Veterans Health Administration (VHA)

Office of Informatics and Analytics (OIA)

Veterans and Consumers Health Informatics Office (V/CHIO)

Business Use Case (BUC)# : 20120315.2 My Health*e*Vet (MHV) Secure Message (SM) Patient Application Program Interface (API)

MHV End User Functionality   
Enhancements FY12 - FY16

New Service Request (NSR) 20120315



**May 2015**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 12/11/2013 |  | Initial draft created | Terry Cooper |
| 12/20/2013 |  | Updated with review comments | Terry Cooper |
| 12/29/2013 |  | Updated with latest comments from Carnetta Scruggs | Terry Cooper |
| 12/30/2013 |  | Updated per Marcia Berg comments | Terry Cooper |
| 12/31/2013 |  | Final version; Data Analytics input to be added when available | Terry Cooper |
| 1/17/2014 |  | Updated per PD input | Terry Cooper |
| 3/26/14 | 0.9 | Editorial review | Marcia Berg |
| 4/22/2014 | 0.11 | Updates from editorial review | Terry Cooper |
| 4/23/2014 | 1.0 | Sign off by MHV Director | Theresa Hancock |
| 4/30/2015 | 1.1 | Revised description of Message Account Log (MAL) now called “API Trace Log”. Incorporated additional technical edits including definitions table; updated AAH screenshot in Appendix; Added My Health***e***Vet Metrics of Use Global Requirements | Andrew Slawter |
| 5/12/2015 | 1.2 | Technical review approved by Carnetta Scruggs | Andrew Slawter |
| 5/26/2015 | 1.3 | Editorial review approved by Marcia Berg | Andrew Slawter |
| 5/26/2015 | 2.0 | Baseline for 12.13 | Andrew Slawter |

Table of Contents

[1. MHV Secure Messaging APIs 1](#_Toc418186556)

[Use Case Description 1](#_Toc418186557)

[Brief Description 1](#_Toc418186558)

[2. Preconditions 3](#_Toc418186559)

[Preconditions 3](#_Toc418186560)

[3. Basic Flow of Events 4](#_Toc418186561)

[3.1. Basic Flow 6](#_Toc418186565)

[4. Alternative Flows 7](#_Toc418186566)

[4.1. API Permission Denied 7](#_Toc418186571)

[4.2. API User Not Found 7](#_Toc418186572)

[4.3. API Empty Secure Message List 7](#_Toc418186573)

[4.4. API Empty Triage List 7](#_Toc418186574)

[4.5. API Empty Subject List 7](#_Toc418186575)

[4.6. API Folder Not Found 7](#_Toc418186576)

[4.7. API Message Move Not Permitted 7](#_Toc418186577)

[4.8. API Folder Already Exists 8](#_Toc418186578)

[4.9. API Folder Not Empty 8](#_Toc418186579)

[4.10. API System Error 8](#_Toc418186580)

[4.11. API Token Invalid 8](#_Toc418186581)

[4.12. Message Not Found 8](#_Toc418186582)

[4.13. API Attachment List Empty 8](#_Toc418186583)

[4.14. API Message Format Error 8](#_Toc418186584)

[4.15. API Unsupported File Format 8](#_Toc418186585)

[4.16. API File Size Exceeded 9](#_Toc418186586)

[4.17. API File Incomplete 9](#_Toc418186587)

[4.18. API To Folder Not Found 9](#_Toc418186588)

[4.19. API From Folder Not Found 9](#_Toc418186589)

[4.20. API Attachment Not Found 9](#_Toc418186590)

[5. Post Conditions 10](#_Toc418186591)

[Post Conditions 10](#_Toc418186592)

[6. Special Requirements 11](#_Toc418186593)

[7. Extension Points 12](#_Toc418186594)

[8. Sample Use Flow 13](#_Toc418186595)

[A. Appendix – Glossary 14](#_Toc418186596)

[B. Appendix – API Requests 17](#_Toc418186597)

[Base API Template 17](#_Toc418186598)

[1. Secure Messaging Patient APIs 18](#_Toc418186599)

[1.1. Initiate Session 18](#_Toc418186600)

[1.2. Retrieve Folder List 19](#_Toc418186601)

[1.3. Retrieve Folder Metadata 20](#_Toc418186602)

[1.4. Retrieve Message List 21](#_Toc418186603)

[1.5. Retrieve Message Content 22](#_Toc418186604)

[1.6. Retrieve Triage List 23](#_Toc418186605)

[1.7. Retrieve Subject List 24](#_Toc418186606)

[1.8. Retrieve Message Attachment List 25](#_Toc418186607)

[1.9. Retrieve Message Attachment 26](#_Toc418186608)

[1.10. Retrieve Message History 27](#_Toc418186609)

[1.11. Execute Send Message 28](#_Toc418186610)

[1.12. Execute Send Message with Attachment 29](#_Toc418186611)

[1.13. Execute Delete Message 30](#_Toc418186612)

[1.14. Execute Move Message 31](#_Toc418186626)

[1.15. Execute Reply Message 32](#_Toc418186627)

[1.16. Execute Save Draft Message 33](#_Toc418186628)

[1.17. Execute Folder Creation 34](#_Toc418186629)

[1.18. Execute Folder Deletion 35](#_Toc418186630)

[C. Appendix – Error Code Mapping 36](#_Toc418186631)

[D. Appendix – Account Activity History (AAH) 37](#_Toc418186632)

[9. Appendix - My Health*e*Vet Metrics of Use Global Requirements 39](#_Toc418186633)

[E. Appendix – Approval Signatures 41](#_Toc418186648)

Table of Figures

[Figure 1 API Primary Flow (includes basic alternative flows as well) 5](#_Toc418185722)

[Figure 2: Sample MHV SM Example of Sent Message 13](#_Toc418185723)

[Figure 3: Account Activity History Sample 37](#_Toc418185724)

# MHV Secure Messaging APIs

This Business Use Case (BUC) document describes the Business Use Case for the My Health***e***Vet (MHV) Application Programming Interface (API) set that supports Secure Messaging (SM). The primary purpose of the MHV API is to permit systems apart from MHV to send requests to MHV to retrieve data, post data, or execute a function of Secure Messaging, with the request results returned to the caller.

This document reflects the Use Case needs as they are understood at this time. The use of the Agile Methodology in MHV requires that this document, and all BUCs, is subject to revision as additional relevant information surfaces. Additional constraints and limitations from Product Development or others may also require revision to this BUC. As a result, this document includes notes where there are known topics that may require further elaboration or clarification as the development of the features proceeds.

## Use Case Description

The MHV SM APIs provide the ability for external systems to send requests to MHV for execution and receive a response from MHV. This Use Case outlines the primary flow for the MHV SM APIs, and the External API Reference Appendix contains references to API’s that are available for interfacing with the MHV API.

## Brief Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | **Role** | **Frequency of Use** | |
| External System | This is the primary actor of the Use Case process and makes calls to MHV for execution of the API. | 24/7 | |
| MHV | The MHV system receives the API requests, executes the requests, and returns the results of the request to the caller. | 24/7 |

External System

Execute API

**Business Use Case Diagram**

MHV System

Figure 1: Business Use Case Diagram with Actors

# Preconditions

## Preconditions

1. MHV and its component subsystems are functional and operational, including the API interface server.
2. Each request is properly validated per the requirements of the Overarching MHV API BUC# 192
3. The external application has authenticated the user and provides the necessary information in the initial API call to allow MHV to identify the user.
4. The API data structure for every API Request Type contains the following elements:

|  |  |
| --- | --- |
| **API Component** | **Component Description** |
| API Request Type | Designated name or number for that API call |
| User Context | Exact content TBD pending resolution of security mechanism or mechanisms that are supported by Product Development. In all cases, this property contains sufficient information that allows MHV to identify the MHV user and the name of the calling system/application. |
| Sent Data Component(s) | Data component(s) sent via the API call; varies by specific Request Type and is detailed in each Request Type description in [Appendix\_B\_API\_Requests](#Appendix_B_API_Requests) |

1. Each API response contains the following elements:

|  |  |
| --- | --- |
| Response Value(s) | Varies by specific Request Type; Typically “Success” or “Error;” If Error, then Error Condition field is populated. |
| Error Condition(s) | Error condition tuple comprised of Error Code; Error Messages (ex: EA0022;The designated folder cannot be found) |
| Response Data Component(s) | Data component(s) that are returned to the caller for each response (optional); further defined in by each Request Type |

# Basic Flow of Events

The basic flow for the Secure Message APIs is provided below. Note that this flow is generic and represents the flow for each API in the MHV Secure Messaging set as defined in [Appendix\_B\_API\_Requests](#Appendix_B_API_Requests).

Figure 2: API Primary Flow (includes basic alternative flows)





## Basic Flow

|  |  |  |  |
| --- | --- | --- | --- |
| **Process Step** | **Business Rules** | | **Business Data Reference** |
| Step 1: MHV evaluates the API call against the MHV user permissions.  Note: Appendix B API Details lists each of the advertised APIs | 1. If the user is authorized to perform the requested API function, then the next step is performed. Each API call contains a negotiated “token” that is used to determine the user for which the API call is to be executed. Each API call must be validated to ensure that the API call is permitted (permissions error) for this user. The scope of this condition ranges from user level (Basic, Premium, etc.) to programming errors for conditions such as calling a Provider Secure Message API when the user is a Patient user. All of these conditions will be grouped as “permission errors,” although more specific error codes may be returned when available. 2. If the user is not authorized for the API function, then an Error response is returned to the user and the processing is terminated. See [Appendix\_E\_Error\_Code\_Mapping](#Appendix_E_Error_Code_Mapping) | * Error Code | |
| Step 2: MHV executes the API call for the MHV user and adds any necessary entries to the MHV AAH. | 1. The requested API call is executed and requested data is returned to the user. 2. Each API may have multiple error conditions which generate corresponding error codes. *Each of these error codes is defined in Appendix E Error Code Mapping* | * API Response Code * Error Code * API Response Data | |
| Step 3: The MHV system records the transaction in the API Trace Log (ATL) | 1. MHV records the API transaction in the API Trace Log (ATL).   Note: the API Trace Log is intended for diagnostic and traceability purposes. It is not accessible by the user. | * ATL | |

# Alternative Flows

The Alternative Flows delineate the various alternative flows that can occur when MHV executes a particular API. Note that not every API has all Alternative Flows. Refer to [Appendix E Error Code Mapping](#Appendix_E_Error_Code_Mapping) to determine which API has which Alternative Flows



## API Permission Denied

1. This Alternative Flow continues from Step 1.
2. If the requested API is not permitted by the user permissions, then a generic “Permission Denied” error code is returned to the user. No additional information is provided in order to ensure that unauthorized users do not gain additional information or knowledge about the unauthorized data.
3. The flow continues with Step 4.

## API User Not Found

1. This Alternative Flow continues from Step 2.
2. If the user (Veteran, Provider, etc.) is not found, then an error condition is set in the response field and an error tuple is provided in the response Error Condition field.
3. The flow continues with Step 3.

## API Empty Secure Message List

1. This Alternative Flow continues from Step 2.
2. If the a requested Secure Message List is empty, then an error condition is set in the response field and an error tuple is provided in the response Error Condition field.
3. The flow continues with Step 3.

## API Empty Triage List

1. This Alternative Flow continues from Step 2.
2. If the Secure Messaging Triage List is empty, then an error condition is set in the response field and an error tuple is provided in the response Error Condition field.
3. The flow continues with Step 3.

## API Empty Subject List

1. This Alternative Flow continues from Step 2.
2. If the Secure Messaging Subject List is empty, then an error condition is set in the response field and an error tuple is provided in the response Error Condition field.
3. The flow continues with Step 3.

## API Folder Not Found

1. This Alternative Flow continues from Step 2.
2. If the specified Folder is not found, then an error condition is set in the response field and an error tuple is provided in the response Error Condition field.
3. The flow continues with Step 3.

## API Message Move Not Permitted

1. This Alternative Flow continues from Step 2.
2. If the Secure Messaging Move request is not permitted, then an error condition is set in the response field and an error tuple is provided in the response Error Condition field.
3. The flow continues with Step 3.

## API Folder Already Exists

1. This Alternative Flow continues from Step 2.
2. If the folder already exists for a request to create that same folder, then an error condition is set in the response field and an error tuple is provided in the response Error Condition field.
3. The flow continues with Step 3.

## API Folder Not Empty

1. This Alternative Flow continues from Step 2.
2. If the Delete Folder requests deletion of a folder that still contains messages, then an error condition is set in the response field and an error tuple is provided in the response Error Condition field.
3. The flow continues with Step 3.

## API System Error

1. This Alternative Flow continues from Step 2.
2. If MHV encounters any system error (database not available, out of memory, etc.) then an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The flow continues with Step 3.

## API Token Invalid

1. This Alternative Flow continues from Step 2.
2. If the calling system provides an invalid access token, then an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The flow continues with Step 3.

## Message Not Found

1. This Alternative Flow continues from Step 2.
2. If the message identified in the API call is not found, an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The flow continues with Step 3.

## API Attachment List Empty

1. This Alternative Flow continues from Step 2.
2. If the list of Attachments is empty, then an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The flow continues with Step 3.

## API Message Format Error

1. This Alternative Flow continues from Step 2.
2. If the message format and content provided via the API does not meet validity checks, then an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The flow continues with Step 3.

## API Unsupported File Format

1. This Alternative Flow continues from Step 2.
2. If the supplied attachment is not of the acceptable formats, then an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The supported formats are:
   1. Text file (txt): a filename extension for text files.
   2. Portable Document Format (pdf): a file format used to represent documents in a manner independent of application software, hardware, and operating system.
   3. DOC file (doc): a word processing file format, typically used by Microsoft Word.
   4. Microsoft Excel file format (xls): a spreadsheet file format.
   5. Graphics Interchange Format (gif): a bit map image format.
   6. Joint Photographic Group (jpg): bitmap compression format for picture and image files.
   7. Rich Text Format (rtf): a document file format developed by Microsoft for cross-platform document interchange.
   8. Portable Network Graphics (png): a raster graphics file format that supports lossless data compression.
4. The flow continues with Step 3.

## API File Size Exceeded

1. This Alternative Flow continues from Step 2.
2. If the size of the attachment file exceeds 3 MB, then an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The flow continues with Step 3.

## API File Incomplete

1. This Alternative Flow continues from Step 2.
2. If the attachment file is not complete (only part of the file has been transmitted from the external system), then an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The flow continues with Step 3.

## API To Folder Not Found

1. This Alternative Flow continues from Step 2.
2. If the move message API indicates a “To” folder that is not found, then an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The flow continues with Step 3.

## API From Folder Not Found

1. This Alternative Flow continues from Step 2.
2. If the move message API indicates a “From” folder that is not found, then an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The flow continues with Step 3.

## API Attachment Not Found

1. This Alternative Flow continues from Step 2.
2. If the attachment indicated by the API request is not found, then an error condition is set in the response field and an error tuple is provided in the response Error Condition Field.
3. The flow continues with Step 3.

# Post Conditions

## Post Conditions

1. API request completed and a response was returned to external system

# Special Requirements

1. Each API transaction (API call) is recorded in the API Trace Log (ATL). The ATL will record information in a human readable format to answer questions designed to assist Information Technology Security personnel who may be called on to respond if there is security incident. The ATL will adhere to VA and Federal Government records retention policies including the length of time API transactions are required to be accessible. Reports and extracts from the ATL will be determined once reporting tools are determined and in place. The ATL is intended for diagnostic, audit and traceability purposes and is not visible to the MHV user.
2. The ATL will answer the following questions:

* What was the point of origin of the request?
  + What external application made the API request?
  + What external user made the API request (this should include IP address)?
* What MHV user account was accessed?
* What API was called (Request Type)?
* What information was requested?
* Was the transaction successfully completed?
  + If not, what error codes were returned?
* What was the date and time of the API call?

1. My Health***e***Vet Metrics of Use Global Requirements

* Refer to **Appendix E** My Health***e***Vet Metrics of Use Global Requirements

1. Message lists in Secure Messaging are limited to a date range of today through to today minus 365 days. Messages older than one year are not available via the API or in the MHV Secure Messaging application.

# Extension Points

There are no Extension Points in this BUC.

# Sample Use Flow

For this example, the external system wishes to retrieve and display the user’s Secure Messaging folders. The SM APIs they would call in this order are:

1. Initiate Session (User Information)
2. Retrieve Folder List
3. Retrieve Folder Metadata (for each folder in folder list)
4. Retrieve Message List (for each folder in folder list)

They now have sufficient information to populate information that would look like this in the MHV environment:

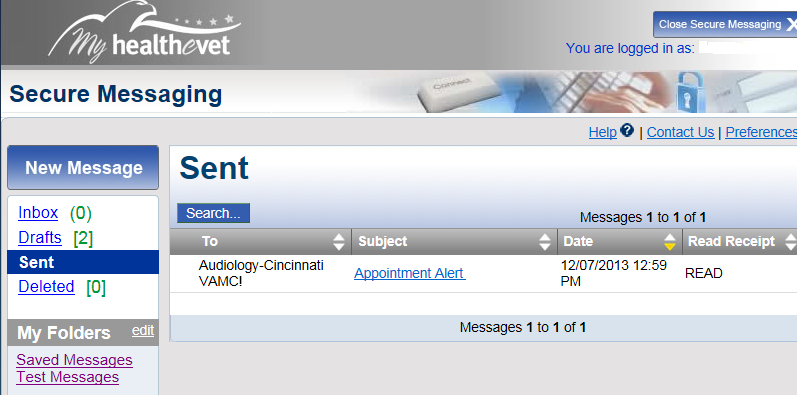


Figure 1: Sample MHV SM Example of Sent Message

1. Appendix – Glossary

| **Term** | **Definition** |
| --- | --- |
| AAH | Account Activity History. Also referred to as the Account Activity Log (AAL) |
| AAL | Account Activity Log. Also referred to as the Account Activity History (AAH). |
| Actor | A role played by a user when using the system. An actor may be a person or another system. An actor may input information or receive information. |
| Alternative Flow | Secondary paths which are variations of the Basic Flow of Events. |
| API | Application Programming Interface |
| API Request Type | Designated name or number for that API call |
| ATL | API Trace Log. Formerly referred to as Message Account Log (MAL), and fully described in Section 6 Special Requirements. |
| Basic Flow | The normal or usual course of events within a Business Use Case. |
| BUC | Business Use Case. A sequence of transactions performed by a user to produce a measurable result of value; a dialog between an actor and the system; a specific way of interacting with the system to accomplish a business-related goal. |
| CDW | Corporate Data Warehouse |
| DOC | DOC file (doc): a word processing file format, typically used by Microsoft Word. |
| ERD | Entity Relationship Diagram |
| Error Condition | Error condition tuple comprised of Error Code; Error Messages (ex: EA0022;The designated folder cannot be found) |
| GIF | Graphics Interchange Format (gif): a bit map image format. |
| ID | Identifier |
| Inbound Request | Inbound requests are viewed from the perspective of the Over-Arching MHV API and seen as any electronic request coming from an external source to the Over-Arching MHV API for processing. |
| Interface | A point of communication between two systems or between an actor and a system (often a GUI today.) |
| IP | Internet Protocol address. A unique ID which can be used to identify an Actor or a website |
| JPG | Joint Photographic Group (jpg): bitmap compression format for picture and image files. |
| ATL | API Message History Log is the record of each API execution and records the Date/Time, Request Type, User Identification, Calling System, and Response (with Error, if any) |
| MHV | My Health***e***Vet |
| NSR | New Service Request |
| Outbound Request | Outbound responses are viewed from the perspective of the Over-Arching MHV API and seen as any response created by the Over-Arching MHV API to answer a request. Outbound responses are sent back to the request source. |
| OIA | Office of Informatics and Analytics |
| PD | The Product Development organization under the MHV program |
| PDF | Portable Document Format (pdf): a file format used to represent documents in a manner independent of application software, hardware, and operating system. |
| PNG | Portable Network Graphics (png): a raster graphics file format that supports lossless data compression. |
| Precondition | An event which must logically occur before the Basic Flow of Events begins. |
| Primary Actor | A role one carries out when using the system in order to implement a defined business process. |
| Request Type | A request type is defined in the definition document for external APIs. Request types inform the Over-Arching MHV API as to what application to extract data for (Secure Messaging, etc.…). |
| Response Data Component | Data component(s) that are returned to the caller for each response (optional); further defined in by each Request Type |
| Response Value | Varies by specific Request Type; Typically “Success” or “Error;” If Error, then Error Condition field is populated. |
| RTF | Rich Text Format (rtf): a document file format developed by Microsoft for cross-platform document interchange. |
| Send Data Component | Data component(s) sent via the API call; varies by specific Request Type and is detailed in each Request Type description in [Appendix\_B\_API\_Requests](#Appendix_B_API_Requests) |
| SM | Secure Messaging |
| SSL | Secure Socket Layer: a layer in the definition of a network protocol |
| SSN | Social Security Number |
| TBD | To Be Determined |
| TXT | Text file (txt): A filename extension for text files. |
| User Context | Exact content TBD pending resolution of security mechanism or mechanisms that are supported by Product Development. In all cases, this property contains sufficient information that allows MHV to identify the MHV user and the name of the calling system/application. |
| V/CHIO | Veterans and Consumers Health Informatics Office |
| VSSC | VHA Support Services Center |
| VHA | Veterans Health Administration |
| XLS | Microsoft Excel file format (xls): a spreadsheet file format. |

1. Appendix – API Requests

This appendix provides the details for each MHV Secure Message API request type that is callable by an external system.

The following base template is used for the definition of each API and its structure/content.

## Base API Template

|  |  |
| --- | --- |
| **API Component** | **Component Description** |
| API Name | Unique name or designation for this API call (*i.e.* Request Type) |
| Purpose | What does this API call do? An API can do one of three things: 1) Retrieve Data; 2) Send Data; 3) Request execution of some function |
| User Context | Sent via the API to communicate who is making this API call. MHV must validate that this user context exists in MHV. Note that external systems may not necessarily have the user credentials aligned with MHV user credentials. The context should also contain a data element that identifies the calling system identification (name). |
| Sent Data Component(s) | Data component(s) sent via the API call; defined in each specific request type. |
| Normal Response(s) | Varies by specific Request Type; Typically “Success” or “Error;” If Error, then Error Condition field is populated. |
| Error Condition(s) | Error Condition: Describes the required error conditions from a business perspective, and those most relevant to the business. For specific list of error conditions, refer to each API’s Interface Control Document. Error condition tuple comprised of Error Code; Error Message (i.e. XXXXXX;Message). |
| Response Data Component(s) | Data component(s) that are returned to the caller for each response (optional); further defined in by each Request Type |

# Secure Messaging Patient APIs

## Initiate Session

See the Overarching API BUC for details on the requirements for initiating a session. The Initiate Session API call is the first step to using any MHV or SM APIs, and is commonly described for use across all MHV and SM APIs.

## Retrieve Folder List

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Retrieve\_Folder\_List |
| Purpose | This API receives a User Context from the calling service and returns a list of folders for the user. There is no differentiation between custom and system folders. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | None |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | Folder List |

## Retrieve Folder Metadata

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Retrieve\_Folder\_Metadata |
| Purpose | This API receives a User Context from the calling service and returns the metadata about the folder. Metadata includes number of messages in the folder, size of folder, etc. There is no differentiation between custom and system folders. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | None |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User Not Found; * Folder Not Found * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | Folder List |

Usage Note: Folder List Metadata contains

* Number of Messages
* Size of folder
* Permission to move messages to this folder
* Permission to move messages from this folder

## Retrieve Message List

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Retrieve\_Message\_List |
| Purpose | This API receives a User Context and Folder Identifier from the calling service and returns the message list in that folder. The message body is not returned, but the following fields are returned:   * Date/Time of message receipt * From addressee * To addressee(s) * CC addressee(s) * Subject |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Folder Identifier |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Empty Message List * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | Folder Metadata |

Note: Paging of the message list is not supported. The need for this capability will be assessed based on feedback from the consumers of the first phase of the APIs.

Note: Server-side sorting of messages is not supported. The need for this capability will be assessed based on feedback from the consumers of the first phase of the APIs.

## Retrieve Message Content

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Retrieve\_Message\_Content |
| Purpose | This API receives a User Context and Message Identifier from the calling service and returns the message content (message body) for that Message. The Message is automatically marked as Read. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Message Identifier (from message list) |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Message Not Found * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | Message Content |

## Retrieve Triage List

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Retrieve Triage List |
| Purpose | This API receives a list of current acceptable message Triage destinations. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | None |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Triage List Empty * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | None |

## Retrieve Subject List

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Retrieve Subject List |
| Purpose | This API receives a list of acceptable message Subject strings (also known as Category). |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | None |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Subject List Empty * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | Subject List data |

**=Note: Additional List items cannot be added via API calls.**

## Retrieve Message Attachment List

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Retrieve\_Message\_Attachment\_List |
| Purpose | This API receives a User Context from the calling service and retrieves a list of attachments for a supplied Message identifier. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Message Identifier; |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Message not found; * Attachment List Empty * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | Attachment List |

## Retrieve Message Attachment

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Retrieve\_Message\_Attachment |
| Purpose | This API receives a User Context from the calling service and retrieves a specified attachment from a supplied Message identifier. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Message Identifier; Attachment Identifier |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Message not found; * Attachment Not Found * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | Attachment File |

## Retrieve Message History

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Retrieve\_Message\_History |
| Purpose | This API receives a User Context from the calling service and retrieves the history of the message indicated.. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Message Identifier |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Message not found; * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | Message History |

## Execute Send Message

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Execute\_Send\_Message |
| Purpose | This API receives a User Context from the calling service and sends a message on behalf of the user. A copy of the messages is placed in the user’s Sent folder. |
| inUser Context | User Context of the user of this API |
| Sent Data Component(s) | Content of new message |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Permission Denied * Message Format Error * System Error * Token Invalid |
| Response Data Component(s) | None |

Notes on Use:

To send a message that has been saved in the Draft folder, the external system must first retrieve the full Draft message content (Retrieve Message Content) and then populate a new message with that content before calling this Execute Send Message API. If the message in the Draft folder is to be removed, then it must also have a specific Execute Move Message to move the message to the Deleted folder.

## Execute Send Message with Attachment

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Execute\_Send\_Message\_With\_Attachment |
| Purpose | This API receives a User Context from the calling service and sends a message with attachment on behalf of the user. A copy of the messages is placed in the user’s Sent folder. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Content of new message + Attachment File |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Permission Denied * Message Format Error * Unsupported File Format * File Size Exceeded * File Incomplete * System Error * Token Invalid |
| Response Data Component(s) | None |

## Execute Delete Message

(This API is listed only to ensure reviewers understand that this capability is provided by Execute Move Message where the destination (To) folder is the Deleted system folder.)



## Execute Move Message

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Execute\_Move\_Message |
| Purpose | This API receives a User Context from the calling service and Moves a message from one folder to another on behalf of the user. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Message Identifier, From Folder Identifier; To Folder Identifier |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found * Message not found * To Folder not found * From Folder not found * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | None |

**If this API is to be used as a Delete Message feature, that intention should be spelled out here.**

## Execute Reply Message

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Execute\_Reply\_Message |
| Purpose | This API receives a User Context from the calling service and Replies to an identified message in a particular folder on behalf of the user. A copy of the message is placed in the Sent System folder. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Message Identifier, Folder Identifier; Reply message content |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Message not found; * Folder not found; * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | None |

Note on Use:

A user who changes Triage Groups may have messages from a Triage Group that is not in the list of Triage Groups after the change. As a result, the user cannot send messages to the old Triage Group. If the user attempts to reply to a message from an old Triage group, that attempt will fail.

## Execute Save Draft Message

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Execute\_Save\_Draft\_Message |
| Purpose | This API receives a User Context from the calling service and Saves an identified message in the System Draft folder on behalf of the user. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Folder Identifier; Saved message content |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Folder not found; * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | None |

## Execute Folder Creation

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Execute\_Folder\_Creation |
| Purpose | This API receives a User Context from the calling service and Creates a new Custom folder for the user |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Folder Identifier |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Folder already exists; * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | None |

## Execute Folder Deletion

|  |  |
| --- | --- |
| **API Component** | **Requirement** |
| API Name | Execute\_Folder\_Deletion |
| Purpose | This API receives a User Context from the calling service and Deletes the identified Custom folder for the user. |
| User Context | User Context of the user of this API |
| Sent Data Component(s) | Folder Identifier |
| Normal Response(s) | * Success; * Error |
| Error Condition(s) | * User not found; * Folder not found; * Folder Not empty * Permission Denied * System Error * Token Invalid |
| Response Data Component(s) | None |

1. Appendix – Error Code Mapping

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Permission Denied | System Error | Token Invalid | User Not Found | Empty Message List | Folder Not Found | Move Not Permitted | Folder Exists | Folder Not Empty | Message Not Found | Triage List Empty | Subject List Empty | Attachment List Empty | Attachment Not Found | Unsupported File Format | File Size Exceeds Max | File Incomplete | To Folder Not Found | From Folder Not Found | Message Format Error |
| **API Request Type** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Retrieve Session | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Retrieve Folder List | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Retrieve Folder Metadata | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Retrieve Message List | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Retrieve Message Content | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Retrieve Triage List | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Retrieve Subject List | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Retrieve Message Attachment List | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Retrieve Message Attachment | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Execute Send Message | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Execute Send Message with Attachment | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Execute Move Message | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Execute Reply Message | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Execute Save Draft Message | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Execute Folder Creation | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Execute Folder Deletion | ? |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

2. Appendix – Account Activity History (AAH)

The Account Activity History is viewable by the user (Patient/Veteran). The concern about the new entries from the API is that these new entries are consistent and meaningful to the user. As a result, all APIs would make entries according to the following two rules:

The AAH “Performed By” field will be populated with “System” when the entry is added as the result of an API call; Sample screen shot is provided in Figure 3.

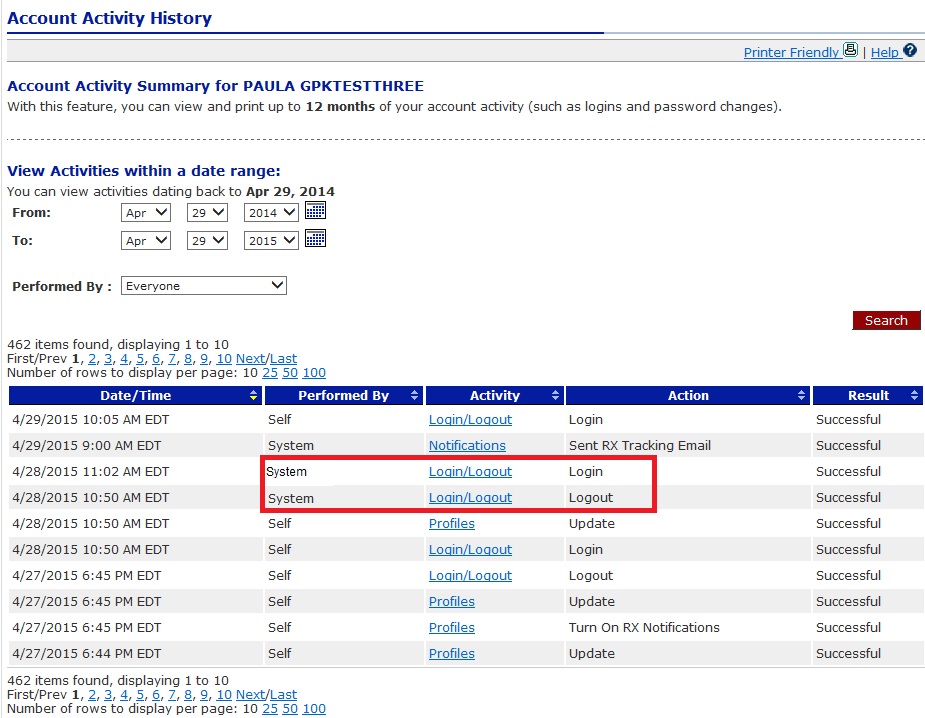


Figure 2: Account Activity History Sample

The AAH “Activity Detail” field will be populated with “**Remote System**: <Application Name>” where <Application Name> is the name of the remote application that is making the API call. An example with “**health*e*living assessment**;” is shown.

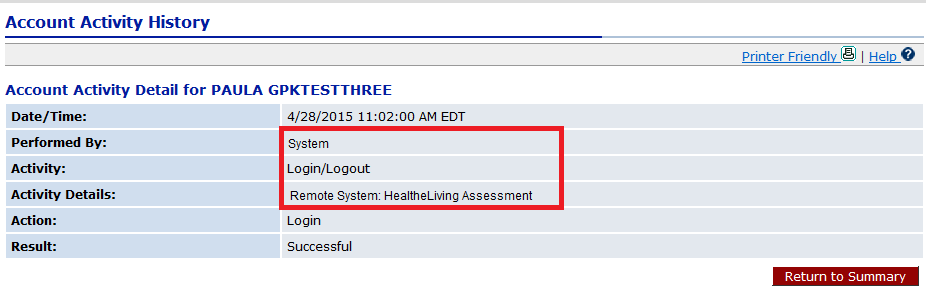


Figure 4: Account Activity History Detail

# Appendix - My Health*e*Vet Metrics of Use Global Requirements

**My Health*e*Vet Metrics of Use Global Requirements**

**Updated 3/13/2015**

===========================================================================================

For EVERY deliverable, My Health***e***Vet has fourteen global requirements to address for Metrics of Use.

Each should be evaluated at the beginning of Requirements Gathering as MUST DO or NOT APPLICABLE.

The MUST DO’s will be enumerated through-out the entire requirements development process.

|  |
| --- |
| MHV VCHIO Data Analytics Global Requirements |
| 1. Deliverable Production Installation Requirement: Any new table in the My Health***e***Vet Data Storage Schema (Oracle) must have Read permission to support data flows to V/CHIO and others (CDW, VSSC, etc…). Data flows delivered by email to the DA Analytics team require developers to grant READ access to the MHV System Pseudo User **MHVPREPORT.** (Other MHV data extracts may run through different MHV System Pseudo Users – appropriate access should be validated in advance of every install.)   **MHVPREPORT** has the following roles:  PORTALMAIN\_RO  METRICS\_RO  SDSADM\_RO  EVAULT\_RO  WEBLOGIC\_RO  SMS\_RO  MHVPORTAL\_RO  PHRMGR\_RO  PD should deliver a script with “grant” statements, for every new object they create.  grant select on <owner>.<object name> to <role\_name>;  The owner of the object will determine the role name to use. |
| 1. Prior to Go/No-go decision for any MHV release – materials for VCHIO will include complete data dictionary documentation for any MHV Tables affected by the changes. Summary of changes should be addressed in the Go/No-go. |
| 1. Data Documentation – should include written Table description that addresses Table purpose and fit in the full MHV Schema. This will support White House mandates for Open Data and compliment current requirements for CDW Metadata. |
| 1. Prior to Go/No-go decision for any MHV release - materials for VCHIO will include complete Entity Relationship Diagrams (ERD) for any MHV Tables affected by the changes. Summary of changes should be addressed in the Go/No-go. |
| 1. My Health***e***Vet User Account Activity Logging |
| 1. V/CHIO Online Metrics - Administrative Portal Modification |
| 1. V/CHIO Online Metrics - Corporate Data Warehouse (CDW) Data Mart Extract |
| 1. V/CHIO Online Metrics - VHA Support Services Center (VSSC) Reporting |
| 1. V/CHIO Online Metrics - Information Technology Performance Dashboard Modifications |
| 1. Google Analytics – Basic Page Coding on every page |
| 1. Google Analytics – Special Item Coding On Demand |
| 1. My Health***e***Vet Web Trends on Demand – Basic Page Coding |
| 1. My Health***e***Vet Web Trends on Demand – Special Item Coding |
| 1. My Health***e***Vet Metrics of Use data captured/displayed/reported by non-V/CHIO sources |

1. Appendix – Approval Signatures

The requirements defined in this document are the high level business requirements necessary to meet the strategic goals and operational plans of the Veterans/Consumers Health Informatics Office.

**My Health*e*Vet Program Management Office**

Signifies that the customer approves the documented requirements, that they adequately represent the customers desired needs, and that the customer agrees with the defined scope for the Business Use Case.

Signed:  Date: June 10, 2015

Theresa Hancock, Director, Veterans/Consumers Health Informatics Office