Enrollment System (ES)

Messaging

Technical Specification

**

Department of Veterans Affairs

Office of Information and Technology (OIT)

Product Development

Documentation Version 1.6

Revision History

| Date | Revision | Description | Author |
| --- | --- | --- | --- |
| 01/22/2019 | 1.6 | Added Section for FDD from HCA | Joseph Cortes |
| 01/14/2019 | 1.5 | Added Section for Preferred Name | Joseph Cortes |
| 10/23/2018 | 1.4 | Added Section for MOH Award Date in ES | Joseph Cortes |
| 08/27/2018 | 1.3 | Updated test scenario for association | Wen Lin |
| 08/23/2018 | 1.2 | Updated Z05 Message with how to test details including HL7 segments and data examples | Wen Lin |
| 08/22/2018 | 1.1 | Added Section for Early Separation Reason Code in ZMH Message Segment | Joseph Cortes |
| 08/10/2018 | 1.0 | Initial document | Wen Lin |

Table of Contents

[1. Introduction 4](#_Toc536022821)

[1.1. Purpose 4](#_Toc536022822)

[1.2. Scope 4](#_Toc536022823)

[2. Modify Rules for Z05 Message 5](#_Toc536022824)

[2.1. Change Request 5](#_Toc536022825)

[2.2. Rule Changes 5](#_Toc536022826)

[2.3. Code Changes 15](#_Toc536022827)

[3. Add Early Separation Reason Code to ZMH Message Segment 18](#_Toc536022828)

[3.1. Change Request 18](#_Toc536022829)

[3.2. Code Changes 18](#_Toc536022830)

[4. Add MOH Award Date in ES 20](#_Toc536022831)

[4.1. Change Request 20](#_Toc536022832)

[4.2. Database Changes 20](#_Toc536022833)

[4.3. Code Changes 20](#_Toc536022834)

[4.4. User Interface Changes 22](#_Toc536022835)

[5. Preferred Name with MVI as Source 24](#_Toc536022836)

[5.1. Change Request 24](#_Toc536022837)

[5.2. Code Changes 24](#_Toc536022838)

[5.3. User Interface Changes 25](#_Toc536022839)

[6. Future Discharge Date from HCA 28](#_Toc536022840)

[6.1. Change Request 28](#_Toc536022841)

[6.2. Code Changes 28](#_Toc536022842)

[6.3. User Interface Changes 29](#_Toc536022843)

[6.4. Rule Changes 30](#_Toc536022844)

# Introduction

## Purpose

The purpose of this document is to discuss changes to system components and processes related to Messaging.

## Scope

Technical specification describes the changes made to extend or modify existing messaging functionality. Each entry should include the relevant change request, as well as any code, UI and required database changes.

# Modify Rules for Z05 Message

## Change Request

**CR 768318** business requirements describe the changes to the Enrollment System that allow sending of Z05 Demographic Data Transmission messages to VistA sites without filtering. All rules that could prevent a Z05 message from being sent to will be removed so that a Z05 message is sent to all sites of record when there is a change to any demographic field transmitted on the message.

## Rule Changes

This will involve modifications in processAddress and processAssociation rule flows. The processAddress rule flow is invoked when an address/contact update is made because of a Z07 message receipt from a VAMC site. After the Z07 message receipt, developers want to modify the processAddress rule flow so that it transmits the updated address Z05 to all VAMC sites including the sending facility that sent the Z07.

ProcessAddress Rule Flow.

processAddress rule flow is invoked when the address/contact update is made as a result of receipt of a Z07 message from a site then we want to modify it so that it transmits this address update Z05 to all the sites including the sending facility that sent the Z07.

Figure : ProcessAddress Rule Flow

rule IsPermAddressHasBadAddressReason

IsPermAddressHasBadAddressReason Rule.

How to test: process a Z07 message adding a bad address reason to an existing permeant address and the address update date is more recent.

Figure : IsPermAddressHasBadAddressReason Rule

How to test: process a Z07 message adding a bad address reason to an existing permeant address; and the address update date is more recent.

ORUZ07 Message – PID Segment

Found a Veteran with a permanent address having no bad address reason

Updated the existing permanent address with bad address reason

Updated RF1 Segment with SAD reference the more recent update date

**PID Segment Example:**

PID^1^1008589404V203273^1008589404V203273~~~USVHA&&0363~NI~VA FACILITY ID&200M&L|796781315~~~USSSA&&0363~SS~VA FACILITY ID&608&L|7171927~~~USVHA&&0363~PI~VA FACILITY ID&608&L|~~~USVBA&&0363~PN~VA FACILITY ID&608&L^^BLSTCDCOKRP~BFSTCDCOKRP~BATCH~^^19760118^F^^1002-5-SLF~~005~1002-5~~CDC|2106-3-SLF~~005~2106-3~~CDC^5555 LEGACY DRIVE~#APT 555 ~PLANO~TX~75024~USA~VAB1~""~085|~~DELMAR~NY~~~N^085^(972)334-1234~ORN~CP|(214)332-1234~PRN~PH|~NET~INTERNET~PII^^^M^^^796781315^^^2135-2-SLF~~0189~2135-2~~CDC^^""

**RF1 Segment Example:**

RF1^^^SAD^^^509~USVAMC^20180816^^^

| **Bad Address Type** | **Description** |
| --- | --- |
| VAB1 | VA - Bad Address, Undeliverable |
| VAB2 | VA Bad Address, Homeless |
| VAB3 | VA Bad Address, Other |
| VAB4 | VA Bad Address, Address Not Found |

rule isPermAddressUploadBadAddressReason

isPermAddressUploadBadAddressReason Rule.

How to test: process a Z07 message adding a bad address reason to an existing permanent address and the address update date is equal to existing record.

Figure : isPermAddressUploadBadAddressReason Rule

How to test: process a Z07 message adding a bad address reason to an existing permanent address; and the address update date is equal to existing record.

ORUZ07 Message – PID Segment

Found a Veteran with a permanent address having no bad address reason

Updated address data and added a bad address reason

Updated RF1 Segment with SAD reference the same address update date on record

**PID Segment Example:**

PID^1^1008589404V203273^1008589404V203273~~~USVHA&&0363~NI~VA FACILITY ID&200M&L|796781315~~~USSSA&&0363~SS~VA FACILITY ID&608&L|7171927~~~USVHA&&0363~PI~VA FACILITY ID&608&L|~~~USVBA&&0363~PN~VA FACILITY ID&608&L^^BLSTCDCOKRP~BFSTCDCOKRP~BATCH~^^19760118^F^^1002-5-SLF~~005~1002-5~~CDC|2106-3-SLF~~005~2106-3~~CDC^5789 LEGACY DRIVE~#APT 555 ~PLANO~TX~75024~USA~VAB1~""~085|~~DELMAR~NY~~~N^085^(972)334-1234~ORN~CP|(214)[332-1234~PRN~PH|~NET~INTERNET~PII^^^M^^^796781315^^^2135-2-SLF~~0189~2135-2~~CDC^^](mailto:332-1234~PRN~PH%7C~NET~INTERNET~PII%5e%5e%5eM%5e%5e%5e796781315%5e%5e%5e2135-2-SLF~~0189~2135-2~~CDC%5e%5e)""

**RF1 Segment Example:**

RF1^^^SAD^^^509~USVAMC^20180816^^^

rule IsTempAddressUpdateDateMoreRecentAndActive

IsTempAddressUpdateDateMoreRecentAndActive Rule.

How to test: process a Z07 message with a temporary address update and the address update date is more recent.

Figure : IsTempAddressUpdateDateMoreRecentAndActive Rule

How to test: process a Z07 message with a temporary address update; and the address update date is more recent.

ORUZ07 Message - ZTA Segment

Added a ZTA segment with a valid start date and a more recent update date

Set the end date of ZTA segment NULL

**ZTA Segment Example:**

ZTA^1^1^20180523^""^234 TEST TEMP AVE~~LINCOLNIA~VA~22312~USA~C~~059^059^^20180523110027-0500^742

rule IsTempAddressUpdateDateMoreRecentInactiveAndNull

IsTempAddressUpdateDateMoreRecentInactiveAndNull Rule.

How to test: process a Z07 message with a temporary address incoming fields are null and the address update date is more recent.

Figure : IsTempAddressUpdateDateMoreRecentInactiveAndNull Rule

How to test: process a Z07 message with a temporary address incoming fields are NULL; and the address update date is more recent.

ORUZ07 Message - ZTA Segment

Find a Veteran with a temporary address and end date is NULL

Added a ZTA segment with a future start date and a more recent update date

**ZTA Segment Example:**

ZTA^1^1^20180823^""^""^059^^20180525110027-0500^742

rule IsConfAddressUpdateDateMoreRecent

IsConfAddressUpdateDateMoreRecent Rule.

How to test: process a Z07 message with a confidential address update and the address update date is more recent and the start date is not null.

Figure : IsConfAddressUpdateDateMoreRecent Rule

How to test: process a Z07 message with a confidential address update and the address update date is more recent; and the start date is not NULL.

Details:

ORUZ07 Message –

Added a confidential address to PID Segment SEQ=13, DataType=XTA

Added RF1 Segments CAD reference with more recent update date

**Confidential Address Type:**

|  |  |
| --- | --- |
| **Confidential Address Type** | **Description** |
| VACAA | VA - Confidential Address Appointment/Scheduling |
| VACAC | VA - Confidential Address Copayments/Veteran Billing |
| VACAE | VA - Confidential Address Eligibility/Enrollment |
| VACAM | VA - Confidential Address Medical Records |
| VACAO | VA - Confidential Address All Others |

**PID Segment Example:**

PID^1^1008589404V203273^1008589404V203273~~~USVHA&&0363~NI~VA FACILITY ID&200M&L|796781315~~~USSSA&&0363~SS~VA FACILITY ID&608&L|7171927~~~USVHA&&0363~PI~VA FACILITY ID&608&L|~~~USVBA&&0363~PN~VA FACILITY ID&608&L^^BLSTCDCOKRP~BFSTCDCOKRP~BATCH~^^19760118^F^^1002-5-SLF~~005~1002-5~~CDC|2106-3-SLF~~005~2106-3~~CDC^1234 LEGACY DRIVE~#APT 111 ~PLANO~TX~75024~USA~P~~085|222 Confidential Ave.~Apt. 22B~PARNASSUS~PA~15068~USA~VACAE~conf line 3~085~~~20170828&20171231^085^(972)334-1234~ORN~CP|(214)[332-1234~PRN~PH|~NET~INTERNET~PII^^^M^^^796781315^^^2135-2-SLF~~0189~2135-2~~CDC^^](mailto:332-1234~PRN~PH%7C~NET~INTERNET~PII%5e%5e%5eM%5e%5e%5e796781315%5e%5e%5e2135-2-SLF~~0189~2135-2~~CDC%5e%5e)""

**RF1 Segment Example:**

RF1^^^SAD^^^509~USVAMC^20180801^^^

RF1^^^CAD^^^500^20180812^^^

rule isConfAddressExpired2

isConfAddressExpired2 Rule.

How to test: process a Z07 message with a confidential address update and the start and end date of incoming confidential address are null.

Figure : isConfAddressExpired2 Rule

How to test: process a Z07 message with a confidential address update; and the start and end date of incoming confidential address are NULL.

Details:

ORUZ07 Message – PID Segment SEQ=13, DT=XTA

Found a matching confidential address and updated the address start date NULL and end date NULL

added RF1 Segment with reference CAD and more recent update date

**PID Segment Example:**

PID^1^1008589404V203273^1008589404V203273~~~USVHA&&0363~NI~VA FACILITY ID&200M&L|796781315~~~USSSA&&0363~SS~VA FACILITY ID&608&L|7171927~~~USVHA&&0363~PI~VA FACILITY ID&608&L|~~~USVBA&&0363~PN~VA FACILITY ID&608&L^^BLSTCDCOKRP~BFSTCDCOKRP~BATCH~^^19760118^F^^1002-5-SLF~~005~1002-5~~CDC|2106-3-SLF~~005~2106-3~~CDC^1234 LEGACY DRIVE~#APT 111 ~PLANO~TX~75024~USA~P~~085|222 Confidential Ave.~Apt. 22B~PARNASSUS~PA~15068~USA~VACAE~conf line 3~085~~~""^085^(972)334-1234~ORN~CP|(214)[332-1234~PRN~PH|~NET~INTERNET~PII^^M^^^796781315^^^2135-2-SLF~~0189~2135-2~~CDC^^](mailto:332-1234~PRN~PH%7C~NET~INTERNET~PII%5e%5e%5eM%5e%5e%5e796781315%5e%5e%5e2135-2-SLF~~0189~2135-2~~CDC%5e%5e)""

**RF1 Segment Example:**

RF1^^^SAD^^^509~USVAMC^20180801^^^

RF1^^^CAD^^^500^20180812^^^

processAssociation rule flow is invoked when the associate update is made because of the receipt of a Z07 message from a site. Developers then want to modify the rule flow so that it transmits this association update Z05 to all sites including the sending facility that sent the Z07.

A close up of a map

Description generated with high confidence

Figure : ProcessAssociation Rule Flow

rule SendZO5AddressFormat

SendZO5AddressFormat Rule.

How to test: process a Z07 message with a new association and the association type is not VA Guardian, Guardian Civil or POW.

Figure : SendZO5AddressFormat Rule

How to test: process a Z07 message with a new association; and the association type is not VA Guardian, Guardian Civil or POW.

Process a Z07 message with association update and the update date is more recent

Details:

ORUZ07 Message –

add a new ZCT Segment to ORUZ07 Message or

update ZCT segment in ORUZ07 Message with update date more recent than the existing one.

**ZCT Segment Example:**

ZCT^1^1^EMERG~ONE^Friend^304 CASE DR~~DALLAS~TX~75004-1234^9723341234^9723341234^^^20180807203246-0500

ZCT^2^3^EMERG~TWO^Neighbor^304 CASE STDD~~VALARICAO~FL~33594-1234^9723341234^9723341234^^^20180807203246-0500

Contact Type:

1=NOK, 2=2nd NOK, 3=e-contact, 4=2nd e-contact, 5=designee

rule SendZO5GuardianFormat

SendZO5GuardianFormat Rule.

How to test: process a Z07 message with a new association and the association type is VA Guardian, 
OR,
Process a Z07 message with association update and the update date is more recent.

Figure : SendZO5GuardianFormat Rule

How to test: process a Z07 message with a new association; and the association type is VA Guardian.

OR,

Process a Z07 message with association update and the update date is more recent.

Details:

ORUZ07 Message –

add a new ZGD Segment to ORUZ07 Message or

update ZGD segment in ORUZ07 Message with update date more recent than the existing one.

**ZGD Segment Example:**

ZGD^1^1^PVA^""^""^133 HOLLAND~~ALBANY~NY~12208^(518)225-2210^20171010

## Code Changes

Code changes are made in AssociationRuleServiceImpl.java. TriggerZ05 methods are invoked when there is an association update. FilterSitesPersonTriggerEvent filters out the sending facility. By modifying the sending facility to “NULL” in map object, it will not filter out any site. Even though the codes are commented out and the method is never used in ES, developers still want to make the change just in case.

Code changes to be made in AssociationRuleServiceImpl.java.

Code changes need to be made in AssociationRuleServiceImpl.java. The triggerZ05 method is invoked when there is association update. FilterSitesPersonTriggerEvent filters out the sending facility. By modifying the sending facility to null in map object, it will not filter out any site.  Even though the codes are commented out and the method is never used in ES. 

Figure : Code changes to be made in AssociationRuleServiceImpl.java

Code changes also need to be made in ContactInfoRuleServiceImpl.java. The private method notifyVistaForContactInfo is invoked when phone or email contact information is updated. By creating a new HashMap, inheriting the keys from updatedKeysAndSitesMap and setting the value to NULL, the FilterSitesPersonTriggerEvent will filter out nothing and the Z05 message will be triggered to all sites including the sending facility.

Code changes to be made in ContactInfoRuleServiceImpl.java.

Code changes also need to be made in ContactInfoRuleServiceImpl.java. The private method notifyVistaForContactInfo is invoked when phone or email contact information is updated. By creating a new HashMap inheriting the keys from updatedKeysAndSitesMap but setting the value to null, FilterSitesPersonTriggerEvent will filter out nothing and Z05 message will be triggered to all sites not excluding the sending facility.

Figure : Code changes to be made in ContactInfoRuleServiceImpl.java

How to test: Process a Z07 message with phone number update; and the update date is more recent.

Process a Z07 message with Email update and the update date is more recent.

Phone number change test:

Details:

ORUZ07 Message –

updated PID Segment SEQ=13, DT=XTA with a new phone number

added RF1 Segments with most recent update date

**PID Segment Example:**

PID^1^1008589404V203273^1008589404V203273~~~USVHA&&0363~NI~VA FACILITY ID&200M&L|796781315~~~USSSA&&0363~SS~VA FACILITY ID&608&L|7171927~~~USVHA&&0363~PI~VA FACILITY ID&608&L|~~~USVBA&&0363~PN~VA FACILITY ID&608&L^^BLSTCDCOKRP~BFSTCDCOKRP~BATCH~^^19760118^F^^1002-5-SLF~~005~1002-5~~CDC|2106-3-SLF~~005~2106-3~~CDC^5678 LEGACY DRIVE~#APT 222 ~PLANO~TX~75024~USA~P~~085|8000 LEGACY DRIVE~#APT 111 ~PLANO~TX~75024~USA~R~~085^085^(469)334-1234~ORN~CP|()[332-1234~PRN~PH|~NET~INTERNET~PII^^^M^^^796781315^^^2135-2-SLF~~0189~2135-2~~CDC^^](mailto:mailtoPII)""

**RF1 Segment Example:**

RF1^^^SAD^^^509~USVAMC^20180801^^^

RF1^^^PHH^^^608~USVAMC^20180223110027-0600

RF1^^^CPH^^^608~USVAMC^20180223110027-0600

Email Phone change test:

Details:

ORUZ07 Message –

updated PID Segment SEQ=13, DT=XTA with a new email address

added RF1 Segments with most recent update date

**PID Segment Example:**

PID^1^1008589404V203273^1008589404V203273~~~USVHA&&0363~NI~VA FACILITY ID&200M&L|796781315~~~USSSA&&0363~SS~VA FACILITY ID&608&L|7171927~~~USVHA&&0363~PI~VA FACILITY ID&608&L|~~~USVBA&&0363~PN~VA FACILITY ID&608&L^^BLSTCDCOKRP~BFSTCDCOKRP~BATCH~^^19760118^F^^1002-5-SLF~~005~1002-5~~CDC|2106-3-SLF~~005~2106-3~~CDC^5678 LEGACY DRIVE~#APT 222 ~PLANO~TX~75024~USA~P~~085|8000 LEGACY DRIVE~#APT 111 ~PLANO~TX~75024~USA~R~~085^085^(469)334-1234~ORN~CP|()[332-1234~PRN~PH|~NET~INTERNET~PIIv^^^M^^^796781315^^^2135-2-SLF~~0189~2135-2~~CDC^^](mailto:332-1234~PRN~PH%7C~NET~INTERNET~PII%5e%5e%5eM%5e%5e%5e796781315%5e%5e%5e2135-2-SLF~~0189~2135-2~~CDC%5e%5e)""

**RF1 Segment Example:**

RF1^^^SAD^^^509~USVAMC^20180801^^^

RF1^^^EAD^^^608~USVAMC^20180805110027-0600

# Add Early Separation Reason Code to ZMH Message Segment

## Change Request

It has been requested that a 10th field be added to the ZMH Message Segment in the Z11 Messages. This new field will contain the code for the Early Separation Reason, which is already in the ZMH Message Segment as the 9th field.

## Code Changes

Code changes need to be made to ZMH.java to add the Early Separation Reason Code as the 10th field in the ZMH message segment. This involves adding get and set methods which allow for retrieving and storing the code in the message segment.

Code changes in ZMH.java.

Code changes need to be made to ZMH.java to add the Early Separation Reason Code as the 10th field in the ZMH message segment. This involves adding get and set methods which allow for retrieving and storing the code in the message segment.

**Figure 13:** **Code changes in ZMH.java**

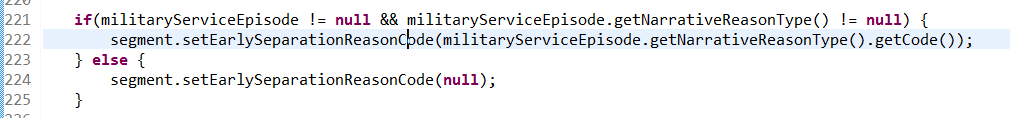


Figure : Code changes in ZMHBuilder.java

In addition, modifications need to be made to ZMHBuilder.java to retrieve the code from the Narrative Reason (if it exists) and store it in the ZMH message segment.

Change to ZMH schema.

Modifications need to be made to ZMHBuilder.java to retrieve the code from the Narrative Reason (if it exists) and store it in the ZMH message segment.

Figure : Change to ZMH schema

The schema for the ZMH segment, ZMH.xsd, must be updated to include the Early Separation Reason Code. In addition, ZMH.xml must also be updated to add the field.

Modifications also need to be made to ensure that the Early Separation Reason Code can be viewed by the user under the Facility Tab in the Enrollment System application. This involves updating esr\_transform.xsl to include the Early Separation Reason Code.

Update to template in order to show Early Separation Reason Code in UI.

viewed by the user under the Facility Tab in the Enrollment System application. This involves updating esr_transform.xsl to include the Early Separation Reason Code.

Figure : Update to template in order to show Early Separation Reason Code in UI

# Add MOH Award Date in ES

## Change Request

It has been requested that developers modify the ES application to support the MOH Award Date field. This change will include the following components:

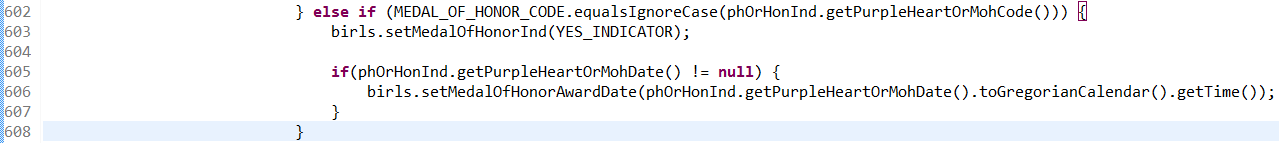
* Updating the MSDS interface to support the MOH Award Date
* Updating the ADR to allow for storage of the MOH Award Date, as well as to add a new capability for modifying MOH information.
* Modifying the user interface for the Military Service and Military Service History screens to display the MOH Award Date
* Updating the Z11 Message to include the MOH Award Date and MOH Status Date

## Database Changes

The database table, which stores information about Medal of Honor records, is called MEDAL\_OF\_HONOR. This table will be modified to add a new DATE column called AWARD\_DATE, which will be used to store the MOH Award Date field. MEDAL\_OF\_HONOR\_H, which stores historical Medal of Honor records, will be modified in the same manner.

The table that stores information pertaining to capabilities is PERMISSION\_TYPE. A new row will be added to this table corresponding to the “Edit MOH” capability.

## Code Changes

The MOH Award Date is already being sent to the eMIS client, but only the MOH indicator is being processed. This occurs in EMISMilitaryInformationServiceClient, where the indicator is stored in a BIRLS object. Therefore, the BIRLS class will be modified to add a field for the MOH Award Date (there are two classes named BIRLS, so both will be updated), and then EMISMilitaryInformationServiceClient will be updated to store the Award Date retrieved from eMIS in this field.

**Figure 17: Code Changes to EMISMilitaryInformationServiceClient.java**

The MEDAL\_OF\_HONOR table in the database is mapped to the MedalOfHonor class, so a new field will be added to this class corresponding to the award date. The hibernate configuration in MedalOfHonor.hbm.xml will be updated to complete the mapping between the Award Date field and the column in the MEDAL\_OF\_HONOR table.

Once the MOH Award Date is retrieved from eMIS, it is stored in a MedalOfHonor object. This happens in MilitaryServiceBuilderForMSDS. The requirements specify that the Award Date should not be stored in ES if it is a future date, or if it is earlier than the Veteran’s 15th birthday; and so both conditions will be checked before storing the date in the MedalOfHonor object. If the MOH Award Date received from eMIS is invalid, the value of the Award Date stored in ES will be set to NULL.

****To add the MOH Award Date and MOH Status Date to the Z11 HL7 Message, the setMedalOfHonor method in ZMHBuilder, which currently sets the MOH Indicator, will be modified to store both dates in the ZMH segment. Both dates will be stored in the serviceEntryDateAndServiceSeparationDate field of the ZMH segment after being properly formatted

**Figure 18: Code Changes to ZMHBuilder.java**

The MOH Status Date will be sent regardless of the value of the MOH Indicator. If the value of the Indicator is set to ‘N’, the MOH Award Date will be sent as a NULL value.

**Example ZMH Message Segment with MOH Indicator set to ‘Y’:**

ZMH^1^SL^AIR FORCE~""~HONORABLE^20000101~20010101

ZMH^2^FDD^AIR FORCE~""~""^20130101~""^^^^20190101

ZMH^3^MH^Y^20180101~20181025

**Example ZMH Message Segment with MOH Indicator set to ‘N’:**

ZMH^1^SL^AIR FORCE~""~HONORABLE^20000101~20010101

ZMH^2^FDD^AIR FORCE~""~""^20130101~""^^^^20190101

ZMH^3^MH^N^""~20181025

To support the new “Edit MOH” capability, the Capability class will be updated to add the Code for Edit MOH, so that it will be accessible from other parts of the application.

## User Interface Changes

Both the **Military Service** and **Military Service History** screens will be updated to add a Medal of Honor Award Date field.

If the MOH Indicator is received from the MSDS Broker with a value of “Yes”, the **MOH Award Date** will be displayed in the **Military Service** screen as a read-only label.

Read-Only MOH Award Date Field in Military Service Screen.

If the MOH Indicator is received from the MSDS Broker with a value of “Yes”, the MOH Award Date is displayed in the Military Service screen as a read-only label.

Figure : MOH Award Information Received from MSDS

If the **MOH Award Date** was not received from MSDS and the user has the “Edit MOH” capability, then they will be able to manually enter a value for the Award Date.

MOH Award Date as Editable Field.

If the MOH Award Date was not received from MSDS, and the user has the “Edit MOH” capability, then the Award Date will be presented as an editable field.

Figure : Manually Entered MOH Award Information

The layout for the **Military Service** screen is specified in militaryServiceOverview.jsp. The JSP will be modified to add the **Medal of Honor** **Award Date** field. The field will be implemented so that it can be rendered as either a label or an editable text field, based on the user roles and the source of the Medal of Honor information. The **Document Receipt Date** field will also be modified to appear as a label if the user does not have the “Edit MOH” capability, or if the date was received from the MSDS Broker.

Similar updates will also be applied to the **Medal of Honor Indicator**, **Document Type**, and **Source of Change** fields. The screen will render these fields as read-only if the user does not have the “Edit MOH” capability, or if the data was received from the MSDS Broker.

A **MOH Award Date** field will be added to MilitaryServiceInfoForm so that the value of the Award Date can be passed between the view and the controller.

When the **Medal of Honor Award Date** is editable, it will be a required field. In addition, the field will be validated to ensure that the date entered is not a future date, and that it is not before the Veteran’s 15th birthday. This will be enabled by modifying the validateMedalOfHonor method in MilitaryServiceValidator to check whether the MOH Award Date entered by the user meets the conditions given above.

In addition, the **No Data** radio button under the **Medal of Honor Indicator** will be disabled.

The **Medal of Honor Award Date** will also be displayed in the **Military Service Change History** screen. This will require modifying the configuration in militaryservice/struts-actions.xml to add the **MOH Award Date** field.

Change to militaryservice/struts-actions.xml.

The Medal of Honor Award Date will be displayed in the Military Service Change History screen. This will require modifying the configuration in militaryservice/struts-actions.xml to add the MOH Award Date field.

Figure : Change to militaryservice/struts-actions.xml

# Preferred Name with MVI as Source

## Change Request

It has been requested that developers modify the ES application to support the **Preferred Name**. This will involve changes to the ES user interface to allow the user to add or modify the Preferred Name in the Demographic **Identity Traits** tab and the **Add a Person Search** screen. It will also involve changes to the ES Banner so that the **Preferred Name** can be displayed along with the legal name. Lastly, the interface between ES and MVI will be updated to support sending the Preferred Name.

## Code Changes

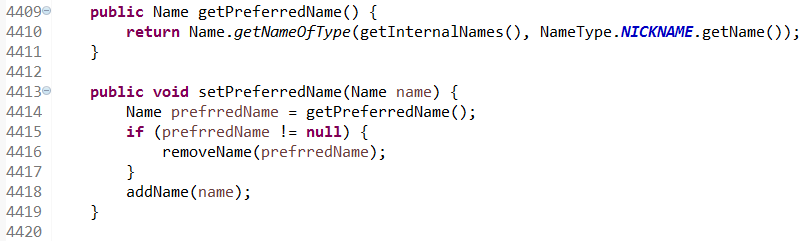
****Each Person record has a set of Name records associated with it. In addition to the parts of the name itself (first name, last name, etc), a Name record also contains a NameType, which specifies what type of name it is. A Preferred Name is identified by a NameType with code “N”, for “Nickname”. To support the Preferred Name functionality, developers will add a getter and setter method to Person which will retrieve and store a Name using this NameType.

Figure : Getter and Setter for Preferred Name

Because developers are simply leveraging the existing name functionality, no additional changes will be needed in either the database or the Person type.

Developers will be updating both the 1301 (Add) and 1302 (Update) requests of the IDM Web Service to include the Preferred Name. To this end, a new method will be added to the IdmServiceVO class which returns the Preferred Name from the list of name records.

A new method, called getPreferredNameFromSet will be added to IdmWebServiceDelegateImpl to allow the Preferred Name to be extracted from a set of names. Then, the populate1301Person and populate1302Person methods will be updated so that the Preferred Name can be populated in the Person portion of both requests.

## User Interface Changes

The **Identity Traits** tab on the **Demographics** screen will be updated to include a Preferred Name field. This will be a free text field which will have a width of 40 characters, and will allow up to 80 characters of input. The layout of the **Identity Traits** tab is specified in demographicIdentityTraitsContent.jsp, so this file will be modified to add tags for the text field. The file DemographicMessages.properties will also be modified to add the label text for the new field.

Although MVI supports multiple components for the Preferred Name (first name, middle name, etc), ES will only have this one field. Therefore, when a user enters a value in the Preferred Name field, it will be stored in the given (or first) name field of the corresponding Name record. Likewise, the given name will be used to populate the field if it already has a value.

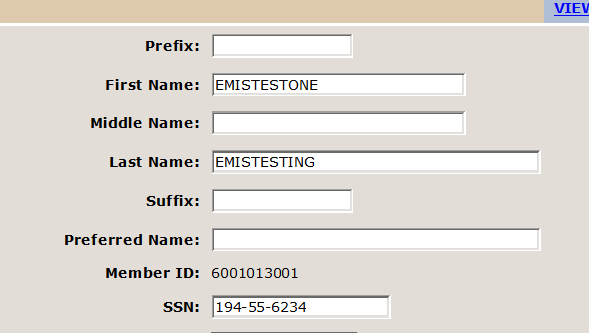


Figure : Preferred Name field on Identity Traits tab

The data on the **Identity Traits** tab is passed to and from the controller in a DemographicIdentityTraitsForm object. The class will be updated to include a field for the Preferred Name, along with a getter and setter method for this field.

DemographicIdentityTraitsConversionService is the type used to transfer data between the Person record and the DemographicIdentityTraitsForm. The doConvertPersontoEditableForm method will be modified to retrieve the Preferred Name from the Person record and store it in the form, so that the value can be populated in the screen. In addition, the convertFormtoPerson method, which is called when the data on the screen is updated will be modified to populate the Person record with the Preferred Name information in the form.

The controller for the **Identity Traits** screen, which is found in DemographicsIdentityTraitsAction.java will be modified to validate the input from this new field. A new method will be added to the controller that will return a boolean value corresponding to whether the Preferred Name is valid. The method will check the following conditions (if a Preferred Name has been provided by the user):

* The Preferred Name is less than or equal to 80 characters in length.
* The Preferred Name only contains alphabetical characters, spaces, hyphens, and apostrophes.

If any of these conditions are not true for the provided name, the method will return false, and an ActionMessage will be added corresponding to the condition that was violated. If all the conditions are met, the method will return true. This method will be invoked from the update method of DemographicsIdentityTraitsAction, so that it will be executed along with the other validations for the form.

In addition, developers will be removing the “View Submitted Identity Traits” and “View Historical Identity Traits” tabs from ES. To do this, developers will remove the links to these tabs which are found in the **Identity Traits** screen. In their place, developers will add a link with the name “View Identity Audit - MVI Toolkit”, which will allow the user to navigate directly to the MVI Toolkit home screen for the current veteran.

The **Preferred Name** field will also be added to the **Add a Person Search** screen. The layout for this screen is specified in searchNewCriteria.jsp. As with the **Identity Traits** screen, a free text field will be added which will allow up to 80 characters of input. However, the field will only be 20 characters wide, in keeping with the rest of the screen’s layout.

SearchActionForm will be modified to add a **Preferred Name** field, so that the information entered on the screen can be sent to the controller. In SearchAction, which serves as the controller for the **Add a Person Search** screen, the createIdmServiceVO and convertIdmServiceVOToPerson methods will be modified to add the mapping of the Preferred Name. This will ensure that the Preferred Name can be sent and received using the IDM Web Service.

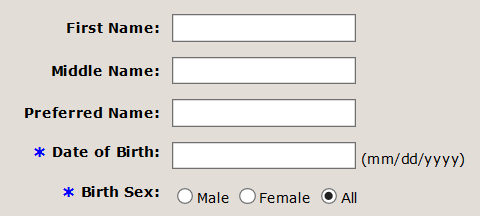
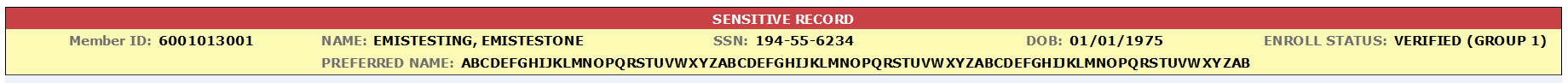


Figure : Preferred Name field in Add a Person Search screen

Lastly, the ES banner will be modified to display the Preferred Name. The layout for the banner is specified in the file veteranBanner.jsp.

The ES banner is represented as an HTML table. The first row in the table is the header, which displays whether the record is a sensitive record, or if it has a Future Discharge Date. The second row is the body of the banner, which shows the Member ID, Name, SNN, DOB, etc.

When a record has a Preferred Name, a new row will be added to the body of the banner. This row will have two cells. The first cell will act as a placeholder, to allow the Preferred Name to appear directly underneath the Name. The second cell will contain the Preferred Name. This cell will be given a colspan of 5, so that if the Preferred Name has a lot of characters, it will render beneath the other data elements, such as the SSN and DOB.

**Figure 25: ES Banner showing Preferred Name of maximum length (80 Characters)**

VeteranHeaderBean is the class which contains the information that is displayed in the banner. A new field will be added to this class so that the Preferred Name can be populated in the banner.

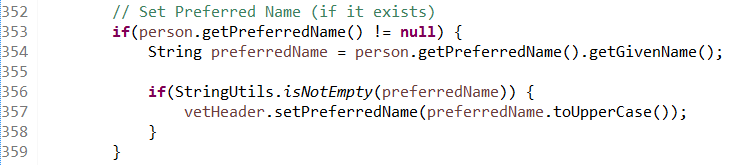
The information shown in the banner is populated in the updateHeader method of PersonAbstractAction. Code will be added to this method which will retrieve the Preferred Name from the Person record, format it, and store it in the VeteranHeaderBean. As previously mentioned, although MVI supports multiple parts for the Preferred Name, ES will only use the given (or first) name to populate the Preferred Name in the application.

Figure : Code changes in PersonAbstractAction to populate Preferred Name in ES Header

# Future Discharge Date from HCA

## Change Request

It has been requested that the developers modify the ES application to support receiving the Future Discharge Date (FDD) from HCA applications. When an application containing an FDD is received, the system will set an enrollment status of “Not Applicable”, rather than “Pending”.

It has also been requested that the developers update ES so that only Future Discharge Dates less than a year in the future will be accepted by the application. This change will also apply to Future Discharge Dates that are received from MSDS.

Lastly, it has been requested that ES be modified to update the Veteran indicator of a record based on the active duty indicator that is received from MSDS.

## Code Changes

When new data is received from HCA, the processVOA method of MessageService, which is implemented in MessageServiceImpl, is called. This method will be modified to check if a Future Discharge Date is present in any of the military service episodes received from HCA. If one is found, the Person record will be updated in the following ways:

* The Veteran Indicator will be set to false
* The Incoming Eligibility Verification Status will be set to “Verified”
* The Incoming Eligibility Verification Source will be set to “VAMC”
* “Sharing Agreement” will be added to the record as a received eligibility on the incoming record
* “Tricare” will be added to the record as a received eligibility on the incoming record

To check whether any of the incoming military service episodes have a future discharge date, a new method will be added to MessageServiceImpl called hasHCAFdd. This method will iterate through every military service episode in every military service site record received from HCA. If an episode is found which does not have a discharge type (meaning that the Veteran has not yet been discharged), the method will return a value of true. Otherwise, it will return false.



Figure : Implementation of hasHCAFdd method

Typically, the Veteran indicator is not changed while processing incoming VOA data. However, because the Veteran indicator will be changed to “false” when an FDD is received, additional changes will be made to processVOA to ensure that the incoming eligibility verification and Veteran indicator are populated correctly.

In addition, the determinePatientType method in PersonServiceImpl will be modified to return a Patient Type of “Non-Veteran” when the person has an eligibility type of “Sharing Agreement” but does not have a current period of service. This will ensure that the patient type for the person is set correctly for the scenario where an FDD is returned from HCA.

To support the changes that they will be making to the business rules (detailed below), developers will modify the MilitaryServiceInputParameter class. A new method will be added to this class called resultHasHCAFdd which will search the military service site records from HCA and return true if one of them has a Future Discharge Date. The checkForExistingHecSiteRecord method will also be modified to create a blank military service record if one doesn’t already exist.

## User Interface Changes

Currently, the Future Discharge Date (FDD) field on the **Military Service** screen is validated to make sure that the date entered is not more than two years in the future. This validation occurs in the isFutureDischargeDateCorrect2 method of the MilitaryServiceInfoForm class (which is invoked from MilitaryServiceValidator).

This method will be modified to check if the provided FDD is more than a year in the future, rather than two years. If it is, a message will be displayed to the user informing them that the date they entered is too far in the future. The text of the error message is stored in MilitaryServiceMessages.properties, so the file will be edited to reflect this change.

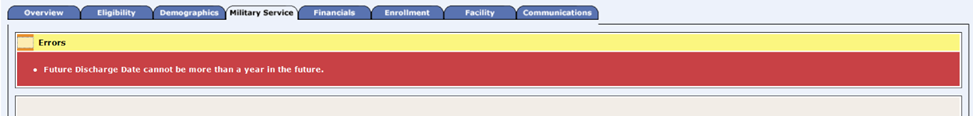


Figure : Updated validation of Future Discharge Date on Military Service screen

## Rule Changes

The incoming eligibility verification data will be processed and accepted as part of the ProcessNewVeteranRuleFlow. A check on the FDD being self-reported will be added to determine whether the eligibility verification data should be accepted.

The DetermineEligibility rule flow will be updated to set the primary and secondary eligibility of the person when an FDD is received from HCA. The PrimaryCodeIsSharingAgreement rule will be modified to check whether the incoming person is a new record with an FDD from VOA. If both conditions are met, the person’s primary eligibility will be set to “Sharing Agreement”. The SecondaryCodeIsTriCare rule will be modified in the same manner, and will set the person’s secondary eligibility to “Tricare/CHAMPUS” if both conditions are met.

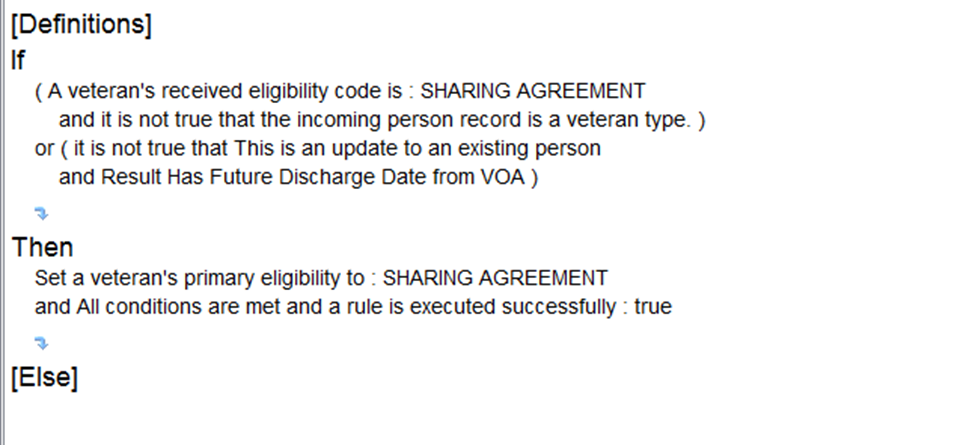


Figure : Modified PrimaryCodeIsSharingAgreement rule

In addition, the NewPersonIsNonVeteran rule in the ProcessNewVeteran rule flow will be updated to account for the FDD. If an FDD has been self-reported, the rule will accept the eligibility verification data from the incoming HL7 message.

The ProcessMilitaryService rule flow is invoked when military service information is updated, either through the UI or a service (such as MSDS). The acceptFutureDischargeFromEMIS rule, which is part of this rule flow, determines whether to accept a Future Discharge Date that has been received from MSDS. As currently implemented, this rule only checks whether the “Accept FDD from MSDS” system parameter is set to “Y”; and accepts the incoming FDD if it is. The rule will therefore be updated to also consider how far in the future the FDD is.

Developers will modify the ProcessMSDS rule flow to set the Veteran indicator based on the incoming active duty indicator. The SUC1013\_6\_11to18\_SetVeteranIndicatorRules sub flow implements the rules used to determine how the Veteran indicator is set, so it will be modified to consider the incoming active duty indicator. If the incoming active duty indicator is “No”, then the Veteran indicator will be set to “Yes”; and if the incoming active duty indicator is “Yes”, then the Veteran indicator will be set to “No”.