Financial Institution Batch File Transfer Specifications Version 4.2

Date: March 8, 2019

TABLE OF CONTENTS

1. INTRODUCTION

1.1. The e4641 Program

SSA's e4641 program was developed by the Social Security Administration (SSA) to greatly simplify and modernize the exchange of account information between SSA and the financial institution (FI) community when required for the determination of a customer's eligibility for Supplemental Security Income (SSI) benefits. The exchange of account information between SSA and the FI community is referred to as "asset verification" which is facilitated exclusively through the e4641 program. The e4641 program permits the exchange of Asset Verification requests (requests) securely and efficiently between SSA and the FIs, expediting the eligibility determination process for the FI's customer, while at the same time complying with anti-fraud and anti-overpayment measures mandated by the federal government. An FI's failure to provide the required account information accurately and efficiently will result in the delay or denial of SSI benefits for their customers. There are different methods that an FI can utilize to process requests, but for FIs that receive a high-volume of requests and/or have to sustain a large manual workforce to accommodate workloads, there is no method more efficient, more accurate and more cost-effective than the automated options provided by SSA via the e4641 program.

In response to the FI community's demand for a method to eliminate the manual burden of processing requests, SSA developed two methods – batch and webservice – that enabled FIs to automate the entire workflow of request processing. This document specifically covers automation via batch. If your FI is interested in the option of a webservice interface, please contact SSISupport@SSAForm4641.net.

1.2. Benefits of Automation

The benefits of automation are substantial and can be realized by any FI, regardless of size.

- Eliminate request backlogs immediately: Upon implementation of the automated solution, all outstanding requests associated with your institution will be processed immediately through the new automated workflow.
- Compatibility guaranteed: There is no threat of SSA not supporting the automated solution since it is direct with the agency and not through a third-party processor.
- Produces unmatched efficiencies: Facilitates quicker review of your customers' SSI applications, allowing benefits to be deposited more efficiently.
- Creates operational bandwidth: Allows for the reallocation of manual resources towards other needs.
- Establishes operational stability: Protects the institution from month-to-month fluctuations in volume and the threat of over or under-staffing.

- No fees incurred/cost neutral: There is no cost associated with utilizing the automated methods, while the FI's own development costs are recovered very quickly through savings.
- Enhances quality of information: Eliminates manual entry errors and the risk of misreporting.
- Strengthens anti-fraud compliance: Supports anti-fraud/improper payments measures required by the federal government.
- Establishes a solution for all asset verification needs: The automated solution is based on a federal standard and therefore can be used with all federal and state government agencies requiring asset verification.

1.3. Acceptance and Compatibility

The long-term availability and compatibility of the batch process are assured through the e4641 program since it interfaces directly with SSA. Therefore, there is no risk of incompatibility or non-acceptance due to circumstances outside the control of the FI. Services provided external to the e4641 program (third-party processors) are not assured of acceptance or compatibility.

1.4. e4641 Program Administrators

Accuity is the company contracted with SSA to administer the e4641 program, including the integration of SSA's automated solutions with the FI community. Over the past decade, Accuity has developed a highly-effective integration program that navigates the FI through each stage of the automation project. Accuity's integration program is proven, successfully converting a multitude of FIs, of varying size and degrees of complexity, into fully automated processors within an average timeframe of 3 – 4 months. Automation specialists will be available to your FI throughout the automation project and will provide support wherever necessary.

1.5. Singular Solution for Asset Verification

Because the format utilized by SSA is a standard for asset verification, creating a batch interface to process SSI requests provides your institution the mechanism to automate asset verification for any government agency, including Medicaid, which is a state-based implementation. Instead of having to manage innumerable endpoints to support various asset verification processes, the batch process will provide your institution with a singular end-point to manage all of them without any additional development. The process you develop for SSI requests can be fully repurposed for all other asset verification needs. Additional information regarding how this works can be provided upon request.

2. BATCH PROCESS OVERVIEW

2.1. Current Manual Process

Currently, your institution is responding to requests manually via one of the other available methods (web, fax or mail). Requests are sent to your FI from SSA via the method your FI selected on a daily or semi-daily basis, which your FI then responds to via the same method. Regardless of the method utilized, there are manual elements involved with the processing of the requests, including, but not limited to:

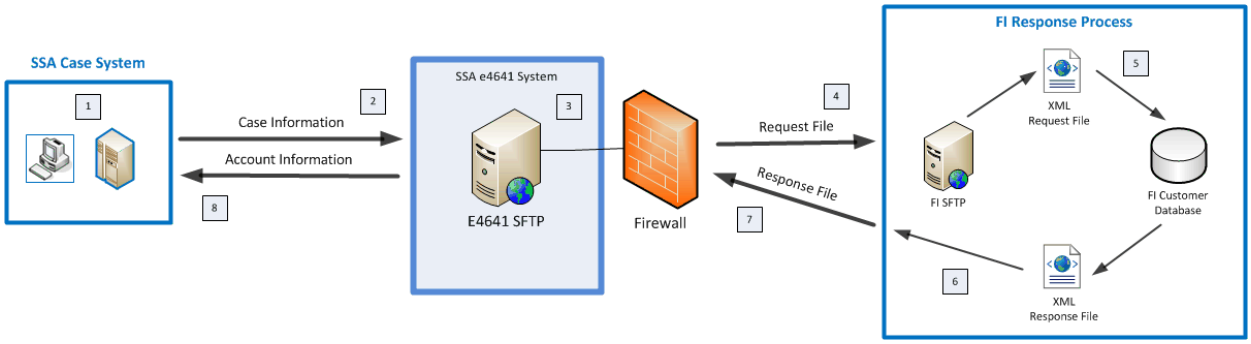
- Accessing the e4641 web-based application (web FIs only);
- Retrieving requests off fax machine (fax FIs only);
- Retrieving and opening envelopes of requests (mail FIs only);
- Searching for matching record in FI account system;
- Retrieving all accounts and monthly balance information associated with customer within the date-range specified;
- Entering balance information into web-form or writing in information by hand on paper form;
- Conducting quality control review on information entered;
- Submitting response via e4641 web application (web FIs only);
- Faxing forms (fax or mail FIs) or placing forms in envelopes for mailing.

The process of responding for each FI is unique and can be made efficient based on the procedures an FI has put in place to support the manual workflow. Regardless of the efficiencies added by the FI, there is no method more efficient, more accurate and more cost-effective than automation. All manual processes that currently burden the FI are eliminated by the batch process.

2.2. Batch Process Automated Workflow

The below workflow diagram provides a high level overview of the batch exchange relationship that will be created between your institution and the e4641 program. Each stage in the process is described in the outline below the diagram.

SSA e4641 Program
Batch Processing Workflow



Overview of batch processing stages: Please note that each step in this process can be fully automated and completed within a 24 hour timeline, start to finish:

1. The SSA Claims Rep creates a case for the Customer (SSI applicant) on SSA's case management system.

2. Relevant information (i.e. SSN, name, aliases, banking relationships, etc.) is derived from the case to generate an Asset Verification request for your FI.

3. Requests for your FI are accumulated throughout the day and staged for the creation of a single batch request file that night.

4. The request file is created in XML format, encrypted (AES or PGP) and either "pushed" to a landing zone at the FI (shown in diagram) or "pulled" by the FI from the e4641 SFTP environment.

5. The FI decrypts (AES or PGP) the XML request file. The FI will run a query against their customer account database using specific information from the request file (SSN, first and last name and date range) for each request to pull all accounts and balance information associated with the Customer.

6. Once the file is processed, the FI converts all of the response information for each request into a Response File in XML format. (There are three general response types – Accounts Found, No Accounts Found and Will Not Respond - that are covered in the schema definition sections of this document.)

7. The FI encrypts (AES or PGP) the XML response file and places the file in a dedicated directory on the e4641 SFTP server.

8. Once on the server, the e4641 program retrieves and decrypts the response file and processes all of the response information to SSA's case management system.

2.3. What and Who is Involved?

The resources typically involved with an automation development project are relatively consistent across FIs. Each of the following components or resources will most likely be involved in some aspect of the project. The title of each component is a generic name and may be referred to as something different within your financial institution:

1. Project Management – Individual or group responsible for managing the project, interfacing with internal and external components and ensuring project milestones are met. Will be the primary contact for e4641 Management. May be from business and/or technical groups.

2. Connectivity Services – The individual or group responsible for establishing connections outside the FI's network, setting firewall rules and building SFTP environments. Creates the ability to exchange request and response Files. Sometimes referred to as "Technology Operations," "Network Communications," "Messaging Communications," etc.

3. Business Processes – Individual or group that understands the current request workflow and can assist with ensuring response requirements are met via the automated solution. Sometimes referred to as "Business Operations" or "Business Analyst."

4. Developers – Individual or group responsible for developing automated process of searching account system and retrieving response information. Sometimes referred to as "IT Developers."

5. Quality Assurance – Individual or group responsible for conducting testing during and after development. Depending on the FI, this may include developers and/or Business Operations. Sometimes referred to as "Quality Control," "User Acceptance Testing Group," etc.

Supporting the integration process for SSA is:

6. e4641 Automation Team – The e4641 Automation Team (e4641 Team) is comprised of the Accuity personnel responsible for facilitating integration. This expert team includes project management, network operations, developers and Quality Assurance personnel, each of which will be engaged with the project and available to the FI for support.

2.4. Stages of Automation Project

In general, there are four stages involved with an FI's development and implementation of an automated interface with the e4641 program. Each stage contains critical tasks that must be executed fully and correctly to ensure efficient progression through the project. The stages and the key tasks associated are identified below:

1. Orientation and Design

    a. Batch Profile document issued to the FI for review and completion.

b. Schema specifications (included in this document) along with sample XSD, request and response files provided.

c. Schema review meeting conducted with FI technical and business resources to ensure complete understanding of workflow and data requirements.

d. Support provided to FI throughout design phase. Weekly meetings scheduled to discuss status and to answer FI questions as the project moves to development phase.

2. Development/Configuration

a. FI develops automated method of processing requests and creating response file according to specifications.

b. In parallel with development, the Connectivity Group at the FI works with the e4641 Team to create SFTP exchange relationships for both test and production environments. The completed Batch Profile document is used.

c. Weekly meetings are maintained until connectivity is established and the FI no longer requires weekly support. Communication with the FI will remain fluid.

3. Testing

a. The e4641 standard automation test plan will be provided to the FI. The FI may augment the test plan if additional testing is desired.

b. End-to-End tests are conducted. Responses provided via automation are compared to responses previously provided by the FI manually to identify discrepancies. A list of common errors and issues are screened against throughout testing.

c. The FI and the e4641 Team work through issues and continue to test until results are accurate.

d. The last test file will be a "load test."

4. Deployment

a. Deployment is scheduled to transition the FI from the current processing method to Batch. Deployments are executed over weekends.

b. If a request backlog exists, all backlogged requests are added to the initial request file for processing via Batch.

c. Users of the e4641 web application will be inactivated the Friday before the transition occurs (some users will be reactivated after deployment).

d. The initial request file is delivered to the FI in the early morning of the first business day after the weekend deployment.

e. Processing requests is fully automated.

2.5. FI Qualifications Checklist – Batch Processing

To establish a Batch connection with SSA, the FI will be required to develop the technology internally to pull the required information from their Customer database and then return that information in the required format back to SSA. The Batch process is available to all FIs, regardless of size or need, but to ensure that an FI has the ability to support the associated technology and security requirements, the FI must adequately meet the general qualifications provided in the checklist below.

- ❑ Develop the required technology to interface with the e4641 System according to the detailed specifications provided in this document.
- ❑ Comply with all security and format standards provided in the detailed specifications, including but not limited to:
  - Support the reading and rendering of XML according to specification;
  - PGP or AES encryption;
  - Use SSH keys;
  - Use SFTP for exchange of files.
- ❑ Provide the IT resources at the FI's own expense for the entirety of the project.
- ❑ Comply with User Acceptance Testing.
- ❑ Provide historical balance data for a minimum of 24 months, beginning with the current month (Data archives of less than 24 months will not be sufficient).
- ❑ Provide first of the month balances for each balance reported.
- ❑ Provide open and closed dates for each account where the account was opened or closed during the timeframe requested.
- ❑ Provide full names of all account holders on each account and account titles.
- ❑ Develop and support the logic required to pull the required information for each response included in the response file.

- Provide a SFTP site that has a static IP address and possibly firewalled security features or be able to provide a statically IP configured server that will access the e4641 SFTP site.

3. BATCH PROCESSING REQUIREMENTS

The following sections provide detailed specifications for the development of the Batch processing interface with the e4641 System.

3.1. Exchange of Batch Files

The configuration of the FI and e4641 exchange relationship via SFTP will be based on information captured in the Batch Profile Document.

3.1.1. Batch Profile Document

The Batch Profile Document is one of the most critical components of an efficient and successful Batch integration. This document incorporates all of the connectivity information associated with both end-points, the FI and e4641. The Batch Profile Document will be provided to the FI for review and completion, after which it will serve as a common specifications source that both end-points will base their exchange relationship on. Development of the exchange relationship will occur in parallel to the FI's development of the request/response processing solution.

3.2. SFTP Server Configuration

Financial institutions will login (via script or manually) to a designated SFTP server to transfer Batch files. Dedicated directories will be configured for the FI where request and response files will be placed. Automated processes (authenticated users) to place or retrieve Batch files will be restricted to the directories assigned to that FI's account. Access logs to the SFTP server are maintained, including failed and successful login attempts, as well as all file transfers, which are used for "sanity" checks and reporting. The e4641 SFTP server will be using key authentication instead of username/password authentication, so private/public keys will need to be configured.

The e4641 System will track the creation of the request file and the results of processing response files in the e4641 database. FI users with access to the e4641 web-based application will be able to review file processing activity and results directly from the homepage, allowing the FI to confirm successful processing of both request and response files.

3.2.1. SFTP Clients and Generating SSH Keys

There are a few SFTP clients that are available for free and listed below:

WinSCP is available here (windows platform): http://winscp.net/eng/index.php

PuTTYgen is available here (windows & Unix):

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

An SFTP client is available as part of the OpenSSH suite on most Linux systems.

As part of the SSH Batch connectivity, key generation and exchange will be necessary. Accuity uses the standard UNIX/Linux ssh key generation methodology. One way to create a unique public/private key pair is to use the following command:

$> ssh-keygen –q –t rsa –N" –f~/.ssh/id_rsa

This will create a RSA based ssh key pair installed into a user's home directory under the .ssh directory. Please make sure that the permissions on the private key are set correctly. The public key generated will have a .pub extension and will need to be installed onto the sftp server account in which access is to be gained.

On Windows, please refer to this page for information on how to generate SSH keys:

http://winscp.net/eng/docs/ui_puttygen

3.2.2. Accessing SFTP Via Script

If feasible, it is strongly recommended that the FI utilize a scheduler and script to automate the process of accessing the SFTP server to retrieve and/or place Batch files. This will prevent the inherent issues associated with a manual logging into the server. For FIs that use Windows, it is most likely a scheduled service that runs a script to execute the job. For Unix FIs, a "cron job" can be used to schedule the script to execute the job. The e4641 Team can provide your FI with guidance on this process, if necessary.

3.2.3. Directories

There will be two folders created for the FI on the e4641 SFTP server: (1) requests and (2) responses. The e4641 System will deposit encrypted files into the requests folder, which the financial institutions will then download. Financial institutions will upload encrypted files of responses into the responses folder. Access to the secure folders will be limited to specific IP's as viewed by the e4641 SFTP server (information will be provided in the Batch Profile document. See Section 3.1.1 "Batch Profile Document").

3.2.4. Push or Pull Request File

Financial institutions can choose to receive the encrypted request files on their SFTP server (push) or they can choose to login to the e4641 SFTP server and retrieve the request files (pull). In the case of "push," the e4641 System will deliver the encrypted request files to the FIs. In the case of "pull," the FI is responsible for retrieving the encrypted request files from the e4641 SFTP server. It is strongly recommended that the FI set up an automated job to pick-up or retrieve the request files.

****Pulling of request files is highly preferred*

3.2.5. Push Response File

Financial institutions **MUST transfer their encrypted response files to the e4641 server via SFTP.** The e4641 System does **NOT** login and retrieve the response files from the FIs. It is strongly recommended that the FI set up an automated job to transmit the response file.

3.2.6. File Transfer Notifications

Success or failure messages will be communicated via email. However, since email is not a secure communication channel, details of processing success or failure will not be included in the email. Examples of notifications are provided in Sections 4.1.1 "Timing and Frequency" and 5.1.1 "Timing, Frequency and Content" of this document.

3.3. Encryption

All request and response files must be encrypted both in flight as well as at rest; therefore, the files will be encrypted using either AES or PGP. If feasible for the FI, AES is recommended for its ease of use, but PGP is equally acceptable.

- The e4641 AES 256 bit encryption standard uses the encryption/decryption tool AESCrypt (A free AES encryption/decryption tool is available at http://www.aescrypt.com/). The e4641 System, upon creating request files, will encrypt the file given the associated FI key. Then this file will be dropped into the specific folder allocated for the FI. The FI will then decrypt the file, using the shared key before processing it.

- The PGP encryption will require the generation of a key pair and an exchange of public keys. For request files sent to the FI, the FI must provide a public PGP key that will be used to encrypt the file before it is sent to the FI. The FI will then decrypt the file using its own private key and pass code. For response files, the FI will use the Accuity PGP public key to encrypt the file which will then be sent to Accuity to be decrypted using Accuity's PGP private key and pass code.

Upon sending the response file, the FI will encrypt the file and then place the file in the secure folder for processing by e4641.

4. REQUEST FILE SPECIFICATIONS

4.1. Request File Processing

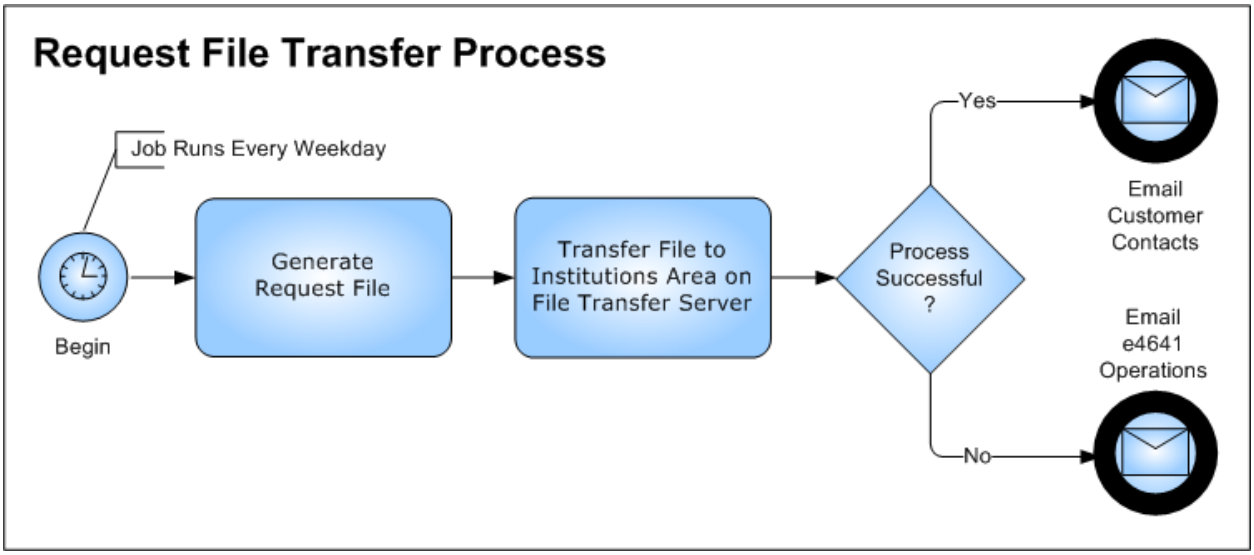


Figure 1: Request File Transfer Process

The request file contains each request targeted to that FI throughout the processing day and provides the information necessary for the FI to execute an automated search process to locate and retrieve account information for each Customer identified. The information in the request file incorporates both the request detail (only a few elements are relevant to a search) and referential information that allows the e4641 System to reconcile a response from your FI to the originating request in the e4641 System.

4.1.1. Timing and Frequency

One request file is created for an FI for each processing weekday where at least one request was sent to the FI by SSA. The request file will be created within the same timeframe each day, (e.g. 1 AM to 2 AM CST Monday through Friday, not on federal holidays), but FIs are encouraged to configure their retrieval of the request file between 4 AM to 6 AM CST to ensure readiness. The request file will contain all the new requests addressed to your FI throughout the preceding processing day. The size of the file will vary based on the number of requests, but will never exceed 20 MB (average size of a request file is 1 to 3 MB).

Once generated, the request file is encrypted (AES or PGP) and either placed for retrieval from the "requests" directory on the e4641 SFTP server or the request file is pushed to a landing-zone on the FI's SFTP server. An email notification is automatically sent by the e4641 System to the designated contacts at your FI when the file is ready for retrieval.

The success email message for request file creation will be a simple text email with the following content. The failure email for the request file will only go to e4641 Operations personnel. Files that fail are resent.

Example request file email:

```
From: ssassiadmin@ssaform4641.net
Sent: Monday, January 30, 2010 1:02 AM
To: Contact, FI
Subject: Notice of e4641 Request File Availability

Request File 1010_201001300101 is now available.
```

4.2. Request File Format

To establish a supportable standard within the FI community, as the format of asset verification, request and response files utilize the XML format in UTF-8 encoding.

The naming convention for the Request files will be as follows: **SSA-<batch_id>_<timestamp>.aes** or **SSA-<batch_id>_<timestamp>.pgp**

The following is an example of the format and contents of a request file. Only one request is represented in this example:

```
<?xml version="1.0" encoding="UTF-8"?>
<AVSRequests count="1" created="20130711012821">
<request id="112583" startDate="201108" endDate="201307" retransmit="Y"
submitted="20130729">
  <SSARep>
    <lastName>Elk</lastName>
    <middleName>N.<middleName>
    <firstName>Anne</firstName>
    <address>4709 Golf Road, 6th Floor</address>
    <city>Skokie</city>
    <state>IL</state>
    <zip>60076</zip>
    <phone>xxx-xxx-xxxx</phone>
  </SSARep>
  <customer>
    <lastName>Bakker</lastName>
    <middleName>N.</middleName>
    <firstName>Abel</firstName>
    <ssn>123-45-6098</ssn>
    <address>123 Third St</address>
    <city>Kenilworth</city>
    <state>NJ</state>
    <zip>07033</zip>
  </customer>
  <applicant>
```

```
    <lastName>Baker</lastName>
    <middleName>N.</middleName>
    <firstName>Abel</firstName>
  </applicant>
  <accounts>
    <account SSIDirectDeposit="N" joint="N">123456</account>
    <account SSIDirectDeposit="N" joint="N">652983</account>
    <remarks>This is a sample remark.</remarks>
  </accounts>
</request>
</AVSRequests>
```

A sample request file will also be provided to the FI.

4.3. Request File Elements

The XML elements are described in the following tables.

a) <AVSRequests> Element

Excerpt:

```
<AVSRequests count="1" created="20130711012821">
```

Field	Data Type	Description
AVSRequests count	Numeric	Count of requests in the file (>0).
AVSRequests created	Numeric	File creation date (YYYYMMDDHHmmSS).

The "count" and "created" attributes on the AVSRequests element are required. The value of the count is equal to the total number of requests in the file (the number of times the "Requests" element appears). At least one request element will be present in a file.

b) <request> Elements

Excerpt:

```
<request id="112583" startDate="201108" endDate="201307" retransmit="Y"
submitted="20130729">
```

Field	Data Type	Description
request endDate	Numeric	Ending Month and Year (YYYYMM) for account information.
request id	String	Unique Request ID.
request retransmit	Character	Flag (Y/N) indicating that this request is being retransmitted.
request startDate	Numeric	Starting Month and Year (YYYYMM) for account information.
request submitted	Numeric	Date (YYYYMMDD) request was submitted in e4641 System.
SSARep		Container for SSA Rep information.
customer		Container for requested customer information.
applicant		Container for applicant/recipient information.
accounts		Container for account information.

The request id, submitted, retransmit, startDate and endDate attributes on the request element will always be provided. The startDate and endDate attributes define the date-range in which the FI needs to conduct a search for accounts associated with the Customer identified on the request. These elements serve as critical parameters for ensuring the FI's search is only considering accounts that were open within the specified timeframe.

Each Request element will have sub elements of *SSARep*, *customer* and *accounts*.

c) <SSARep> Element

Excerpt:

```
<SSARep>
    <lastName>Elk</lastName>
    <middleName>N.<middleName>
    <firstName>Anne</firstName>
    <address>4709 Golf Road, 6th Floor</address>
    <city>Skokie</city>
    <state>IL</state>
    <zip>60076</zip>
    <phone>xxx-xxx-xxxx</phone>
  </SSARep>
```

Field	Data Type	Description
lastName	Character	SSA Representative Last Name.
middleName	Character	SSA Representative Middle Name.
firstName	Character	SSA Representative First Name.
address	Character	SSA Representative Office Address.
city	Character	SSA Representative Office City.
state	Character	SSA Representative Office State.
zip	Character	SSA Representative Office Zip.
phone	Character	SSA Representative Office Phone.

Though the SSA Rep information will be provided for each request in the file, the FI may ignore it.

d) <customer> Element

Excerpt:

```
<customer>
  <lastName>Bakker</lastName>
  <middleName>N.</middleName>
  <firstName>Abel</firstName>
  <ssn>123-45-6098</ssn>
  <address>123 Third St</address>
  <city>Kenilworth</city>
  <state>NJ</state>
  <zip>07033</zip>
</customer>
```

Field	Data Type	Description
lastName	Character	Customer Last Name.
middleName	Character	Customer Middle Name.
firstName	Character	Customer First Name.
ssn	Character	Customer Social Security Number (NNNNNNNNN).
address	Character	Customer Street Address.
city	Character	Customer City.
state	Character	Customer State.
zip	Character	Customer Zip Code.

"Customer" elements firstName, lastName, ssn, address, city, state, and zip will always be provided. The Customer is the individual that the search for accounts at the FI is conducted for. For the FI, the SSN is the primary search element, with a secondary element being parts of or the whole name. Depending on the matching logic incorporated by the FI, the address components may also be used in validating the identity of the Customer. Please refer to the Section 6.3.5 "Identity Matching Logic" later in this document for examples of logic used by other FIs.

e) <applicant> Element

Excerpt:

```
  <applicant>
    <lastName>Baker</lastName>
    <middleName>N.</middleName>
    <firstName>Abel</firstName>
  </applicant>
```

Field	Data Type	Description
lastName	Character	Applicant/Recipient Last Name
middleName	Character	Applicant/Recipient Middle Name
firstName	Character	Applicant/Recipient First Name

The "applicant" in this case is not relevant to the search for accounts. This person is only identified for the purposes of SSA internal processing and will not be provided for every request. The FI can ignore this information.

f) <accounts> Elements

Excerpt:

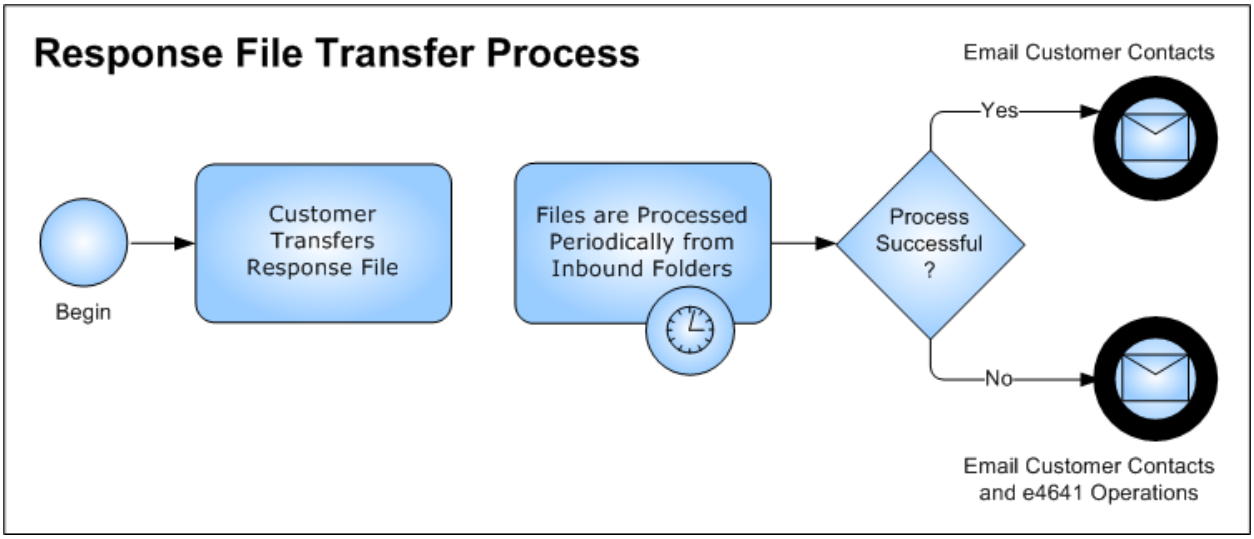
```
<accounts>
    <account SSIDirectDeposit="N" joint="N">123456</account>
    <account SSIDirectDeposit="N" joint="N">652983</account>
    <remarks>This is a sample remark.</remarks>
  </accounts>
```

Field	Data Type	Description
account	Character	Account number.
account SSIDirectDeposit	Character	Indicator (Y/N) that the account is used for Direct Deposit of SSI payments.
account joint	Character	Indicator (Y/N) that the account is a joint account.
remarks	Character	Remarks or instructions from SSA Rep.

Since the requirement is that the FI return ALL accounts associated with the Customer identified on the request, the information provided in the "accounts" element is irrelevant. The information in these fields is not very reliable and therefore it is strongly recommended that they be ignored by the FI as they can create significant complexities. Account information will not be provided for every request.

5. RESPONSE FILE SPECIFICATIONS

5.1. Response File Processing



The response file will contain the required response information for each request included in the file. Once the FI has generated and encrypted the XML response file, the file will be placed (automatically or manually) in the "Response" directory on the e4641 SFTP server. The e4641 System automatically polls the e4641 SFTP server for any new files multiple times within 30 minutes. When a new file is identified, the e4641 System will immediately run validation checks against the XSD (schema definition) and processing rules to ensure the file meets all the format and content requirements. Once the response file is validated, each response within the file is immediately processed and made available for SSA review on their case management system. If validation fails, then the file will not be processed and e4641 Team will contact the FI for resubmission. These specifics and other supporting details are covered in the following sections.

5.1.1. Timing, Frequency and Content

Unlike the request file, there is no required timeframe in which an FI must provide a response file. The FI can configure the delivery time based on their preference and/or limitations. The e4641 System sweeps the SFTP server several times an hour to check for new files, so whenever a new file is present it will be retrieved automatically. Most FIs deliver one response file each processing day, but it is not required. It is also acceptable for FIs to process multiple files in a day.

Request files and response files are "asynchronous," meaning they are not dynamically bound in any way. The e4641 System utilizes the Request ID to reconcile requests to responses; therefore, there is no constraint on the FI to provide a response in a specific response file. The count of requests in a request file does not need to match the count of responses in the subsequent response file. Every request must be responded to, but a request received on Monday can be responded to in the file delivered on Wednesday.

When a response file is received on the e4641 SFTP server, a confirmation email notification will be sent to the designated contacts at the FI. If the response file fails validation, an email notification will be sent to the list of FI contacts (specified by the FI) indicating that the file was not successfully processed. Examples of the two types of emails are below:

Example of successful response file email notification:

```
From: ssassiadmin@ssaform4641.net
Sent: Monday, January 30, 2010 1:02 AM
To: Crabapple, Edna
Subject: Notice of e4641 Response File Completion

Response File <file name> has been processed successfully. You can review
details of processing in the e4641 web application.
```

Example failed response file email notification:

```
From: ssassiadmin@ssaform4641.net
Sent: Monday, January 30, 2010 1:02 AM
To: Skinner, Seymore
Subject: Notice of e4641 Response File Completion

Response File <file name> could not be processed. Please review the details
of processing in the e4641 application.
```

5.2. Common Issues, Mistakes or Challenges

This section provides a list of common issues, mistakes or challenges that an FI may encounter as they work to integrate Batch processing. Several dozen FIs have completed the transition. Though each installation is unique, many of the issues faced, mistakes made or challenges encountered were consistent across FIs. These common threads are shared for your benefit below and the resolution to each is covered throughout this document:

- Design Phase
    - Failure to incorporate response validation rules in design

      Resolution: FI and e4641 Team will conduct schema and processing rules review sessions before and during development.

    - Not understanding response types fully

      Resolution: Explicit definitions of response types are included in this document. Also, the FI and e4641 Team will conduct schema and processing rules review sessions before and during development.

- Designing too rigid matching logic

  Resolution: Utilize matching logic that allows for certainty, but avoids misdetections due to human data errors. An example of a very effective logic is included in this document.

- Development/Connectivity

  - Not incorporating date-range in initial search

    Resolution: See Section 5.3.6 "Considering Date Range in Search."

  - Not Implementing XSD validation at the start

    Resolution: See "Response File Validation" section.

  - Submitting test files to e4641 that are not validated, resulting in failures and delays as the e4641 Team diagnosis issue

    Resolution: Incorporate schema validation in development, testing and in production and validate files prior to sending them to e4641 System.

  - Failure to configure "Response Count" correctly

    Resolution: Configure based on requirement. The "AVSResponses count" element represents the total number of responses provided in the file. This number must be greater than zero and match the count of responses explicitly.

  - Failure to configure "Accounts Count" correctly

    Resolution: Configure based on requirement. The "accounts count" element represents the total number of accounts provided for that specific response.

  - Failure to provide at least one balance within the date-range for each Account

    Resolution: Ensure Date Range is considered in search for accounts to prevent detecting accounts without valid balances.

  - Not beginning connectivity piece in parallel with development

    Resolution: Make it a priority to complete Batch Profile document before development begins. This enables the connectivity project to execute in parallel to development.

  - Underutilizing support of e4641 Team

Resolution: There are no dumb questions and the e4641 Team is experienced with broad expertise.

- Not including "open" and/or "closed" dates for an account (or all accounts)

  Resolution: Consider providing account open and closed dates (if applicable) for every account as opposed to those that were opened or closed at some point within the Date Range.

- Trying to map all Account Type codes, instead of using generic codes

  Resolution: Ignore all Account Types your FI does not use or cannot cross-reference. Use generic types like Checking, Savings, etc.

- Not ignoring irrelevant fields (i.e. SSI Direct Deposits)

  Resolution: Read response file Specifications closely. Fields like SSI Direct Deposits and Interest are included as part of the schema by default, but your FI is not expected to support it.

- Failure to insert shared private and public keys correctly

  Resolution: Complete Batch Profile document. The e4641 Team will provide guidance if there are questions regarding the use of keys.

- Testing

  - Not validating Response File

    Resolution: Incorporate schema validation in development, testing and in production and validate files prior to sending them to e4641 System.

  - Not validating Response content

    Resolution: Ensure automated response process enforces all processing rules.

- Integration

  - Expecting everything to work perfectly

    Resolution: Expect that it will not. Expect bugs to emerge within the first couple of weeks of production. The first couple of weeks are used to refine the production process.

5.3. Response File Validation

There are different types and levels of validation that need to be performed by the FI, each of which is designed to ensure the successful processing of a response according to the rules and requirements dictated by SSA. The integration and enforcement of these requirements enables the FI to realize all of the efficiencies inherent with Batch processing, including, most importantly, facilitating the expeditious determination of the Customer's SSI eligibility. Understanding this at the onset of the development process will ensure a shortened testing cycle and a smooth transition into production for the FI.

Below is a brief description of the two levels of validation the FI must perform:

XML Validation (Format): This validation should be conducted prior to a response file being encrypted for delivery to the e4641 SFTP server. Both the e4641 System and the FI are utilizing the same validation rules for enforcing well-formed XML.

Response Content Validation: Each response must be completed according to the requirements and rules dictated by the information SSA requires for SSI eligibility determination. These rules are identical to the rules your FI currently follows, but they will now be integrated into the automated response process. Key requirements and rules are defined in detail later in Section 5.3.4 "Key Response Processing Rules."

Identity Validation: An FI can utilize whatever identity validation rules it feels appropriate to ensure that when a search is conducted via the automated process, the correct Customer is identified and the correct information is being pulled. It is imperative that the FI not utilize a matching logic that is too rigid, otherwise it will create a significant number of failed detections and create more work. An example of logic utilized with proven success by other FIs using Batch is included for your information Section 5.3.5 "Identity Matching Logic." The logic implemented by your FI will be rigorously tested during the testing phase to enable refinement.

5.3.1. XML Schema Validation (XSD)

This validation should be conducted prior to a response file being encrypted for delivery to the e4641 SFTP server. Both the e4641 System and the FI are utilizing the same validation rules for enforcing well-formed XML. These rules are defined in the XSD (schema definition) provided for the FI below as well as in a separate file. Also provided is an example of a simple validation program that an FI can use to enforce XSD validation. This file validation should be instituted right away as part of the development process for the FI. This will allow the FI to determine immediately if the response file is formatted correctly BEFORE attempting to submit the file to e4641.

5.3.2. Response File XSD

The following is the response file XSD that your FI will utilize for schema validation:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="AVSResponses">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="response">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="responseType">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute name="response" type="xs:string"
use="required" />
                      <xs:attribute name="code" type="xs:string"
use="optional" />
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="accounts">
                <xs:complexType>
                  <xs:sequence minOccurs="0">
                    <xs:element maxOccurs="unbounded" name="account">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="accountNumber" type="xs:string"
/>
                          <xs:element name="accountType">
                            <xs:complexType>
                              <xs:simpleContent>
                                <xs:extension base="xs:string">
                                  <xs:attribute name="code"
type="xs:string" use="required" />
                                </xs:extension>
                              </xs:simpleContent>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="owners">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element maxOccurs="unbounded"
name="owner">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="last"
type="xs:string" />
                                      <xs:element name="first"
type="xs:string" />
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
```

```
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="balances">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element minOccurs="1"
maxOccurs="unbounded" name="balance">
                                  <xs:complexType>
                                    <xs:attribute name="date"
type="xs:unsignedLong" use="required" />
                                    <xs:attribute name="balance"
type="xs:decimal" use="required" />
                                    <xs:attribute name="interest"
type="xs:decimal" use="required" />
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                        <xs:attribute name="joint" type="xs:string"
use="optional" />
                        <xs:attribute name="SSIDirectDeposit"
type="xs:string" use="optional" />
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="count" type="xs:unsignedInt"
use="required" />
                </xs:complexType>
              </xs:element>
              <xs:element minOccurs="0" name="remarks" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="requestId" type="xs:string" use="required"
/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="effectiveDate" type="xs:unsignedLong"
use="required" />
      <xs:attribute name="count" type="xs:unsignedInt" use="required" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

5.3.3. Schema Validation Tool

The following tool can be utilized to validate that your response file meets the schema definition. This validation must be executed by the FI before each response file is encrypted and sent to the e4641 SFTP server and should be used throughout development and testing as well as in your FI's production process. Doing so will greatly minimize unnecessary delays in the development and

implementation phases. Your FI is not required to use the tool identified below, but your FI will utilize some mechanism to enforce validation of a response file prior to delivery.

Instructions for utilizing the validation tool are as follows:

Copy the program named "XsdSchemaValidator.java from http://www.herongyang.com/XSD/JAXP-XSD-Schema-XML-Validator-Final-Version.html.

Compile the program with the Java compiler as follows. If you do not have Java, you can download and install it from https://www.java.com/en/download/

>javac XsdSchemaValidator.java

An example of how to use this program to validate your response files prior to encryption:

>java XsdSchemaValidator ResponseSchema.xsd SSA-5945_20140220011148612.xml

5.3.4. Key Response Processing Rules

Each response must be completed according to the requirements and rules dictated by the information SSA requires for SSI eligibility determination. Your FI is already abiding by these processing rules via the current method your FI uses to respond. That same enforcement needs to be integrated within your FI's automated process to ensure that the search for matching account records is executed correctly and the information being provided in the response is accurate and complete. The enforcement of these rules is critical to attaining the efficiencies Batch provides and therefore will be rigorously tested during the testing phase. Below is a list of critical processing requirements that the FI must be aware of:

All Accounts: The FI is required to provide ALL accounts associated with the Customer (SSN).

First of the Month Balances: Balances provided by the FI should reflect the amount as of the first day of each month before any transactions have taken place.

Open/Close Dates: If an account was opened or closed during the date-range identified (Request startDate to Request endDate) on the request, the open and/or closed dates (mm/yyyy) must be provided for that account in the "Remarks" field. An acceptable format would be applying the last 4 digits of an account number followed by the open or closed date, separated by a colon (Example: "1234: Opened 9/2010" or "9876: Closed 10/2009"). For multiple accounts, separate the information with a comma (","). If it simplifies the response process, your FI may insert the open date and closed date for every account in the response. This will prevent your FI having to identify when to include the dates since they will be inherently included by default.

****NOTE: Balance for the month in which an account opens should not be provided due to the balance not being valid as of the first of the month as required.

Account Owners: All account owners of an account associated with the Customer must be disclosed. This information is critical to ensuring that the correct asset value is being applied to the Customer. Incomplete information could have a negative impact on a Customer's eligibility for SSI benefits.

Account Designations: Designations such as UGMA, Rep Payee etc. should also be included ideally in the account owner field; for example: John Smith rep payee Jane Smith if the designation cannot be included in the account owners field it should be placed in the remarks field as explained below.

Account Types: There are multiple account types identified in the "Account Type Codes" section of this document, but the FI is NOT required to utilize all of them. The "accountType" element is required for each account, but the FI can simply utilize the general account types like "CHECKING" or "SAVINGS" as opposed to trying to map to each of the codes offered. One valid code must be used for each account identified in the response.

***NOTE: Although you are not required to utilize all of the account type codes provided on the list, you may not use any code other than what is listed in section 6 "Account types and codes".

Remarks: The purpose of the Remarks field is to allow for the inclusion of additional information relevant to the Customer's relationship with an account that is not captured elsewhere in the file (i.e. account title, conditional access to funds, etc.). Required information, such as account "open" and "closed" dates for those accounts that opened or closed during the timeframe requested by SSA, are inserted in the Remarks (See the "Open/Close Dates" definition in this section). Including additional relevant information associated with an account can expedite the processing of a Customer's application therefore, if feasible, your FI may consider including any additional information.

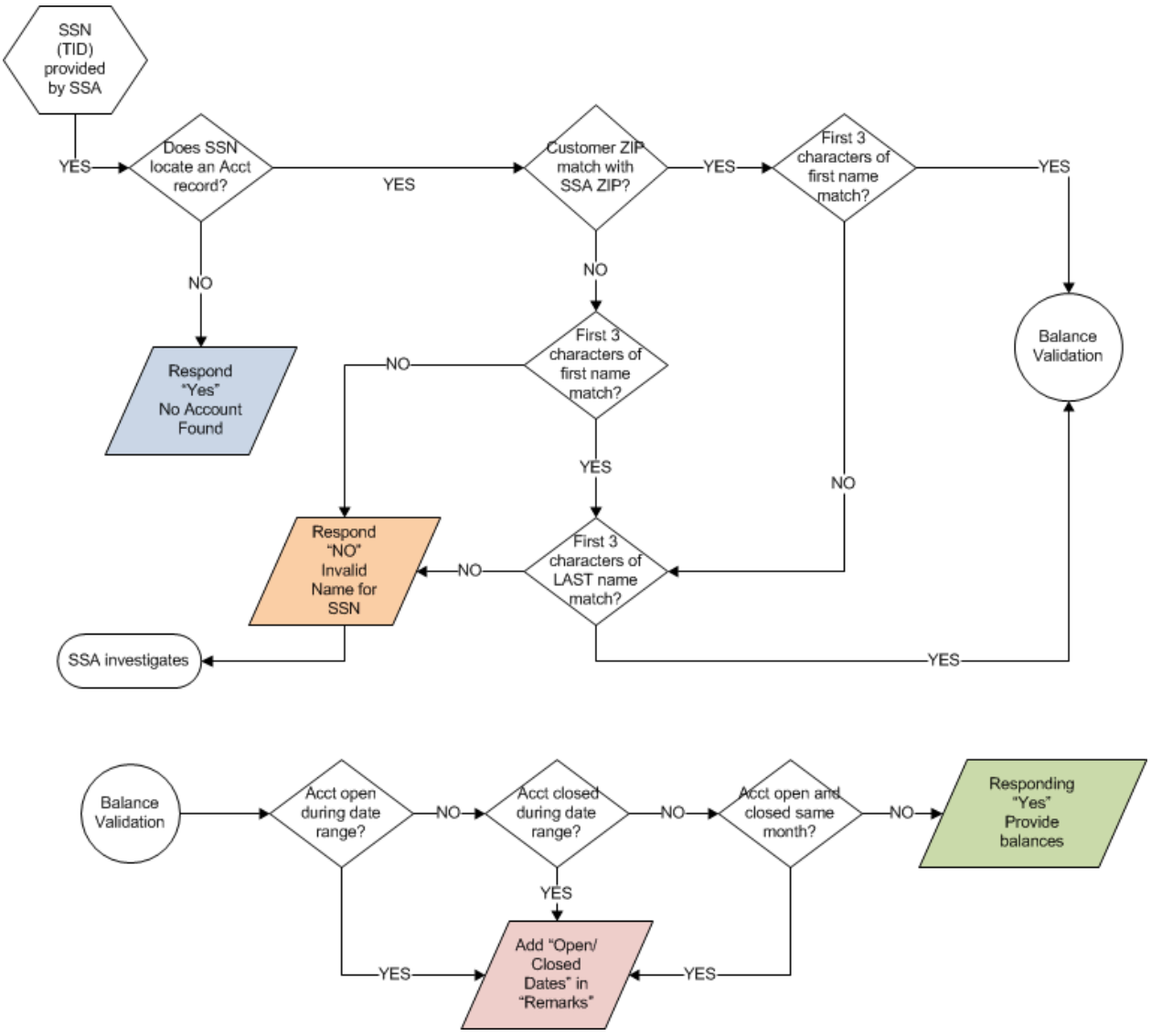
***NOTE: The FI should include a remark anytime the full range of balances cannot be provided, this will avoid the possibility of a response coming back as a manual exception.

5.3.5. Identity Matching Logic

To ensure that your FI achieves a very high success rate in detecting account records for a Customer, some form of matching logic will need to be incorporated to account for situations where the supporting data provided in the request or on your FI's account system lacks clarity. In all cases, the Customer's SSN is the primary element used for detecting a match, but to validate that the identity of the Customer is correct on the FI system, a secondary match is utilized which typically involves components of the Customer's name and, in certain instances, address information. To avoid a significant number of misdetections, your FI is strongly encouraged to use non-explicit secondary match logic. Regardless of the logic utilized, it will undergo rigorous testing to prove its efficacy.

Below is an example of a matching logic that has been refined through extensive testing and analysis over a period of years. The exact or some slight variation of the logic provided here has been

integrated by several Batch FIs and the results are extremely positive, achieving quality ratings that exceed that of human processors.



The above workflow illustrates a common matching logic utilized by other automated FIs. Your FI may elect to utilize a similar decision tree, but alter which components are checked in which order. Please note that SSA does not sanction or require the FI to utilize the logic provided in this example. It is up to the financial institution to determine what logic is most appropriate for their process.

Example of identity matching logic, step-by-step:

If the SSN provided by SSA does not have a match in the FI's database, the response is "No Accounts Found" (Covered in Section 5.5.2 "No Accounts Found"). If the SSN provided does

match, then the next logical step may be to validate with one or more secondary matches. The following is an example of how the logic would execute when an SSN is matched:

1. A match on the SSN is achieved.

2. Check the Customer's zip-code.

3. Check the first three characters of the Customer's first name (FIs attempting to match on the full name explicitly resulted in a high number of missed detections).

4. If both checks match, the FI may move on to balance validation to ensure that the accounts identified have balances within the data range specified (Covered in Sections 5.3.6 "Considering Date Range in Search" and 5.3.7 "Providing Account Balances").
    a. If one of the checks fail, a third check of the first three characters of the Customer's last name is performed.
        i. If the last name matches, the process moves on to balance validation.
    b. If last name match fails too, the FI provides a response of "Will Not Respond," with a reason code of "SSN correct, Name Incorrect" (Covered in Section 5.5.3 "Will Not Respond").

5.3.6. Considering Date Range In Search

Fundamentally, there are only a handful of parameters needed for your FI to conduct a search for accounts associated with a Customer: Customer SSN, Customer Name, Customer Address (optional and only Zip/State) and Date Range. A common mistake that FIs make when designing their search process is that the Date Range (startDate and endDate), which specifies the period of time SSA is researching, is not included as a parameter in the initial search for matching accounts (some FIs first detect if an account exists, then pull the balances via a separate step). This omission invariably results in the detection of accounts that are not applicable to the case due to either being closed before the startDate or opened after the endDate specified on the request. To comply with the processing rules and schema definition, at least one balance has to be provided within the date-range for each account, otherwise the response and/or the response file will be rejected. Your FI can avoid this issue by including the startDate and endDate as parameters in the search for accounts.

5.3.7. Providing Account Balances

Account balances (**as of first day of the month before any transactions**) must be provided for each month in which an account was open during the date-range specified on the request. For most FIs, there is a limit to the number of months they have available for any account (i.e. up to 26 months, 36 months, etc.) and therefore will only be able to respond up to that maximum. In many cases, it is much simpler for the FI to respond with the maximum number of months for every account they provide in a response, as opposed to building a complex process to dynamically pull only specific months. This is acceptable as long as at least one of the monthly balances provided by the FI falls within the date-range on the request. The e4641 System automatically filters out monthly

balances that do not fall within the specified date-range, preventing unwanted information from being included.

Months where no balances exist (months before or after an account was open/closed), the FI should NOT provide any information. Providing "0.00" for months when the account was not open is NOT acceptable, as "0.00" is interpreted as a legitimate balance.

***NOTE: If your FI does not have a true first of month balance you may use the end of month from the previous month as the first of month for the following month. Example: Jan 31st balance would be provided as the Feb 1st balance on the request.

5.3.8. Response File Error Handling

A series of operations are performed when a Response File is received, starting with an attempt to read the file. If a file cannot be read or moved to the processing step, one of the following errors will be raised and the processing of the response file will halted.

File Error	Fatal	Description
FileAccessError	Yes	Response file cannot be read or moved. Configuration issue requiring operations support.
Failed: Cannot create Response XML Document	Yes	The decrypted file's XML is not well-formed or the file cannot be decrypted. Either the file was not encrypted, encrypted with an invalid or unknown key, or the XML contains invalid characters or is not well-formed.
Failed invalid byte 2 of 4-byte UTF-8 Sequence	Yes	Upper ASCII characters found in the file.
Failed Process: Exception: Error accountCount is not equal to actual number of accounts	Yes	The accountsCount number does not match the number of account records entered.
Failed xmlValidation SAXException	Yes	The Response file does not match the XSD.

FIs are required to validate all responses utilizing the XSD provided prior to delivery to the e4641 SFTP server. The e4641 System will also validate all responses prior to processing and will reject any responses that fail to meet the requirements of the schema. Responses which fail validation must be retransmitted or handled manually by the FI.

Response Error	Fatal	Description
Ignored: Invalid Request State	Yes	Request ID is either non-existent or does not belong to this institution.

Response Error	Fatal	Description
Ignored: Unknown Error	Yes	The Response is N for Will Not Respond, and the reason entered is over 1000 characters.
Ignored: Previously Responded	Yes	The Response was received previously through the application, in another Batch file or duplicated in the current Batch file.
Ignored: Error Saving Accounts	Yes	The account number is over the field limit.
Ignored: Error Saving Balances	Yes	The account balance is invalid.
Ignored: Error Saving Remarks	Yes	The remarks are over the field limit.
Ignored: Invalid Dates in Balances	Yes	e.g. the year entered is invalid
Ignored: Invalid Response Element	Yes	An invalid responseType code was entered.
Ignored: Invalid Accounts Element	Yes	e.g. no account number was provided, either part or all of the owner name was not provided, date of balance/interest is not in the range requested, an account number does not have an account type entered, and invalid account type was entered.

5.4. Response File Format

Once an FI's automated response process has completed, the FI will convert/render the response data in an XML file in UTF-8 encoding.

There is no required naming convention for the response files, but Accuity recommends FIs follow Accuity's naming convention for request files, i.e. **SSA-<batch_id>_<timestamp>.aes** or **SSA-<batch_id>_<timestamp>.pgp**

Below is an example of valid response content in XML. This example illustrates the correct composition for the three different response types – Accounts Found, No Accounts Found and Will Not Respond. Though there are no labels that explicitly match these types within the XML, the presence or lack thereof of specific response elements dictate what response your FI is providing for each request. These details are provided in the following sections.

```
<?xml version="1.0" encoding="UTF-8"?>
<AVSResponses effectiveDate="20130729192357" count="4">
<response requestId="112580">
  <responseType response="Y" />
  <accounts count="0"/>
</response>
<response requestId="112581">
  <responseType response="N" code="InvalidNameForSSN"/>
  <accounts count="0"/>
</response>
```

```
<response requestId="112583">
  <responseType response="N" code="Other">FI response msg</responseType>
  <accounts count="0"/>
</response>
<response requestId="112585">
  <responseType response="Y"/>
  <accounts count="1">
    <account joint="Y" SSIDirectDeposit="N">
      <accountNumber>123456</accountNumber>
      <accountType code="CHECKING"/>
      <owners>
        <owner>
          <last>Baker</last>
          <first>Able N.</first>
        </owner>
        <owner>
          <last>Baker</last>
          <first>Jessica B.</first>
        </owner>
      </owners>
     <balances>
         <balance date="201306" balance="101.23" interest="0.00"/>
         <balance date="201307" balance="103.25" interest="0.00"/>
    </balances>
    </account>
  </accounts>
  <remarks>
     3456: Opened 05/2013
     3456: Closed 07/2013
  </remarks>
</response>
</AVSResponses>
```

5.4.1. Response File Requirements

Below are some general rules and requirements associated with the response file, most of which are enforced by the XSD and most of which have been covered elsewhere in this document:

- The format of the response file must be XML file in UTF-8 encoding.
- The response file must adhere to the requirements enforced by the XSD.
- Only one response can be provided per request in the file.
- At least one response must be contained in the response file (no empty files).

5.5. Response Types

As mentioned, an FI can respond to a request in one of three ways, which are generally defined as "Accounts Found," "No Accounts Found" and "Will Not Respond." These categories are provided solely to simplify the FI's understanding of what to respond with and how; they are not labelled explicitly in the schema. Each response type has specific elements and values that are required, which are detailed in the following sections.

5.5.1. Accounts Found

This section provides the definition and element requirements for the "Accounts Found" response type.

Description	Definition
Accounts Found	This Response Type is used when the FI has conducted a search and SSA's minimum requirements for a positive match are met: • The SSN identified on the Request is matched with the SSN of a customer in the FI's database. • The FI customer had at least one account that was open for at least one month within the "date range" specified on the Request.

Below is an example of a response with an account and balances in XML format:

```
<response requestId="112585">
  <responseType response="Y"/>
  <accounts count="1">
    <account joint="Y" SSIDirectDeposit="N">
      <accountNumber>123456</accountNumber>
      <accountType code="CHECKING"/>
       <owners>
        <owner>
          <last>Baker</last>
          <first>Able N.</first>
        </owner>
        <owner>
          <last>Baker</last>
          <first>Jessica B.</first>
        </owner>
      </owners>
     <balances>
         <balance date="201306" balance="101.23" interest="0.00"/>
         <balance date="201307" balance="103.25" interest="0.00"/>
    </balances>
    </account>
  </accounts>
  <remarks>
     3456: Opened 05/2013
```

```
      3456: Closed 07/2013
   </remarks>
</response>
```

Requirements associated with the Accounts Found Response Type:

- The responseType indicator must be response="Y."
- The accounts count must be greater than zero (>0) and equal the number of accounts provided within that response.
- The accountNumber, accountType code, owners and balances elements must be populated. At least one monthly balance that falls within the date range specified on the request must be provided.

5.5.2. No Accounts Found

This section provides the definition and element requirements for the "No Accounts Found" response type.

Description	Definition
No Accounts Found	This response type is used when ANY of the minimum requirements for a positive match are NOT met, including: • SSN does NOT match with customer in FI database. • The FI customer had NO ACCOUNTS open within the "date range" identified on the request. PLEASE NOTE: Even if a match is found on the SSN, if the customer had no account open at any point during the "date range" identified on the request, the correct response type is "No Accounts Found."

Below is an example of a No Accounts Found response in XML format:

```
<response requestId="112580">
   <responseType response="Y" />
   <accounts count="0"/>
</response>
```

Requirements associated with the Accounts Found Response Type:

- The responseType indicator must be response="Y"
- The accounts count must be zero (0)

- No other account or balance elements can be included

5.5.3. Will Not Respond

This section provides an enhanced definition of the response type of "Will Not Respond." This response type should ONLY be used when the SSN has matched, but the identity of the Customer cannot be validated ("SSN Correct, Name Incorrect"). If there is a reason other than the failed identity validation that the FI would use "Will Not Respond," then the FI would utilize the "Other" category and provide a description.

DO NOT use "Will Not Respond" when a positive match has NOT been found. The correct response for not finding an account is "No Accounts Found."

This response type is identified through the use of a specific "responseTypeCode" as indicated below.

Description	Response TypeCode
SSN Correct, Name Incorrect	InvalidNameForSSN
Other Explanation	Other

These two responseType codes are defined in further detail below:

Description	Definition
SSN Correct, Name Incorrect	This response type is used when the FI has a positive match on the SSN, but "Will Not Respond" because the name of the customer provided on the request does not match their records. FI's conducting name matches can use this response type.
Other Explanation	This response type is used when an FI has found a positive match with the request, but "Will Not Respond" for a reason other than those provided. A description of the reason should be provided by the FI if this option is used.

Below is an example of the response type "Will Not Respond" in XML format. This example uses the "Other" followed by the text description:

```
<response requestId="112583">
  <responseType response="N" code="Other">FI response msg</responseType>
  <accounts count="0"/>
</response>
```

Requirements associated with the Will Not Respond Response Type:

- The responseType indicator must be *response="N."*
- The accounts count must be zero (0).
- No other account or balance elements can be included.
- Either the InvalidNameForSSN or the Other type codes must be used when the response indicator is set to *response="N."*
- Do not use this response type when an account opened during the date range has NOT been found. The correct response type is "No Accounts Found."

5.6. Response File Elements

The XML elements are described in the following tables:

a) <AVSResponses> Element

Excerpt:

```
<AVSResponses effectiveDate="20130729192357" count="4">
```

Field	Data Type	Max Length	Required	Description
AVSResponses effectiveDate	Numeric	14	Yes	Effective date for responses (YYYYMMDDHHmmSS).
AVSResponses count	Numeric	12	Yes	Count of response elements. Must be greater than zero (>0).
response			Yes	Container for "response." See <response> Element (b).

b) <response> Element

Excerpt: (Below example for "Will Not Respond" Response Type)

```
<response requestId="112583">
  <responseType response="N" code="Other">FI response msg</responseType>
  <accounts count="0"/>
</response>
```

Field	Data Type	Max Length	Required	Description
responseRequestId	String	12	Yes	e4641 Request ID for this response.
responseType response	Character	1	Yes	Flag value (Y/N) to indicate how a response is being responded to. For Response Types "Accounts Found" and "No Accounts Found" this flag is "Y." For "Will Not Respond" this flag is "N."
responseType code	Character	20	Conditional	Will Not Respond reason code. Required for Response Type "Will Not Respond."
responseType	Character	1000	Conditional	Will Not Respond text for "Other" responseType code. Required if "Other" code type is identified.

Field	Data Type	Max Length	Required	Description
accounts			Yes	Container for account elements. See <account> Element section (C).
remarks	Character	1000		Remarks supporting response. Any additional information supporting response (i.e. account open and closed dates, etc.)

c) <account> Element

Excerpt: (Below is an example for "Accounts Found" Response Type. The "accounts" container is shown with all of its elements ("account," "owners" and "balances") included. The "remarks" element is also represented).

```
<response requestId="112585">
  <responseType response="Y"/>
  <accounts count="1">
    <account joint="Y" SSIDirectDeposit="N">
      <accountNumber>123456</accountNumber>
      <accountType code="CHECKING"/>
      <owners>
        <owner>
          <last>Baker</last>
          <first>Able N.</first>
        </owner>
        <owner>
          <last>Baker</last>
          <first>Jessica B.</first>
        </owner>
      </owners>
     <balances>
         <balance date="201306" balance="101.23" interest="0.00"/>
         <balance date="201307" balance="103.25" interest="0.00"/>
    </balances>
    </account>
  </accounts>
  <remarks>
     3456: Opened 05/2013
     3456: Closed 07/2013
  </remarks>
</response>
```

Field	Data Type	Max Length	Required	Description
accounts count	Numeric	Unlimited	Yes	Number of accounts provided for the response. Must be zero (0) for "No Accounts Found" and "Will Not Respond" response Types.
account joint	Character	1		Flag (Y/N) indicating that the account is a joint account.
Account SSIDirectDeposit	Character	1	Conditional	Ignore.
accountNumber	Character	28	Conditional	Account number. Required for each account provided.
accountType code	Character		Conditional	Code for account type (e.g. DDA, Savings). Required for each account provided. See "Account Type Code" section for codes.
accountType	Character	30	Conditional	Description for accountType. Required if "Other" account type selected.
owners			Conditional	Container for account owners. Required for each account provided.
balances			Conditional	Container for account balances. Required for each account provided.

d) <owner> Element

Excerpt: (Below example for "Accounts Found" Response Type)

```
<owner>
  <last>Baker</last>
  <first>Able N.</first>
</owner>
<owner>
  <last>Baker</last>
  <first>Jessica B.</first>
</owner>
```

Field	Data Type	Max Length	Required	Description
last	Character	100	Conditional	Account owner last name. Required for each account provided.
first	Character	100	Conditional	Account owner first name. Required for each account provided.

e) <balance> Element

Excerpt: (Below example for "Accounts Found" Response Type)

```
    <balances>
        <balance date="201306" balance="101.23" interest="0.00"/>
        <balance date="201307" balance="103.25" interest="0.00"/>
    </balances>
```

Field	Data Type	Max Length	Required	Description
balance date	Numeric	6	Conditional	Year and month (YYYYMM) for balance and interest information. Required for every balance entry. At least one monthly balance that falls within the date range specified is required for each account provided in the response.
balance value	Numeric	16	Conditional	Account balance for this month (Format #.##). Required for every balance entry.
balance interest	Numeric	16	NO	Account interest for this month (Format #.##). Interest is not mandatory.

6. ACCOUNT TYPES AND CODES

NOTE: It is not required that the FI map to all of the account types listed in the chart below. If it is only feasible, or more simple, to map all accounts to the more general categorizations, such as CHECKING, SAVINGS, CD, etc., then the FI is permitted to do so. The key requirements are that all accounts associated with an SSN are provided and that whichever account types are used by the FI, they match the Account Type Code defined in the below chart. Only one Account Type Code is permitted per account.

Description	Account Type Code
Annuity	ANNUITY
Burial/Funeral	FUNERAL
Checking Account	CHECKING
Christmas Club	XMAS_CLUB
Custodial – Other	CUSTODY_O
Custodial – Retirement Home	CUSTODY_R
IRA	IRA
Keogh	KEOGH
Money Market Account	MM
Rent Security	RENT_SECUR
Savings Account	SAVINGS
Time/Cert of Deposit	CD
Trust	TRUST
Other	OTHER

7. EXCEPTION PROCESSING

An "exception process" is a process that is manually facilitated by fax or email outside of the e4641 System as a result of an FI being unable to include the information in the Batch transmission. There are two principle causes of exception processing:

1. Specific processing scenarios that are outside the reach of their automated Batch processes, or

2. There was an error or missing information on a response processed via Batch and the only method of attaining the correct information is through a manual exception process (the "resend" option is not viable).

7.1. Known Exceptions

The exception process is considered a last-resort to correct a mistake or account for a deficit in the response data; therefore, every effort must be made by the FI to eliminate all or nearly all conditions where the exception process will be required and to adhere to the guidelines when manual intervention is unavoidable. Please note that exception processing should be very rarely required. If required, the total number of exception items must constitute less than .02% of the average daily volume (over one month) of requests for an FI.

7.2. Method Limitation

An FI can only utilize one request processing method via the e4641 System. There are no duel or parallel processing pathways that the FI can use to supplement one method or the other. Therefore, if an FI utilizes the Batch process, that FI will not have any ability to manually process requests via the e4641 Secure Internet Solution. All request processing through the e4641 System must be facilitated via the Batch process. Any response to a request that cannot be facilitated by the FI via Batch will be handled via the exception process.

7.3. Exception Process Workflow

All FIs will be made aware of and trained on the exceptions processing workflow, regardless if an FI claims they have achieved total automation. Common causes of requiring an exception process include:

- Complexities in system and data structure of FI;

- Imprecise account creation processes that require manual intervention;

- Dispersed systems due to mergers and acquisitions;

- Retired account types within an FI that are no longer a part of the core system;

- Any other reason that data cannot be retrieved automatically.

Regardless of the reason, the exception process is designed to facilitate the delivery of complete and accurate information to SSA representatives directly via fax.

7.3.1. Steps of Exception Processing

Each Batch FI follows the same general exception workflow which consists of a series of very simple steps. Once Accuity has confirmed that an exception is required, the following process occurs:

1. Accuity notifies the SSA User assigned to the request that additional information will be coming to them via the exception process. Accuity confirms the fax number of the user.

2. The FI is notified of the issue (unless they reported the issue) and provided the following information:

    a. Request ID;

    b. Date of Request;

    c. Comments/Instruction.

3. The FI completes a paper form (already provided) with ALL the information not included in the Batch transmission.

4. The FI faxes the completed form back to Accuity's dedicated fax machine.

5. Accuity performs quality control checks on the content of the form and reconciles any issues with the FI.

6. Once QC is complete, Accuity faxes the completed form with coversheet to the fax number confirmed by the SSA User.

7. Accuity sends the SSA User an acknowledgment email to confirm that the fax was received.

8. BATCH FILE TESTING PLAN

When financial institutions begin responding to Asset Verification requests via Batch processing and establish connectivity, Accuity will create test cases in the UAT environment. This process will start one to three days after connectivity is established depending on the schedule of the FIs and Data Management Group. The creation of test cases will be done using a three-step process that will be outlined below. The following test cases (UAT -1 and UAT -2 and Stress File) will be created by the Data Management Group. The next step is to send through the SFTP server of the FI in a request file which will be encrypted with an associated FI key. The timeframe to validate and QC the data once the response file is returned from the FI will be up to five business days, depending on the Data Management Group workload.

8.1. Batch Testing Steps

The following is a list of steps that all FIs preparing for Batch integration must follow:

1. UAT – 1 – Initial Test File

Create 10-15 test cases that will search for any requests where customers have "No Accounts Found" and/or account information for one account. It is considered low level difficulty scenarios. In order for an FI to pass UAT – 1, it needs to have 70% of the requests correct.

Scenarios:

- "No Accounts Found" – SSN that are not customers of the financial institution
- SSN customers who hold one account with the financial institution
- SSN customer with a joint account

  (Joint Account = Y AND Direct Deposit = N)

  (Joint Account = N AND Direct Deposit = N)
- SSN customers with an alleged account with the financial institution

2. UAT – 2 – Second Test File

Create 15-30 test cases that will search for any requests where customers have "No Accounts Found" and account information for two or more accounts. It is considered medium level difficulty scenarios. In order for an FI to pass UAT – 2, it needs to have 80% of the Requests correct.

Scenarios:

- No Accounts Found

- Multiple response types (X = Will Not Respond)
- Customers with two or more accounts
- Customers with Rep Payee/Beneficiary accounts
- Joint Accounts (Joint Account = Y)
- SSI Direct Deposit Accounts (Direct Deposit = Y)
- Requests that require interests

3. Stress File

Create 300+ test cases that will search for any requests where customers have "No Accounts Found" and account information of one or more accounts. It is considered a stress test in order to assure the financial institution's ability to handle a large quantity of requests. In order for an FI to pass the Stress File, it needs to have 100% of the requests correct.

Scenarios:

- No Accounts Found
- Customers with two or more accounts
- Customers with Rep Payee/Beneficiary accounts
- Joint Accounts (showing all names on the account)
- SSI Direct Deposit Accounts
- Customers with accounts opened and or closed within the dates requested
- Requests that require interests
- Accounts with balances considered high level, over $2,000