writer on this project:

Danielle Reader-Jolley, Public Health Consultant

The members of the Joint Development and Implementation Advisory Workgroup (JDI Workgroup):

Noam Arzt, President, HLN Consulting, LLC

Belinda Baker, IIS Manager, State of Washington Immunization Program (and Board Rep.)

Aaron Bieringer, IIS Technical Lead, State of Minnesota Immunization Program

Bill Brand, Director of Informatics Science, Public Health Informatics Institute (PHII)

Brittany Ersery, IIS Manager, State of Kansas Immunization Program

Michael Flynn, IIS Technical Lead, State of New York Immunization Program

Mandy Harris, IIS Manager, State of Nevada Immunization Program

Therese Hoyle, Public Health Consultant, State of Michigan

Judy Merritt, Director of Data, Scientific Technologies Corporation

Steve Murchie, CEO, Envision Technology Partners (Jim Holsinger sub)

Gary Wheeler, IIS Executive and Strategist, DXC Technology

Warren Williams, Health Scientist, Acting Branch Chief, IIS Support Branch, CDC

Gerri Yett, Program Manager, State of Alaska Immunization Program (retired)

The Address Cleansing Pilot Participants:

Belinda Baker, IIS Manager, State of Washington Immunization Program (and Board Rep.)

Michael Flynn, IIS Technical Lead, State of New York Immunization Program

Therese Hoyle, Public Health Consultant, State of Michigan

Josh Hull, Software Developer, Kunz, Leigh and Associates (Michigan Care Improvement Registry)

Baskar Krishnamoorthy, Business/Project Analyst, Florida Department of Health IIS or SHOTS

Steve Murchie, CEO, Envision Technology Partners (Jim Holsinger sub)

Lonnie Peterson, Child Profile Health Promotion Supervisor, Washington DOH

The AIRA Board of Directors who provided input at various stages of the effort:

Michelle Hood, Nebraska Department of Health and Human Services (President)

Kim Salisbury-Keith, Rhode Island Department of Health (President-Elect)

Mary Woinarowicz, North Dakota Department of Health (Immediate Past President)

Jenne McKibben, Oregon Immunization Program (Secretary)

Belinda Baker, Washington State Immunization Information System (Treasurer)

Bridget Ahrens, Vermont Immunization Registry (Director)

Baskar Krishnamoorthy, Florida Department of Health IIS (Director)

Quan Le, Louisiana Immunization Program (Director)

Megan Meldrum, New York State Immunization Information System (Director)

Kevin Dombkowski, University of Michigan, Child Health Evaluation and Research Unit (Director)

Brandy Altstadter, Scientific Technologies Corporation (Director)

The AIRA Staff who contributed to this document's development:

Rebecca Coyle, Executive Director

Mary Beth Kurilo, Policy and Planning Director

Maureen Neary, Senior Project Manager

Amanda Branham, Staff Support, AIRA

In 2014, AIRA was awarded funds through a Cooperative Agreement for Enhanced Standards Support for the Immunization Information System (IIS) Community. The Cooperative Agreement provides support for the creation of a governance structure to facilitate and guide IIS joint development efforts. AIRA has convened the Joint Development and Implementation Advisory Workgroup (JDI Workgroup) to oversee and advise this process, including the launch of an initial joint effort.

Members of the JDI Workgroup decided on a centralized address cleansing and geocoding service as the initial joint development concept project due to its ability to positively impact a variety of data quality and assessment activities. Outreach to the IIS community was performed to identify solutions that have already been implemented/investigated by AIRA members and determine challenges or barriers that should be considered in the selection of a centralized solution. After reviewing these findings, an effort was pursued to identify an appropriate service provider for AIRA to utilize for the centralized address cleansing and geocoding service.

Market research was performed to identify a list of potential candidate services, and from this list, three service providers were selected for further investigation. After a series of interviews, product demonstrations, and documentation analysis, SmartyStreets was selected as the preferred service provider to partner with AIRA.

The purpose of this document is to provide implementation guidance to IIS projects interested in taking advantage of the SmartyStreets address cleansing and geocoding services being offered through AIRA. This document describes some of the benefits of using address cleansing in an IIS, how and where the service can be leveraged in the various IIS workflows, technical guidance on connecting with SmartyStreets, as well as policy considerations and business rules/best practices for service implementation.

Note: *The selected service supports standardization, verification and geocoding of the address only. This means that the address will be formatted to comply with USPS address specifications, the address will be verified against the USPS database to ensure that it is a valid USPS deliverable address, the address will be corrected/updated as appropriate, and additional metadata related to the specific address will be provided in conjunction with the corrected address. This service **does not** include any patient-specific support for address updates (e.g., address changes and forwarding).*

Much of the content included in this document was assembled from the collective efforts of the JDI Workgroup, the Address Cleansing Working Group (a subgroup of the larger JDI Workgroup), and the four pilot projects that agreed to evaluate the various connection methods between IIS and SmartyStreets. Community lessons learned and best practices will continue to evolve as more IIS move towards implementing and integrating the address cleansing service.

WHY SMARTYSTREETS?

SmartyStreets was selected for the following reasons:

- Impressive validation and correction abilities
- Unlimited transaction pricing
- Easily accessible and readable documentation
- Excellent technical support
- Best option for batch processing of data at rest
- Option for self-hosting by AIRA if needed/preferred



1 <https://smartystreets.com/>

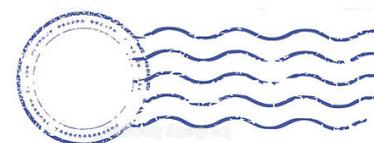
Chapter 2

Benefits of Address Cleansing and Geocoding

Address cleansing and geocoding can provide a number of benefits to both the IIS and Immunization Program operations. Some of the ways that address cleansing/geocoding can be leveraged, as noted by the JDI Workgroup, include:

- All addresses in the IIS database would be verified and formatted in accordance with the USPS database and standards.
- Standardization of addresses will improve patient-level deduplication by eliminating variability in addresses evaluated by the match algorithm.
- Address cleansing and standardization of all address elements improves the quality and accuracy of reports run by various address and geographic parameters (e.g., AFI and other coverage assessments).
- Improved address quality may help to better reconcile the denominator of patients in the IIS versus the actual census population (especially where the IIS count is inflated).
- Geocoding improves mapping capabilities and allows programs to look at data in different ways and better target intervention strategies (e.g., does distance from primary care provider impact coverage rates). It could also be used to create a general map of provider offices.
- For reminder/recall, address cleansing can improve mailing success by avoiding mailings to invalid addresses, decreasing the cost of unnecessary postage, and possibly increasing patient response rates for mailed reminder/recall efforts.
- Integration of the address service ensures that a verified/standardized address is not overwritten by a “bad” address and ensures that the address syntax (city, state, zip, county, and latitude/longitude) remain intact at all times. This has an added benefit of prohibiting users from artificially changing address elements to improve patient counts or coverage assessment outcomes.
- Standardization of addresses creates an opportunity to leverage or improve household grouping functions by identifying all family members associated with a particular address.

Ultimately, standardization of addresses allows addresses to be more easily compared and improves the quality of IIS reporting capabilities.



PLEASE DO NOT BEND!

Chapter 3

Getting Started

Prior to connecting to SmartyStreets, users should perform several activities in preparation for connection to the service.

STEP 1: VISIT THE SMARTYSTREETS WEBSITE

The first step is to visit the SmartyStreets website and become familiar with the service and the connection options discussed in [Chapter 5. Connecting to SmartyStreets](#). Prospective users are encouraged to use the 'Try it now' feature, which provides an interactive sample of the application programming interface (API) features and displays the resulting metadata that will be available to the IIS.

STEP 2: REVIEW THE CURRENT IIS INFRASTRUCTURE

The next step is to review the current IIS infrastructure for how addresses are processed and stored. For IIS that store multiple addresses, historical addresses, and/or support concepts of record "ownership" or public/private field display, additional discussion will be needed about how to handle the cleansed addresses returned from SmartyStreets. This issue will present particular challenges for cleansing of addresses that have already been recorded to the IIS database (e.g., any batch processing of 1+ records) where the IIS will need a mechanism to reattach the cleansed address to the patient record. IIS should discuss the following questions:

- What happens to the new address returned from SmartyStreets? Who is able to view and/or utilize this new address? Is this address displayed to the provider? What about the new metadata elements such as county, latitude/longitude, etc.?
- What happens to the address originally submitted by the provider? Is this address replaced with the new address? Is the original address stored in the database?
- Which username/user ID will be attached to this change in the audit logs (e.g., SmartyStreets as a user/user ID)?

In addition, the IIS should also establish business rules for the following items:

- What unique identifier will the IIS use/generate to match the new address back to the correct patient and provider?
- Under what conditions do you accept or reject the new address?
 - Prospective IIS should review the various SmartyStreets disposition codes³ to determine how each code should be processed (e.g., accept, reject, flag for review).
- Is there a need to store the record-level disposition codes for why or how the address was altered by SmartyStreets in the IIS database?
- Does an unclesed address need to be flagged or noted in the database for future processing?
- If SmartyStreets determines that an address is undeliverable, invalid or vacant, does the IIS have a mechanism to flag the address as invalid in the IIS database based on this information?
- If the IIS is required to interact with a health information exchange (HIE), where does the HIE fit into the proposed dataflow?

2 <https://smartystreets.com/>

3 Address output fields: <https://smartystreets.com/docs/cloud/us-street-api>

4 Idem.

STEP 3: REVIEW THE METADATA

After reviewing address management protocols, the next step is to review the metadata returned by SmartyStreets and determine which fields are of interest to the IIS. After these fields have been identified, a review of the IIS database tables should be performed to determine what additional fields may need to be added to support the additional inbound SmartyStreets data elements. Metadata options can be reviewed using the 'Try it now' feature on the SmartyStreets web page, by reviewing the "Output field definitions⁵" for the HTTP Response, or by reviewing the "sample-output.csv" file that comes with the Command-Line Tool download described in [Chapter 5. Connecting to SmartyStreets](#).

STEP 4: GET CONNECTED

When the IIS is ready to begin the connection process, the program should review the SmartyStreets "Terms of Service⁶" and "Private Label Agreement⁷." The program will then need to contact AIRA for a copy of the AIRA Address Cleansing Service Partner Agreement that outlines basic roles and responsibilities. Programs should

determine who is ultimately responsible for reviewing the necessary agreements and authorizing the IIS to interact with SmartyStreets. Programs should also visit with their jurisdiction's privacy or security officer if there are any HIPAA-related concerns or protocols.

Once the program has signed and returned the Partner Agreement, AIRA will issue a unique Authentication ID (auth-ID) and Authorization Token (auth-token) that will allow the IIS to access the SmartyStreets address cleansing service.

***Note:** The assigned auth-ID and auth-token may be used only by the participating IIS for integrating SmartyStreets into the IIS workflow or processing batch files at the administrative level. Credentials are not to be shared with other programs within the agency or with any end user (e.g., providers, EHRs). This program is offered exclusively for the use of AIRA member IIS.*

***Note:** Participating IIS must also have access to firewall port 443 and ensure that it allows external access to the SmartyStreets service address (<https://us-street.api.smartystreets.com>).*



GETTING STARTED CHECKLIST:

- Get familiar with SmartyStreets (<https://smartystreets.com>).
- Determine your preferred connection method(s) for using the SmartyStreets service and utilize the appropriate documentation on the SmartyStreets website (<https://smartystreets.com/docs>).
- Perform any necessary IIS development to connect to the service, modify import routines, or collect new fields (metadata).
- Review the SmartyStreets "Terms of Service" (<https://smartystreets.com/legal/terms-of-service>) and "Private Label Agreement" (<https://smartystreets.com/legal/reseller-agreement>).
- Agree to and sign the AIRA Address Cleansing Service Partner Agreement.
- Receive unique authentication ID and authorization token issued by AIRA.
- Verify support for port 443 and access to the SmartyStreets service address (<https://us-street.api.smartystreets.com>).

5 <https://smartystreets.com/docs/cloud/us-street-api#http-response-output>

6 <https://smartystreets.com/legal/terms-of-service>

7 <https://smartystreets.com/legal/reseller-agreement>

Chapter 4

Implementation Support

While this document will attempt to provide as much guidance as possible to facilitate the IIS-SmartyStreets connection, there will likely be additional questions or support needs throughout the implementation process. The following summarizes primary roles/responsibilities and whom to contact with common questions.

AIRA has purchased a multiyear “Private Label⁸” service subscription from SmartyStreets (address validation, standardization, and geocoding) for use by AIRA member IIS in good standing. AIRA will provide credentials for accessing the SmartyStreets service at no charge to interested IIS projects. As detailed in the Partner Agreement, AIRA assumes no responsibility for the performance or accuracy of the SmartyStreets service.

SmartyStreets as the service provider is responsible for maintaining the actual address service and providing interface support. These roles and responsibilities are described in the SmartyStreets Terms of Service⁹ agreement posted on its web page. The AIRA subscription with SmartyStreets includes access to technical assistance for participating IIS provided directly through SmartyStreets customer support.

Each IIS will be responsible for making any necessary enhancements required to interact with the SmartyStreets service. Developers are encouraged to review all of the appropriate documentation and developer toolkits provided by SmartyStreets to better understand the level of effort required for implementing the desired connection methods described in [Chapter 5. Connecting to SmartyStreets](#).

AIRA will also convene a virtual “user group” for IIS projects actively implementing the SmartyStreets service. This user group will provide an opportunity for members to discuss challenges and best practices for integrating the service, as well as maximizing possible benefits of address cleansing and geocoding. This user group will be facilitated on a monthly basis as long as there is an ongoing need/interest among participating members.

The AIRA subscription with SmartyStreets includes access to technical assistance for participating IIS provided directly through SmartyStreets customer support:

 **(877) 216-8883**

 **<https://smartystreets.com/contact>**

⁸ <https://smartystreets.com/legal/reseller-agreement>

⁹ <https://smartystreets.com/legal/terms-of-service>

The following table provides examples of whom to contact in a variety of scenarios:

Issue	Resource
Interested in more information about the SmartyStreets service and what it can offer	<ul style="list-style-type: none"> ■ SmartyStreets website¹⁰ (general info, product demonstrations, development toolkits) ■ AIRA: (202) 552-5761 or mneary@immregistries.org
Ready to get started (administrative)	<ul style="list-style-type: none"> ■ SmartyStreets website¹¹ (terms of use) ■ AIRA Partner Agreement
Ready to get started (technical)	<ul style="list-style-type: none"> ■ SmartyStreets website¹² (development toolkits, technical documentation) ■ AIRA (auth-ID/auth-token, user group participation)
Receiving errors related to invalid auth-ID/auth-token	<ul style="list-style-type: none"> ■ AIRA: (202) 552-5761 or mneary@immregistries.org
Having difficulty connecting with the SmartyStreets service after IIS interface development is complete	<ul style="list-style-type: none"> ■ SmartyStreets technical support: (877) 216-8883 or https://smartystreets.com/contact
Having difficulty interpreting SmartyStreets metadata	<ul style="list-style-type: none"> ■ SmartyStreets website (output field definitions¹³)
Having difficulty architecting the IIS to take advantage of the cleansed/geocoded addresses	<ul style="list-style-type: none"> ■ AIRA (user group participation)
Disagreement with how SmartyStreets handled a specific address/set of addresses	<ul style="list-style-type: none"> ■ SmartyStreets technical support: (877) 216-8883 or https://smartystreets.com/contact
Accidentally submitted ePHI in the record(s) submitted to SmartyStreets	<ul style="list-style-type: none"> ■ SmartyStreets technical support: (877) 216-8883 or https://smartystreets.com/contact ■ Contact the jurisdiction's privacy or security officer
General questions	<ul style="list-style-type: none"> ■ AIRA: (202) 552-5761 or mneary@immregistries.org

¹⁰ www.smartystreets.com

¹¹ <https://smartystreets.com/legal/terms-of-service>

¹² <https://smartystreets.com/docs>

¹³ <https://smartystreets.com/docs/cloud/us-street-api#http-response-output>

Connecting to SmartyStreets

There are four primary methods for connecting to the SmartyStreets address cleansing and geocoding service. The method(s) adopted by an IIS program will ultimately be determined by how the IIS/Immunization Program intends to leverage the improved addresses and where in the dataflow/workflow the address cleansing will occur.

SECTIONS WITHIN CHAPTER 5

The following sections will describe the options for connecting SmartyStreets to the IIS to support the following processes:



Manual Batch Processing (Command-Line Tool)



Automated Batch Processing



HL7 Real-Time Solution



User Interface API (Direct Data Entry)

Note: Regardless of which option an IIS chooses to pursue, only the minimum required fields¹⁴ should be submitted to the SmartyStreets service. IIS should never submit any patient, vaccination or other ePHI in the file.

Address processing requires only the following fields:

- Street Address 1 (e.g., 55 Maple Ln)
- Street Address 2, as needed (e.g., Apt. 7)
- City
- State
- Zip
- Unique Identifier/Session ID/Address ID¹⁵ (to match address back to patient)



MANUAL BATCH PROCESSING (COMMAND-LINE TOOL)

The SmartyStreets Command-Line Tool is a small piece of software that is installed on the user's desktop for processing large batch files through the address cleansing service. The Command-Line Tool should be used as the primary method for performing the initial cleansing of the entire IIS database to establish a baseline of cleansed and geocoded address data. Once installed, the Command-Line Tool can also be used as needed for any manual batch processing or spot cleansing of non-routine batches; however, after the initial database cleansing, IIS are encouraged to implement an HTTP POST script to establish ongoing batch processing routines (see [Automated Batch Processing](#)).

The Command-Line Tool download and installation instructions can be found at <https://smartystreets.com/docs>. From the left-hand menu, look for "SmartyList" at the bottom, and then select the option for "Command-Line Tool." From this web page, select the appropriate platform download for the IIS administrator's desktop. Extract the files in the downloaded folder, and then save the extracted folder to the user's desktop or the defaulted location.

Tip: copy the pathway for the extracted folder and paste it into a cheat sheet for future use (e.g., C:\Users\MickeyMouse\Desktop\smartylist).

¹⁴ <https://smartystreets.com/docs/cloud/us-street-api#input-fields>

¹⁵ IIS are strongly encouraged to implement a Session ID or Address ID. Solutions should avoid using IIS Patient ID or other stored identifier that could be tracked back to a specific patient in the future.

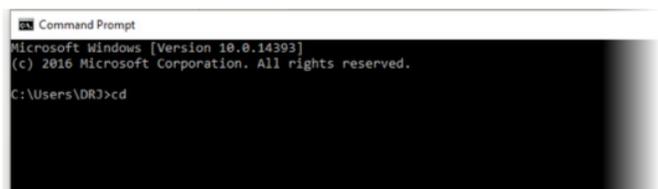
Once the Command-Line Tool is installed, users will need the auth-id and auth-token issued by AIRA to begin processing files. The installation folder includes a small test file of 500 records labeled "sample-input.csv." This file can be used to review the input format and perform an initial test of the installation.

To process a file, users will start by opening the command window by typing "cmd" into the start line. See the screenshot example below for Microsoft OS. The Mac process will be slightly different. The first step will be to change the directory to point to the SmartyList folder. The directory will likely be defaulted to the C:\ drive. To change the directory, enter "cd" after the > prompt, followed by the pathway copied to the cheat sheet (e.g., C:\>cd\Users\MickeyMouse\Desktop\smartylist). After pressing enter, the new prompt line will reflect the change to the updated directory.

Start Line:



Command Window:



At the new prompt, the user will specify the necessary criteria for SmartyStreets to process the file by entering/copying the following string syntax: smartylist -input="sample-input.csv" -auth-id="xxxxxx" -auth-token="xxxxxx". The input in quotations will be the name of the file containing the records to be cleansed. The auth-id and auth-token in quotations will be the unique string of characters sent by AIRA for the specific IIS account.

Tip: Copy this populated segment and paste it into a cheat sheet for future use; only the name of the input file will need to be updated with each new session.

The resulting prompt line will look like this:

```
C:\Users\MickeyMouse\Desktop\smartylist> smartylist  
-input="sample-input.csv" -auth-id="xxxxxx" -auth-  
token="xxxxxx"
```

Note: Users should make sure that each input file has a unique file name. This unique name will then be pulled through and used for the naming of the corresponding output file.

After pressing enter, the system will assess the file to determine the number of records to be processed. Confirm the number of records to be processed by entering "y" at the confirmation prompt, and the file will be sent to SmartyStreets for processing. Depending on the size of the input file, the records should process within a few seconds or milliseconds. A status message will display the progress towards completion, followed by a notice when the file has finished. The output file will be posted back to the same folder where the input file originated. The installation folder also includes a "sample-output.csv" as an example of the output content and format.

Note: Installing the Command-Line Tool and running the initial test file should take no more than 30 minutes to complete.

Users are encouraged to contact SmartyStreets Support for any technical assistance related to the installation or queueing of the first file for processing:

 **(877) 216-8883**

 **<https://smartystreets.com/contact>**

The Command-Line Tool can run an unlimited number of records at any given time; the user's computer will be the only limiting factor for file size. The output file will assign a sequence number to each row, which represents the order in which the addresses were submitted and processed by SmartyStreets. In order to interpret the disposition of each address, users should refer to the "Address output fields¹⁶," also located under "SmartyList" on the left-hand navigation menu. The documentation on this web page details each of the columns represented in the output file. Several columns will be of particular interest to IIS and may also be used for reporting summarized evaluation metrics to AIRA:

- [summary]
- [notes]
- [dpv-match-code]
- [dpv-footnotes]
- [footnotes]

For batch processing, programs should use the "strict" match strategy. This is the default setting suggested by SmartyStreets. This match strategy will allow for a single exact match corresponding to a valid USPS address. If an exact match does not exist, the address will return an empty set "[]". These addresses will then need to be flagged as invalid and reviewed manually by the provider or an IIS administrator.

Additional Considerations for Batch Processing:

- Does the user have authorization to install software on their local machines? If not, does an IT resource need to be contacted?
- Each input file should have a unique filename that can be used to link it with the corresponding output file.

INTEGRATING THE ADDRESS CLEANSING API (HTTP REQUEST/HTTP RESPONSE)

The remaining methods (Automated Batch Processing, HL7 Real-Time Solution, and UI Direct Data Entry) rely on slight variations of the same basic technical mechanism (HTTP Request and Response) for connecting with the SmartyStreets API. The following subsections will describe the nuances of the remaining implementation options followed by the [technical guidance](#) for leveraging these approaches to address processing.



AUTOMATED BATCH PROCESSING

After the IIS database has been fully cleansed, a routine process should be implemented to maintain address cleansing on all newly introduced records or updates submitted for existing records. This can be accomplished by writing a script that will automate ongoing batch processing by sending the address list directly to the SmartyStreets API. The script should be written to select up to 100 addresses at a time that meet specified criteria (e.g., create date, modified date) and to send them to the SmartyStreets API using an HTTP POST request. The HTTP POST script should be written to run at a routine interval (e.g., hourly, nightly, weekly, monthly). Unlike the Command-Line Tool, the HTTP POST will be limited to a maximum number of 100 records per POST, so the script should be set to run often enough to capture all new and newly updated records to keep the database up to date.

Additional Consideration for Automated Batch Processing:

- What contingency behavior will be in place should the SmartyStreets service become unavailable for any reason? Examples: make a second attempt in 15 minutes and/or flag the record as "unverified" and resubmit with a future batch process.

16 <https://smartystreets.com/docs/plugins/smartylist/address-output>



HL7 REAL-TIME SOLUTION

There are two primary methods that can be used to process inbound HL7 messages. One method is to leverage the HTTP POST process described in the previous section. With this method, all inbound HL7 messages will be processed and recorded in the IIS database according to standard IIS workflows/business rules. The HTTP POST will then select for any new IIS records and submit them to SmartyStreets using the address cleansing API. This can be programmed to occur in a batch process (e.g., nightly or weekly) or to occur on a message-by-message basis (e.g., the record is recorded to the IIS database and is then sent immediately to SmartyStreets for cleansing/geocoding; see also HTTP GET¹⁷).

The second method is to leverage the SmartyStreets API in real-time as the HL7 message is presented to the IIS for pre-processing before it gets recorded to the IIS database. In this scenario, the address components will be peeled off and sent to SmartyStreets for processing using the HTTP GET protocol. The cleansed address and metadata returned by SmartyStreets then become part of the official HL7 submission that gets recorded in the IIS.

If performance is a concern, developers may want to establish a client-side proxy server to cache an address that has already been cleansed and validated. For example, a single vaccination encounter may result in separate VXU messages for each vaccination administered. In that scenario, there would be many attempts to cleanse the same patient address. Also consider that addresses are often the same between patients and responsible parties, so a single HL7 message may include multiple instances of an address within the same message. With a proxy server, after the first response is received from SmartyStreets, a proxy's cache could be utilized to avoid waiting for additional responses (with the same information) from SmartyStreets. The proxy server can be configured to store the cleansed address indefinitely, or it can be stored for a specified period of time (e.g., week/month) before the cached address is purged and/or refreshed.

Developers may also want to account for possible timeouts by coding for a single retry attempt. Addresses that do not successfully get processed through SmartyStreets should be flagged in the system for future batch or HTTP POST processing.

Additional Considerations for an HL7 Real-time Solution:

- For real-time HL7, since the cleansing occurs before the record is recorded to the IIS, if the cleansed address differs from what was submitted by the provider, does the originally submitted address need to be stored somehow?
- What contingency behavior will be in place should the SmartyStreets service become unavailable for any reason? Examples: make a second attempt in 15 minutes and/or flag the record as "unverified" and resubmit with a future batch process.
- Are there performance concerns that would be alleviated by implementation of a client-side proxy server to cache addresses that have already been processed and cleansed (e.g., multiple VXUs for a single patient)?

For specific questions about how to architect the HL7 real-time solution in an IIS, contact:

Nathan Bunker, AIRA

✉ nbunker@immregistries.org

📞 (202) 552-0208

Josh Hull, Michigan IIS

✉ j.hull@kunzleigh.com

📞 (616) 777-7549

¹⁷ <https://smartystreets.com/docs/cloud/us-street-api#http-request-methods>



USER INTERFACE API (DIRECT DATA ENTRY)

The SmartyStreets API can also be integrated directly into the IIS address fields used for direct data entry in the User Interface. This can be implemented

in one of two ways. The first option is to validate the entire address after it has been entered and saved to the record by leveraging the HTTP GET protocol.

The alternative option is to have the address validated in real time. In this scenario, the API will provide immediate feedback to the user with real-time address corrections/suggestions as the user enters the address on the page. This feature will behave exactly like the 'Try it now' feature on the SmartyStreets web page¹⁸. For an example, attempt the following scenario using "Peachtree" in Atlanta, GA, where "Peachtree" applies to all of the following thoroughfares:

- | | |
|---------------|----------------|
| PEACHTREE ST | PEACHTREE CIR |
| PEACHTREE RD | PEACHTREE WALK |
| PEACHTREE DR | PEACHTREE CT |
| PEACHTREE AVE | PEACHTREE BLVD |
| PEACHTREE WAY | PEACHTREE SQ |
| PEACHTREE PL | |

By entering "80 Peachtree, Atlanta, GA" using the "address components" lookup, you will get four possible responses based on deliverable addresses in the USPS database:

- | | |
|------------------|------------------|
| 80 PEACHTREE CIR | 80 PEACHTREE ST |
| 80 PEACHTREE PL | 80 PEACHTREE WAY |

From this list, the user would then select the correct address from the available options that correspond to viable USPS deliverable addresses. Once selected, the remaining metadata for the address will be displayed.

As with all of the other methods previously described, the address results from SmartyStreets will contain the full set of available metadata in the response. IIS should determine which fields they are interested in collecting and whether these fields should be displayed to the user on the data entry screen or simply be recorded to the database on the back end.

Additional Considerations for a User Interface API:

- Does the IIS need a mechanism to manually override an address when it is being handled as an unknown or "bad address" by SmartyStreets? For example, new construction where the address has not yet been officially added to the USPS/SmartyStreets database.
- Does the IIS have contingency behavior in place in case the SmartyStreets service becomes unavailable for any reason? Examples: allow manual override and flag the record as "unverified" to be resubmitted with a future batch process.

18 <https://smarystreets.com>



HTTP REQUEST/HTTP RESPONSE TECHNICAL GUIDANCE

To implement any of the previously described options for integrating the SmartyStreets service into the workflows of the IIS ([Automated Batch Processing](#), [HL7 Real-Time Solution](#), and [UI Direct Data Entry](#)), developers should use the following guidance.

All necessary documentation to establish these connections can be found at <https://smartystreets.com/docs>. To begin, developers should select “Cloud API Reference¹⁹” from the left-hand menu and familiarize themselves with the sections titled “Do not read this” and “Technical Requirements.” Developers should then refer to the section titled “US Street Address API²⁰,” which provides additional details necessary to code the HTTP Request and HTTP Response (input fields²¹, HTTP headers, output field definitions²², and sample request/response messages). The HTTP Request/Response protocol can be implemented to support both batch address cleansing (automated batch processing using HTTP POST²³) and individual address requests (Real-time HL7 or UI Direct Data Entry using HTTP GET²⁴).

For automated processing (Automated Batch Processing and HL7 Real-Time), developers should use the “strict” match strategy. This is the default setting suggested by SmartyStreets. This match strategy will allow for a single exact match corresponding to a valid USPS address. If an exact match does not exist, the address will return an empty set “[]”. These addresses should then be automatically flagged as invalid and will need to be handled manually by the provider or an IIS administrator.

For interactive data entry (e.g., coding the API into a web-user front end), a match strategy of “range” may be used to offer up all potential address matches to best assist the manual-entry user. See the example in [User Interface API \(Direct Data Entry\)](#). In this scenario, the address is being validated as it is being entered, and the end user is provided with a “range” of possible match options. If this is not desirable or is prohibited by jurisdictional IT policies, developers can implement the “strict” match strategy described above and validate the address after the address has been fully entered and saved/submitted.

Information on **how to interpret the analysis/results returned by SmartyStreets** can be reviewed on the “Cloud API Reference >> US Street Address API²⁵” page that includes guidance on “Status Codes and Results” (2.A) and “Output Field Definitions²⁶” (2.B). These output details will help to establish IIS business rules for composing the cleansed/processed address in the IIS or HL7 pre-processing routines. SmartyStreets guidance on the HTTP Response includes sample code for parsing the response message, a list of fields that can/will be included in the address response, and details on interpreting the disposition of each address.

All necessary documentation to establish these connections can be found at:

@ <https://smartystreets.com/docs>

19 <https://smartystreets.com/docs/cloud>

20 <https://smartystreets.com/docs/cloud/us-street-api>

21 <https://smartystreets.com/docs/cloud/us-street-api#input-fields>

22 <https://smartystreets.com/docs/cloud/us-street-api#http-response-output>

23 <https://smartystreets.com/docs/cloud/us-street-api#http-request-methods>

24 Idem.

25 <https://smartystreets.com/docs/cloud/us-street-api>

26 <https://smartystreets.com/docs/cloud/us-street-api#http-response-output>

Several output fields will be of particular interest to IIS and may also be used for reporting summarized evaluation metrics to AIRA:

- [dpv-match-code]
- [dpv-footnotes]
- [footnotes]

The IIS may consider storing these fields in the IIS database to retain details about the validity of an address and/or how the address may have been altered from the version originally entered/submitted by the provider. The IIS may also decide to store the originally submitted address for reference and auditing purposes.

After reviewing the initial introductory material, developers should select “</> SDK Reference” from the left-hand menu to access the various Software Development Kits, which include Java²⁷, .NET²⁸, and many others²⁹. From the SDK landing pages, developers should pay specific attention to the sections titled “How to Get It” and “How to Use It,” where they can utilize the sample code supplied for “Verifying Multiple US Street Addresses” or “Verifying a Single US Street Address” depending on how the IIS is implementing the service connection. Please note that much of the SmartyStreets code can be cut and pasted from the SmartyStreets website into the user’s development environment. In general, the SmartyStreets documentation will provide the basic “concepts” for establishing the API interface, but the exact fields/labels will need to be customized to fit the various nuances of each individual IIS. For example, developers may need to modify the field values for the address “lookup” components to align with how addresses are submitted to the IIS (e.g., Street2, Zip³⁰).

Note: For the Direct Data Entry/User Interface API, SmartyStreets also offers a “Website plugin³¹” option. The SmartyStreets website plugin is canned code that can be bound to most address forms. Developers should assess whether this code is compatible for integration into the IIS. If developing custom code for interacting with the API is preferred, developers should follow the general guidance under “US Street Address API³².”

SmartyStreets has also offered the following link as an additional resource to AIRA members: <https://github.com/smartystreets/LiveAddressSamples>. This link includes a number of code samples and supplemental references that may be helpful for implementing the US Street Address API into the IIS.

Note: Developers should estimate about 4-8 hours to complete the interface between the IIS and SmartyStreets API once the appropriate infrastructure has been established in the IIS to import the cleansed address and desired metadata.

For any of these options, users will need the auth-id and auth-token credentials issued by AIRA to begin processing records. Developers are encouraged to contact SmartyStreets Support for any technical assistance related to connecting with the address cleansing service using the HTTP Request/Response protocol ((877) 216-8883 or <https://smartystreets.com/contact>).

27 <https://smartystreets.com/docs/sdk/java>

28 <https://smartystreets.com/docs/sdk/dotnet>

29 <https://smartystreets.com/docs/sdk>

30 See <https://smartystreets.com/docs/cloud/us-street-api#input-fields> for mapping address fields in the IIS to supported values in SmartyStreets.

31 <https://smartystreets.com/docs/plugins/website>

32 <https://smartystreets.com/docs/cloud/us-street-api>

Chapter 6

Final Considerations

To the extent possible, this document has attempted to provide implementation guidance to IIS programs interested in leveraging the SmartyStreets address cleansing and geocoding service provided through AIRA. As this program is pioneering the use of a new shared service, there will likely be many questions that emerge throughout the implementation process related to business rules, architecting the solution, and general best practices. AIRA will facilitate a regularly scheduled user group for interested parties and document discussions for review by the larger IIS community.

In addition to the general implementation considerations and discussion items previously detailed in this document, there are some additional considerations that should be discussed by all IIS:

- Is there a plan for notifying providers about the new address cleansing tool? Is there a plan for educating providers about why the address in the IIS may differ from an address they submitted?
- When an address is determined to be invalid/undeliverable, does the IIS have a mechanism to exclude these addresses from mailing efforts and/or coverage assessments?
- Should the IIS include a report of invalid/undeliverable addresses for review and follow-up by administrators/providers?
- Does the IIS need a mechanism to manually override an address when it has been or is being flagged as not valid by SmartyStreets?

There are also a number of elements that will be IIS specific and may need to be discussed/resolved at the individual program level:

- How addresses are supported in the IIS—multiple addresses, historical addresses, ownership, public/private field display
- How address fields are structured and labeled (e.g., individual fields vs. single address string, support for additional granularity like “Street2”)
- The method(s) that an IIS chooses for interacting with SmartyStreets
- Which metadata fields the IIS is interested in capturing/storing

In order to evaluate the value of the SmartyStreets address cleansing service and to secure long-term availability for the program, IIS programs will be asked to provide a small subset of evaluation metrics to AIRA on a monthly basis as a requirement for using the SmartyStreets service. IIS may need to develop a report or query to retrieve this information. The following metrics will be assessed:

- Total number of records submitted to SmartyStreets during the reporting period
- Number of addresses that were updated as a result of address cleansing
- Number of addresses determined to be invalid/undeliverable as a result of address cleansing

For all additional questions about the address cleansing and geocoding joint services project, please contact Maureen Neary, senior project manager at AIRA:

 **(202) 552-5761**

 **mneary@immregistries.org**