Exhibit B
Social Security Verification (SSV)

System Specification

Release 2.0.0

August 2004
The American Association of Motor Vehicle Administrators (AAMVA) produced this document.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage or retrieval systems, for any purpose other than the intended use by AAMVA, without the express written permission of AAMVA.

©2004 AAMVA All rights reserved.
Table Of Contents

1 INTRODUCTION ........................................................................................................... 5
  1.1 Document Objective .............................................................................................. 5
  1.2 Getting Help (1-888-AAMVA-80) ......................................................................... 5

2 APPLICATION DESCRIPTION .................................................................................. 6
  2.1 Social Security On-line Verification (SSOLV) ....................................................... 6
  2.2 Help America Vote Verification (HAVV) ............................................................. 6

3 IMPLEMENTATION PLANNING ............................................................................... 7
  3.1 Requirements Definition ....................................................................................... 7
  3.2 Design ..................................................................................................................... 7
  3.3 Development ......................................................................................................... 7
  3.4 Testing ...................................................................................................................... 8
    3.4.1 Internal Tests ..................................................................................................... 8
    3.4.2 Casual Testing ................................................................................................ 8
    3.4.3 Structured Testing ........................................................................................... 8
  3.5 Production Implementation .................................................................................... 9
  3.6 SSV Help Desk ...................................................................................................... 9
  3.7 Agreements ........................................................................................................... 10

4 SYSTEM ARCHITECTURE ...................................................................................... 11
  4.1 Basic SSV Configuration ....................................................................................... 11
  4.2 Teleprocessing Monitor ....................................................................................... 11
  4.3 Network Control Software (NCS) ......................................................................... 11
  4.4 AAMVAm Message Interchange Envelope (AMIE) Structure ............................... 12
    4.4.1 Background .................................................................................................... 12
    4.4.2 Message Addressing ...................................................................................... 13
    4.4.3 General Rules for AMIE Message Composition ............................................ 14
    4.4.4 Application Text Blocks ............................................................................... 15
    4.4.5 Message Format of Fields ............................................................................ 16
  4.5 Error Handling Specifications .............................................................................. 16
    4.5.1 Network Errors .............................................................................................. 17
    4.5.2 System Errors ................................................................................................ 17
    4.5.3 Processing Errors ........................................................................................... 18
  4.6 Application Layer Network Interface Software .................................................... 18
    4.6.1 AAMVA's Unified Network Interface (UNI) ................................................... 18
    4.6.2 Message Retry .............................................................................................. 18
    4.6.3 Hard Manual Down ...................................................................................... 19
    4.6.4 Message Locator ........................................................................................... 19
    4.6.5 Call List ......................................................................................................... 20
    4.6.6 Driver Call List Layout ................................................................................ 20
    4.6.7 UNI Platforms Supported ............................................................................ 20

5 SSV TRANSACTIONS .............................................................................................. 21
  5.1 Social Security On-line Verification (SSOLV) Transaction .................................... 21
    5.1.1 'Ss' - SSA Verification Request Message ..................................................... 21
    5.1.2 'Ths' - SSA Verification Response Message ................................................. 23
  5.2 Help America Vote Verification Transaction ..................................................... 25

Social Security On-Line Verification - Release 2.0.0 - August 2004

25
1 INTRODUCTION

1.1 Document Objective

The purpose of this document is to describe the data flows and transactions in the Social Security Verification application. The document is written for the Social Security Administration (SSA) and Jurisdictions who must develop Social Security On-line Verification (SSOLV) and Help America Vote Act Verification (HAVV) systems. The document contains the information necessary for a software application development team to:

- Write an implementation plan;
- Determine requirements for their application, based on nationwide requirements;
- Construct a framework for the design of their system implementation.

Because the requirements are written at a high level, a detailed implementation specification should be produced by the developers to describe how the system will be implemented in their environment.

The original release of this document focused on SSOLV implementation at SSA. This release also contains information for users who make SSOLV inquiries to SSA and a full description of the HAVV transaction.

1.2 Getting Help (1-888-AAMVA-80)

Questions regarding this document or the application itself should be directed to:

Operations Department:
Hours: 8:00 a.m.-6:00 p.m. Eastern Time
Telephone: 1-888-AAMVA-80
Fax: (703) 522-1533
Address: AAMVA, Inc.
4301 Wilson Boulevard, Suite 400
Arlington, Virginia 22203
Website: www.aamva.org
e-mail: opsdept@aamva.org
2 APPLICATION DESCRIPTION

The Social Security Verification application (SSV) consists of two transactions: the Social Security On-line Verification (SSOLV) and the Help America Vote Verification (HAVV). Each transaction contains two messages: an inquiry message sent to the Social Security Administration (SSA) and a response message returned by the SSA.

2.1 Social Security On-line Verification (SSOLV)

Driver licenses and identification cards issued by US Motor Vehicle Agencies (MVA’s) have become the U.S. standard for identification. In order to curb the fraudulent issuance of driver license and identification cards, the MVA’s carefully review documentation that is presented to them to verify the identity of the individual. The Social Security Card is one form of identification that is reviewed. In most jurisdictions, the Social Security Number (SSN) is also used as the standard to uniquely identify individuals on the licensing records.

To minimize the fraudulent issuance of the driver license or identification card, the MVA’s need a way to verify the information contained on the card is valid. The SSN verification needs to be performed on-line while the applicant is still at the MVA counter but prior to the issuance of the driver license or identification card.

The Social Security On-line Verification (SSOLV) transaction has been developed to allow authorized MVA’s to have on-line access to SSA for SSN verification. Using this transaction, a MVA electronically sends SSA a person’s name, date of birth (DOB) and SSN. SSA then compares this data to what is on its Master File. SSA will then respond back to the inquiring MVA, indicating how much of the MVA-submitted data matched against the SSA file.

2.2 Help America Vote Verification (HAVV)

Section 305 of Public Law 107-252 (Help America Vote Act of 2002) requires States and localities to develop centralized, computerized voter databases and to verify voter registration information. Individuals registering to vote must provide their driver’s license number to the State election agency. If the registrant has no driver’s license, they must supply the last four digits of their SSN. The statute requires that the chief State election official and the officials responsible for the State motor vehicle authorities to enter into agreements to match voter registration information with MVA information. The statute further requires the MVA officials and the Commissioner of Social Security to reach agreements for the purpose of verifying name, date of birth, the last four digits of the SSN, and any information recorded in SSA’s records about the death of an individual.

The Help America Vote Verification (HAVV) transaction allows a MVA to submit an inquiry to SSA. The SSA verifies the information and responds back to the MVA with the results.
The Primary Address is the 2-character code for jurisdictions and AAMVA processing sites (normally the postal abbreviation for jurisdictions), and the Interface Code is used to distinguish between multiple systems at a single site. For example, in some states the driver licensing and vehicle registration systems are operated on different physical machines. The Interface Codes would be different for each.

AAMVA manages the overall AIME UserID system, and is therefore responsible for assigning all values as necessary for the GAP Code.

The User Extension field of the Primary User description can be used at the discretion of the users, within the normal parameters for AIME Messages (See the section on General Rules for AIME Message Composition). This field is frequently used to identify a particular workstation that originated the message and therefore should receive the response. Other uses are possible depending on the needs of the users. Usage of this field should be limited to the Transaction Originator because it is the pass-through field.

4.4.3 General Rules for AIME Message Composition

Data in an AIME message may consist of any printable character. This means that non-printable bytes are not allowed in any AIME message. This limitation has been imposed due to the architecture of the AAMVA Net network, which consists of many different types of computers on the network, each possibly having a different data-encoding scheme.

For example, the AT&T NETWORK SERVICES and its mainframes store character data in EBCDIC, while Unisys, Bull, and most other computer types store character data in ASCII. Translation between these code sets is performed as part of the network transmission to or from an ASCII based machine. The translation occurs by replacing a bit pattern from one code set with the corresponding bit pattern from the other code set. As the translation is performed to each byte of data traveling on the data path without regard to the content of the data, non-printable data would be corrupted when the bit patterns were replaced as if the byte contained character data.

Translation adulteration aside, each different machine type stores computational numeric data in a format native to the processor. Assuming numeric data could move between AAMVA nodes without adulteration, the data would probably be unusable by the destination node unless the origination and destination nodes happen to be compatible machine types.

For example, floating point decimal data on an AT&T NETWORK SERVICES mainframe is stored in a specific pattern of bits within two, four, or eight bytes, depending on the resolution required. Elements such as the exponent and mantissa are assigned to certain bits and are represented in defined ways. The same number on a VAX machine is stored with a different bit pattern, different exponent bases, and different byte order. Moving a floating-point decimal data item from an AT&T NETWORK SERVICES platform to a VAX would not yield usable data on the VAX. The reverse is also true.
Eventually, exceptions to this rule may be required to allow movement of complex data in an efficient manner, possibly using encoding and compression schemes. At that time specific exceptions will be defined and will be documented to an extent that potentially affected users will be aware of their limitations. However, the general rule will still apply to all other messages that may be sent between nodes running on different computer types.

To ensure only printable bytes exist in a message, you must initialize all unused areas of each block with spaces. This ensures that un-addressable areas, such as the reserved bytes at the end of most, contain valid AMIE data. The unused fields should also be initialized to spaces regardless of the data type of the field. For example, a date field is normally numeric, yet if the field is not a valid part of the message being built, the field should contain spaces rather than zeroes. Do not initialize AMIE blocks or fields to LOW-VALUES or HIGH-VALUES, as these are binary zeroes or ones, respectively, and do not represent printable data.

All application data elements must contain printable characters that can be used in both ASCII and common versions of EBCDIC. The printable characters are:

```plaintext
space
a to z
A to Z
0 to 9
! " # $ % & ( ) * + , - . / ; : ; < = > ? ^`
```

Other characters are not printable in ASCII and US-EBCDIC, so should be excluded. The user will need to determine if the non-printable characters will be omitted or if they will substitute in another character. The recommendation for the Spanish 'í' and 'ú', is to convert the character to 'n' and 'u' before sending the data.

4.4.4 Application Text Blocks

For this system, the text block pool of an AMIE message contains the following block types:

- Message Exchange Control block (02/2). One Message Exchange Control (MEC) block will be present on each message. See the Message Exchange Control Block section for details.
- Business Application blocks (09/1, 10/1).
- Return-as-received blocks (98/3). Zero to five return-as-received blocks may be used, and they are used by the transaction originator.
- Error blocks (99/1). Zero to five error blocks are used, depending on the number of errors detected. See the Error Handling Section for details.

Because the blocks are sent in the Type/Sub-type number order, the text blocks will be sent in the order shown above.

Most blocks are used once within a message. However, instances exist where an AMIE text block is used multiple times within a message. These multiple repetitions exist when:
• A field is too long to fit in a single 61-byte block. A 108-byte address is transmitted in two AMIE text blocks. The first 61 bytes are sent in the first block and the final 47 bytes are sent in the second block.

• The application data is needed multiple times, where a single occurrence of the data will fit onto one block. The number of blocks will correspond to the number of occurrences of the data. The data is needed multiple times; however, the total length of the data to be repeated exceeds one block. In these situations, the number of AMIE text blocks used is the product of the number of blocks used to hold a single occurrence, times the number of occurrences.

To be unique the Text Block Key will use an incremented line number to distinguish between the multiple occurrences of block types and maintain the sent sequence.

4.4.5 Message Format of Fields

All dates sent in the application specific blocks of the messages are passed as eight character fields in 'ocyyyyymmd' form, (e.g., '19951231'). All numbers sent in a message are passed in an unpacked form with leading zeros (e.g., a field with 6 integer digits with a value of '1,234', is transmitted as '001234', in an alpha numeric field).

For elements that require specific values (such as codes), the fields transmitted must contain the standard values, as defined in the data dictionary.

4.5 Error Handling Specifications

The error handling procedure describes a convention by which every message error will be processed, both by the entity that detected the error and the entity that originated the message. The errors can be categorized as follows:

• network errors;
• system errors, such as program aborts, files off line, or similar conditions;
• processing errors which are caused by faulty application data in the message.

When an error is detected, the message that encountered or contained the error is returned to the sender. There are several flags and fields in the message structure that can convey information regarding errors or unusual circumstances. Depending on the severity of the problem, different combinations of the error flags/fields are used. Information can be found in the following areas:

GNCBKR - NCB ERROR CODE
Set to "Y" (yes)

GNETST - NETWORK STATUS
Set to a value other than zero, that describes the error.
GAPPST - APPLICATION STATUS
Set to a value other than space or zero, that describes the error.

GERUEC - UNI ERROR CODE or
GERCD0 - ERROR CODE
Set to a value other than space, that describes an error.

GERMSO - ERROR MESSAGE DESCRIPTION
A 54 character text field containing the description of the error.

4.5.1 Network Errors

Network errors occur when the origination or destination entity drops from the network or the network itself encounters a failure. There are established availability requirements that minimize occurrences of this nature, but occasionally a failure occurs.

When the originating entity is not connected or the network is completely down, the error is normally detectable and the message can be set aside for later transmission. The Unified Network Interface (UNI) provides this service.

If the destination node is down, the network (NCS) will return the message to the originator with an indication of the error (NCB error code = 'U' for Undeliverable) and the message can be set aside for later transmission. If the destination application is down, UNI can detect the error, notify the originator, and set aside the message for later transmission.

4.5.2 System Errors

In this application system errors may be reported in one of two ways:

- Generic system errors
- SSA file off-line

A generic system error is an error with the system itself, such as program problems, network interface errors, database errors, program aborts, etc. To the extent possible, message recipients should try to detect these conditions and return the original message with the appropriate indicators to inform the originator of the problem (NCB error code = 'Y', processing status = '01', error block attached indicating the error and application status set to appropriate code, if applicable).

The more common system error occurs when the SSA file is off-line. In this instance, the SSA application will return the SSA Verification Response (RS) message with the SSA Verification Response Code set to '9'. Other system errors detected within the SSA application will also be reported with the Response Code set to '9' on the Verification Response message.
4.5.3 Processing Errors

The SSA will not edit data received in the incoming Verification Request (SS) nor will it return corrected information. Therefore, the only error a SOI should encounter would be that of a network or system error.

4.6 Application Layer Network Interface Software

The Application Layer Network Interface Software (ALNIS) is generically defined as a software application residing on the host computer. The main function is the translation between the AMIE message structure and a data element and the message structure used by the application. The application data structure is provided in COBOL and C formats. It also provides a variety of other application interface support features. The interface between the application and the ANLIS is usually platform dependent. An example of ALNIS software is AAMVA’s Unified Network Interface (UNI) software package.

4.6.1 AAMVA’s Unified Network Interface (UNI)

Unified Network Interface (UNI) provides critical services for jurisdictions’ applications. The UNI was developed by AAMVA for its customers running applications requiring data transfer in the AAMVANet Message Interchange Envelope (AMIE) electronic data interchange (EDI) format. Although using AAMVA’s network interface tool is not a requirement, most users will choose to implement the system using the Unified Network Interface (UNI). UNI has several valuable functions available to assist users (such as message control, routing validation, logging, audit trails, and message grouping). A jurisdiction’s network interface team needs to understand UNI’s functions to avoid duplicating those functions within the application.

The purpose of this section is to supplement the UNI documentation by calling attention to several UNI features that have been found particularly useful. Although they are documented in the UNI Application Developer’s Reference, we have included a brief synopsis here along with suggested settings, where applicable.

4.6.2 Message Retry

AAMVA recommends that users configure the parameter list of all on-line update messages to attempt up to three retries in the event the messages are undeliverable. When set, UNI retry is performed automatically. Users should keep in mind that automatic retry may not be appropriate for messages where the state prefers to control retries either manually or programmatically through the application (as may be the case with inquiry messages).

The EARM-CNT-RETRY-MAX field in the UNI parameter list controls the maximum number of times that UNI will attempt to send an outbound message to its destination. This is a 1-digit numeric field, so valid values range from 0 to 9.
If the number of retries is set to ‘0’ and the outbound message is returned as undeliverable, UNI will not retry the message. If the number of retries is set to a non-zero value, UNI will hold the message in its undeliverable message file until such time as UNI determines that the destination’s node or application is again available. UNI actively checks the status of retry destinations and does not attempt a retry until a positive status is attained. UNI checks the status of all other nodes on the network by issuing IN messages at regular intervals and interrogating the RN responses. The default interval is 20 minutes, but this is configurable. UNI will attempt to resend until it has exhausted the maximum number of retries designated.

4.6.3 Hard Manual Down

A hard manual down causes UNI to treat a destination node as though it were down even when it is not. This can be used, for example, when a state must store on-line transactions while its load file is being processed. Issuing a hard manual down on the destination node causes on-line transactions to that node to go to the message pending process given message retry is configured. Transactions will continue to queue up in message pending until the hard manual down is manually removed. As stated earlier, it is very important to pace messages being released from message pending.

Hard manual downs are issued from the UTT200 Network/Application Status screen by adding the site ID of the destination to be downed to the application status list. First, enter an action code of ‘A’, the network ID of the destination, and an application code of ‘11’. The down reason will be set to ‘soft manual’ by the system. To change the down reason to ‘hard manual’, enter an action code of ‘M’. The ‘M’ action code toggles between a soft and a hard manual down. To delete a hard manual down, enter an action code of ‘D’. Message pending will initiate release of messages at the next IN/RN interval.

Before issuing a hard manual down, states should estimate the amount of space needed to store the message pending file. Steps should be taken to ensure that enough space will be available to hold the estimated number of pending messages.

4.6.4 Message Locator

When a transaction is initiated, UNI generates a unique identifier for the message called a message locator. UNI uses the message locator to match messages with their responses. When contacting the AAMVA Operations help desk for support, it is important that you provide the message locator. The message locator provides a means for the AAMVA Operations help desk to find the specific message or messages causing the problem.

The message locator is found in the first 26 bytes of the MEC block. It is comprised of a date/time/sequence number along with the message type.

A sample message locator and its components are shown below:

000502132312001    1UNISS
where:
't000502' is the date
't132312' is the time
't0001' is the sequence number
' ' is a constant
't' is the occurrence of the destination in the PARM-DESC-TABLE-DEST of the parameter list
'tUNI' is a constant
'tSS' is the message type

4.6.5 Call List

UNI provides a parameter list and call list to interface between the jurisdiction's application and the network. The call list data is converted to the AMIE structure before it is sent to network and vice-versa. The parameter list provides a means for matching response messages to inquiry messages, routing messages and store and forward features. The parameter and call lists use a flat file format which makes it easy for developers to address the elements.

4.6.6 Driver Call List Layout

In the Driver Call List, there is a record type indicator (CLMF-DESC-RECORD-TYPE) that is populated by UNI when a message is received. This indicator is used to identify how much of the variable length Call List is being used. In this application UNI sets the indicator to "R", "L" or "S". When the indicator contains:
"L" the type of record is a long record. In this situation the address is included.
"S" the type of record is a short record. In this situation no address is included.
"R" the type of record is a return as received.
So before addressing elements residing in the extended part of the call list, check the record type indicator to ensure a long call list has been delivered.

4.6.7 UNI Platforms Supported

AAMVA's web site (www.aamva.org) has a complete up-to-date listing of supported platforms.
5 SSV TRANSACTIONS

5.1 Social Security On-line Verification (SSOLV) Transaction

Purpose: The SSOLV Transaction is used by an authorized MVA (End User) to request the verification of an SSN provided by an applicant or that is found on the MVA’s database to aid in the prevention of fraudulent identification issuance.

Transaction Message Flow Diagram

1. The MVA (End User) formats the request into the AMIE format and forwards it to the SSA through the AAMVAnet network.

2. SSA receives the request and responds to the State of Inquiry (SOI) with the verification data in the AMIE format.

Note: For detailed information on the message formats, the AMIE blocks and the data elements, refer to Appendices A, B, C and D.

5.1.1 'SS' - SSA Verification Request Message

5.1.1.1 State of Inquiry (SOI) Processing Requirements:

The SOI must provide the following data elements to successfully process the SSA Verification Request (SS):
- Social Security Number (DDVSSN) Required
- Driver Name (DDVNMA) Required (See the SSA Name Formatting Rules in the Appendix)
- Driver Date of Birth (DDVDDB) Required

In addition the SOI may include the following elements:
- Jurisdiction (DDIJU1) Optional

Social Security On-Line Verification - Release 2.0.0 - August 2004
• Driver License Number (DLDNUM) Optional
• Return as Received (GRREC2) Optional

NOTE: Do not attempt to verify SSNs allocated by user applications (e.g. the CDSLIS substitute and pseudo-SSN), because the SSA will always respond that such SSNs are invalid.

5.1.1.2 Social Security Administration (SSA) Processing Requirements:

Upon receiving the SSA Verification Request (SS), the Social Security Administration (SSA) will search for the requested record in its database.

NOTE: The SSA will not edit or check for errors in the SS message, it only verifies the data present.

5.1.1.2.1 SSOLV Name Match Criteria

A name (see the SSA Name Formatting Rules in the Appendix) provided by the MVA will be accepted as verified, if a match is made using any of the following criteria:

1. If the first seven positions of the surname (e.g.: last name) and the first and middle name initials match exactly.

2. If only one initial is provided, the first seven positions of the surname must match and the initial provided will match the first initial of either the first or middle name.

3. The first four positions of the input first name and the first four positions of the file first name match.

4. If no first name is provided, the first four positions of the surname and the first and middle name initials must match.

5. A one letter difference or transposition of two adjacent letters in the first seven positions of the surname and the first and middle name initials match exactly (AB=AB) or are transposed (AB=BA).

6. A one letter difference or transposition of two adjacent letters in the first seven positions of the surname and

   a) the first or middle initial of the MVA name match that of the first name initial of the SSA name when only one initial is present on SSA files (AB=A or BA=A); or
   b) the first initial of the MVA name matches the first or middle initial of the SSA name when only one initial is present on the MVA record (A=AB; A=BA; B=BA; B=AB); or
   c) the MVA first name initial matches the SSA first name initial and the MVA middle name initial disagrees with the SSA middle name initial, but matches the first initial of another
surname for a female (AB SM@TH = AG SMITH X REF - Brown, sex = female, i.e. a maiden name check).

7. An extraneous or missing letter is present in the first seven positions of the MVA surname and the MVA first name initial matches the SSA first or middle name initial.

<table>
<thead>
<tr>
<th>Extraneous Letter</th>
<th>Missing Letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Johnso</td>
<td>A Johnson</td>
</tr>
<tr>
<td>A Johnso</td>
<td>A Johnson</td>
</tr>
<tr>
<td>A Jsohnso</td>
<td>A Johnson</td>
</tr>
<tr>
<td>B Johnso</td>
<td>B Johnson</td>
</tr>
<tr>
<td>B Jsohnso</td>
<td>B Johnson</td>
</tr>
</tbody>
</table>

8. A compound surname may only be verified using one surname. If the single MVA surname contains more than three letters SSA will compare it to up to 13 positions of the SSA name. SSA will compare positions 1-7, then 2-8, then 3-9, then 4-10, then 5-11, then 6-12, and finally 7-13. If a match occurs on any one of these comparisons, the compound surname will be verified.

5.1.1.2.2 SSOLV Date of Birth (DOB) Match Criteria

A DOB will be verified if it matches the SSA DOB using the following criteria:

1. The year of birth on the MVA record matches the year of birth on the SSA record exactly. The day and month are ignored.

2. The year of birth on the MVA record differs from the SSA DOB +/- one year and the month on the MVA record matches the SSA month.

5.1.1.2.3 SSOLV SSN Match Criteria

The SSN sent on the verification request will only be reported as verified if it matches the SSN found on the SSA record exactly.

5.1.2 'HS' - SSA Verification Response Message

5.1.2.1 Social Security Administration (SSA) Processing Requirements:

After checking for a record in its database, the SSA will send the SSA Verification Response (HS) message to the SOI with the SSA Verification Response Code (GMSVRC) in the MEC block.
The following is a list of SSA Verification Response Codes returned and a description of their meaning:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SSN, Name and DOB verified</td>
</tr>
<tr>
<td>2</td>
<td>Invalid SSN</td>
</tr>
<tr>
<td>3</td>
<td>Name did not verify, DOB is valid</td>
</tr>
<tr>
<td>4</td>
<td>DOB did not verify, Name is valid</td>
</tr>
<tr>
<td>5</td>
<td>Name and DOB did not verify</td>
</tr>
<tr>
<td>6</td>
<td>Unable to process request - go to the local Social Security office for more information</td>
</tr>
<tr>
<td>9</td>
<td>System Error. Unable to process at this time</td>
</tr>
</tbody>
</table>

5.1.2.2 **State of Inquiry (SOI) Processing Requirements:**

The SOI should examine the SSA Verification Response Code (GMSVRC) on the HS message.

If the driver identification information is verified and the applicant has nothing on his/her record to prevent the issuance/renewal of a license the application/renewal may be processed.

If the information provided by the applicant does not verify, the MVA will utilize jurisdiction specific procedures for handling the applicant.
5.2 Help America Vote Verification Transaction

Purpose: The HAVV transaction is used by an authorized MVA (End User) to request the verification of a name, date of birth and a partial SSN (last four digits) provided by an applicant to aid in the prevention of fraudulent voter registration.

Transaction Message Flow Diagram

1. The State of Inquiry (MVA) formats the request into the AMIB format and forwards it to the SSA via the AAMVA.net network.

2. The SSA receives the request and responds to the State of Inquiry (SOI) with the verification data in the AMIB format.

NOTE: For detailed information on the message format, the AMIB blocks and the data elements used in HAVV, refer to Appendices A, B, C and D. For detailed information on interfacing your application to AAMVA.net, refer to the Unified Network Interface Application Developers Reference Manual (available through the AAMVA Operations Department).

5.2.1 'IH' - HAVA Verification Request Message

5.2.1.1 State of Inquiry (SOI) Processing Requirements:

The SOI must provide the following data elements to successfully process the HAVA Verification Request (IH):

- Last four digits of SSN (DDVSLF) Required
- Name (DDVNM4) Required (See the SSA Name Formatting Rules in the Appendix)
- Date of Birth (DDVDOB) Required

In addition the SOI may include the following elements:

- Return as Received (CRREC2) Optional
This data will be validated by SSA, the edits performed are shown in the next section. Jurisdictions should ensure they do not send data that will fail these edits.

5.2.1.2 Social Security Administration (SSA) Processing Requirements:

Upon receiving the HAVA Verification Request (IH), the SSA will validate the contents of the message. If any of the following edit rules fail, the response will have the SSA Verification Response Code (GMSVRC) set to 'S', indicating "Invalid Data".

- The last four digits of the SSN must be a number in the range "0001" to "9999".
- The Date of Birth must be a valid date (though the day of birth is not used).
- The First Name must have:
  - Then in positions 2 through 15: A-Z, a single embedded hyphen, apostrophe or space.
  - Last character must be A-Z, an apostrophe or a space; unless the 14th position is A-Z, then the 15th position can be a hyphen, apostrophe, space or an alphabetic character.
  - Consecutive embedded combinations of spaces, hyphen, and/or apostrophe are not permitted.
- Last Name must have:
  - Acceptable characters for position 2 through 20 are A-Z, a single embedded hyphen, apostrophe or space.
  - Last character must be A-Z, an apostrophe or a space; unless the 19th position is A-Z, then the 20th position can be a hyphen, apostrophe, space or an alphabetic character.
  - Consecutive embedded combinations of spaces, hyphen, and/or apostrophe are not permitted.

The name in the message will be in a packed form (see the SSA Name Formatting Rules in the Appendix for details).

Valid messages are then checked against the SSA database using the following elements:

<table>
<thead>
<tr>
<th>Input</th>
<th>Match Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Name</td>
<td>Exact</td>
</tr>
<tr>
<td>First Name</td>
<td>Exact</td>
</tr>
<tr>
<td>Middle Initial</td>
<td>Ignore</td>
</tr>
<tr>
<td>Date Of Birth</td>
<td>Month and year must be exact. Ignore day.</td>
</tr>
<tr>
<td>Last four digits of the SSN</td>
<td>Exact</td>
</tr>
</tbody>
</table>

5.2.2 'RH' - HAVA Verification Response Message
5.2.2.1 Social Security Administration (SSA) Processing Requirements:

After checking for a record in its database, the SSA will send the HAVA Verification Response (RH) message to the SOI with the Response Code in the MBC block.

The following is a list of the SSA Verification Response Codes (GMSVRC) returned for HAVV and a description of their meaning:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Invalid input data</td>
</tr>
<tr>
<td>T</td>
<td>Multiple matches – all deceased</td>
</tr>
<tr>
<td>V</td>
<td>Multiple matches – all alive</td>
</tr>
<tr>
<td>W</td>
<td>Multiple matches – at least one alive ( &amp; at least one deceased)</td>
</tr>
<tr>
<td>X</td>
<td>Single match – alive</td>
</tr>
<tr>
<td>Y</td>
<td>Single match – deceased</td>
</tr>
<tr>
<td>Z</td>
<td>No match found</td>
</tr>
<tr>
<td>9</td>
<td>System Error. Unable to process at this time</td>
</tr>
</tbody>
</table>

5.2.2.2 State of Inquiry (SOI) Processing Requirements:

The SOI should examine the SSA Verification Response Codes (GMSVRC) on the RH message.

If the information provided by the applicant does not verify, the MVA will utilize jurisdiction-specific procedures for handling the applicant.