Enterprise Messaging Infrastructure

Vitria Interface Engine

Standards and Terminology Service

**Service Integration Design Document**



**Department of Veterans Affairs**

**April 2016**

**Version 0.08**

Revision History

| Version No. | Description/Comments | Author | Delivery Date |
| --- | --- | --- | --- |
| 0.01 | Initial document creation and review |  | 06/24/2015 |
| 0.02 | Removed Harris logo | . | 07/02/2015 |
| 0.03 | Updated to incorporate eMI IT review comments from | .  . | 07/30/2015 |
| 0.04 | Updated with implementation details |  | 09/14/2015 |
| 0.05 | Reviewed by SRA prior to being delivered to VA |  |  |
| 0.06 | Updated Contents and diagram |  | 03/01/2016 |
| 0.07 | Added hyperlink to SDD doc and updated queue names. |  | 03/29/2016 |
| 0.08 | Updated HTTPReceiver message flow diagram |  | 04/04/2016 |

Table of Contents

[1.0 Introduction 1](#_Toc444608377)

[1.1 Purpose 1](#_Toc444608378)

[1.2 Scope 1](#_Toc444608379)

[1.3 Audience 2](#_Toc444608380)

[1.4 References 2](#_Toc444608381)

[2.0 Interface Requirement 2](#_Toc444608382)

[2.1 Business Unit 2](#_Toc444608383)

[2.2 Service Level Agreement Metrics 3](#_Toc444608384)

[2.3 Message Type Metrics 3](#_Toc444608385)

[2.4 Logical System Overview 3](#_Toc444608386)

[2.5 Logical Deployment Overview 5](#_Toc444608387)

[2.6 STS – eMI Interface Requirements 5](#_Toc444608389)

[3.0 Nominal VETS Message Flow 8](#_Toc444608390)

[4.0 STS Design 9](#_Toc444608391)

[4.1 Architecture Deviations 10](#_Toc444608392)

[4.2 Pattern 10](#_Toc444608393)

[4.3 Protocol 10](#_Toc444608394)

[4.4 Message Routing 11](#_Toc444608395)

[4.5 Transformation 11](#_Toc444608396)

[4.5.1 Protocol Transformation 12](#_Toc444608397)

[4.5.2 Data Transformation 12](#_Toc444608398)

[5.0 Implementation Details 12](#_Toc444608399)

[5.1 STS 12](#_Toc444608400)

[5.1.1 STS\_HTTPReceiver 13](#_Toc444608401)

[5.1.2 STS\_STS\_MLLPSender 13](#_Toc444608402)

[5.1.3 STS\_MLLPRouter 14](#_Toc444608403)

[5.2 STS VistA Interface 15](#_Toc444608405)

[5.2.1 STS\_MLLPReceiver 15](#_Toc444608406)

[5.2.2 STS\_VISTA\_MLLPSender 16](#_Toc444608407)

[5.3 Error Handling For STS/VistA Message Delivery 17](#_Toc444608408)

[5.4 Project Configuration File 17](#_Toc444608409)

[5.5 Queue Details 18](#_Toc444608410)

[6.0 Timeline 20](#_Toc444608411)

[7.0 Acronyms 21](#_Toc444608412)

[Appendix A. Architecture Design Decisions 23](#_Toc444608413)

[Appendix B. Messages 24](#_Toc444608414)

[Appendix C. Message Mapping 26](#_Toc444608415)

[Appendix D. Outstanding Issues 27](#_Toc444608416)

[Appendix E. Approval Signature 28](#_Toc444608417)

List of Tables

[Table 1 – Software Interfaces 2](#_Toc444082490)

[Table 2 – VistA Business Unit 2](#_Toc444082491)

[Table 3 – VETS Business Unit 3](#_Toc444082492)

[Table 4 – SLA Metrics 3](#_Toc444082493)

[Table 5 – Message Type Metrics 3](#_Toc444082494)

[Table 6 – Validation rule for MSH Segment 6](#_Toc444082495)

[Table 7 – STS/VistA Routing Table (Sample) 7](#_Toc444082496)

[Table 8 – VistA to eMI Interface 10](#_Toc444082497)

[Table 9 – eMI Interface to VETS 10](#_Toc444082498)

[Table 10 – VETS to eMI Interface 11](#_Toc444082499)

[Table 11 – eMI Interface to VistA 11](#_Toc444082500)

[Table 12 – STS Queue Routing Table (Sample) 11](#_Toc444082501)

[Table 13 – Protocol Transformation in STS 12](#_Toc444082502)

[Table 14 – Project Configurable Parameters eMI STS 17](#_Toc444082503)

[Table 15 – Project Configurable Parameters eMI STSVistAInterface 18](#_Toc444082504)

[Table 16 – Queues 18](#_Toc444082505)

[Table 17 – Abbreviations and Terms 21](#_Toc444082506)

List of Figures

[Figure 1 – VETS – eMI Logical System Overview 4](#_Toc447098554)

[Figure 2 – STS – eMI Deployment Overview 5](#_Toc447098555)

[Figure 3 – VETS to VISTA Sequence 8](#_Toc447098556)

[Figure 4 – VistA to VETS Sequence 9](#_Toc447098557)

[Figure 5 – STS Receiver Flow 13](#_Toc447098558)

[Figure 6 – STS Sender Flow 13](#_Toc447098559)

[Figure 7 – Identify VistA Routing Information Message Flow 14](#_Toc447098560)

[Figure 8 – Identify STS Routing Information Message Flow 15](#_Toc447098561)

[Figure 9 – VistA Receiver Flow 15](#_Toc447098562)

[Figure 10 – HL7 Input Sub-Flow 16](#_Toc447098563)

[Figure 11 – VistA Sender Flow 16](#_Toc447098564)

[Figure 12 – HL7 Output Sub-Flow 17](#_Toc447098565)

[Figure 13 - STS Timeline 20](#_Toc447098566)

Introduction

The Standard & Terminology Service (STS) Enterprise Messaging Infrastructure (eMI) application enables the Department of Veterans Affairs (VA) Enterprise Terminology Services (VETS) Terminology Deployment Service (TDS) to communicate with Veterans Health Information System and Technology Architecture (VistA) packages at any of the 129-plus Veterans Affairs Medical Centers (VAMCs). The STS eMI application interface enables the deployment of standardized clinical terminology to the VAMC by VETS terminology analysts.

Purpose

The purpose of this document is to describe the interface specifications of the eMI STS interface to the VistA systems and the VETS system. This document also provides the detailed design of the eMI STS message flow and includes protocol, transformation, broker patterns, happy path, and exception details.

Scope

The scope of this document is to describe the message flow between the VistA systems and the STS system. This document serves as both requirement and design specification for the STS message flow.

The eMI STS application receives Health Level 7 (HL7) messages from the VistA application and sends HL7 messages to the STS. This terminology is consumed by various domain VistA packages and used in various applications such as the Computerized Patient Record System (CPRS). The terminology is transmitted in HL7 formatted messages to VAMCs. Once the messages have been processed on the VistA side, HL7 acknowledgements (ACKs) are sent back to the TDS. The ACK messages contain the processing results of the deployments that were sent.

The scope items include, but are not limited to:

1. Message Types
2. Validation
3. Ports and Protocol
4. Transformation
5. Error Handling
6. Routing
7. Security

The details of the above are described in the following sections. Also, the common architectural specifications, such as security, logging, exception handling, etc. that apply to all the VIE message flows, are defined in the eMI Software Design Document.

The definitions of the HL7 messages that are generated by VistA to be sent to STS are not in the scope of this document. Details of eMI architecture, physical addresses, and system specification are not in the scope of this document.

Audience

This document provides implementation details for project owners and serves as a blueprint for managers, architects, developers, and testers building the system. It is assumed that the reader has a moderate knowledge of Message Broker and HL7.

References

eMI SDD document on the Department of Veterans Affairs (VA) eMI SharePoint site.

Interface Requirement

Table 1 lists the software interfaces that are implemented.

Table 1 – Software Interfaces

| Application | Interface |
| --- | --- |
| VETS 🡪 eMI Austin Information Technology Center (AITC) | Utilizes Hypertext Transfer Protocol (HTTP) to push the HL7 messages to eMI |
| eMI Regional Data Center (RDC) 🡪 VistA | Utilizes Socket to push the HL7 messages from eMI to VistA |
| VistA 🡪 eMI RDC | Utilizes Socket to push the HL7 messages to eMI |
| eMI AITC 🡪 VETS | Utilizes Socket to push the HL7 messages from eMI to VETS |

Business Unit

Data moves from the VA VistA to the VETS system. Table 2 and Table 3 list the point of contact (POC) information for those systems.

Table 2 – VistA Business Unit

| VistA Business Unit | |
| --- | --- |
| Agency | VA |
| Sending Application | VistA Integrated Funds Distribution, Control Point Activity, Accounting, and Procurement (IFCAP) |
| POC Name | TBD |
| Title | TBD |
| Address | TBD |

Table 3 – VETS Business Unit

| VETS Business Unit | |
| --- | --- |
| Agency | VA |
| Receiving Application | VETS |
| POC Name |  |
| Title | TBD |
| Contact Info |  |

Service Level Agreement Metrics

Table 4 lists the Service Level Agreement (SLA) metrics for the STS message flow that are expected to be met.

Table 4 – SLA Metrics

| SLA Type | SLA Data |
| --- | --- |
| Number of messages/day | 850 messages |
| Average Message size | 50 kilobytes (kb) to 10 megabytes (MB) |
| Data Type | HL 7 v 2.4 |
| Throughput | .01 messages per second |

Message Type Metrics

Table 5 lists the message metrics based on message types.

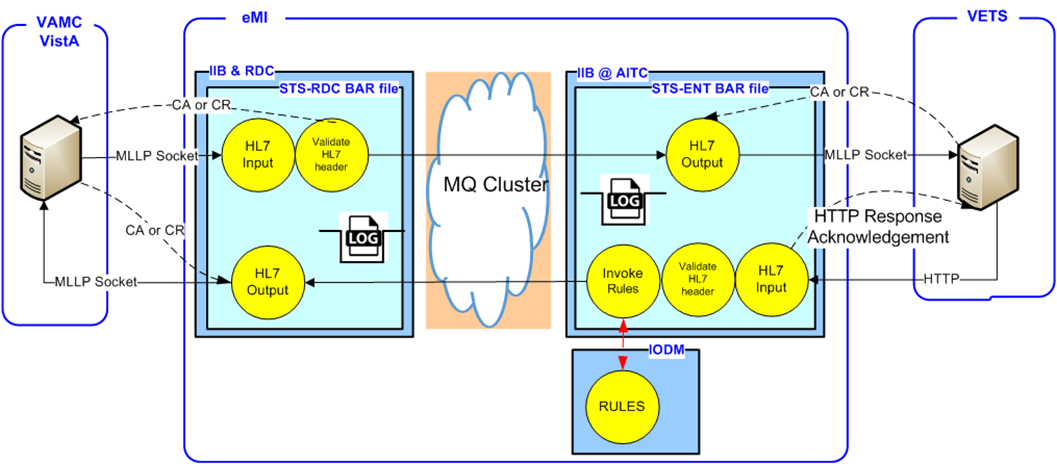
Table 5 – Message Type Metrics

| HL7 Message Type | Estimated Message Size in kb |
| --- | --- |
| MFN^M01 | 50 kb to 10 MB |
| MKK^M01 | 50 kb to 10 MB |
| MFQ^M01 | 50 kb to 10 MB |
| MFR^M01 | 50 kb to 10 MB |

Logical System Overview

Figure 1 shows the VETS/eMI logical system overview. The high-level flow is described following the figure.

Figure 1 – VETS – eMI Logical System Overview

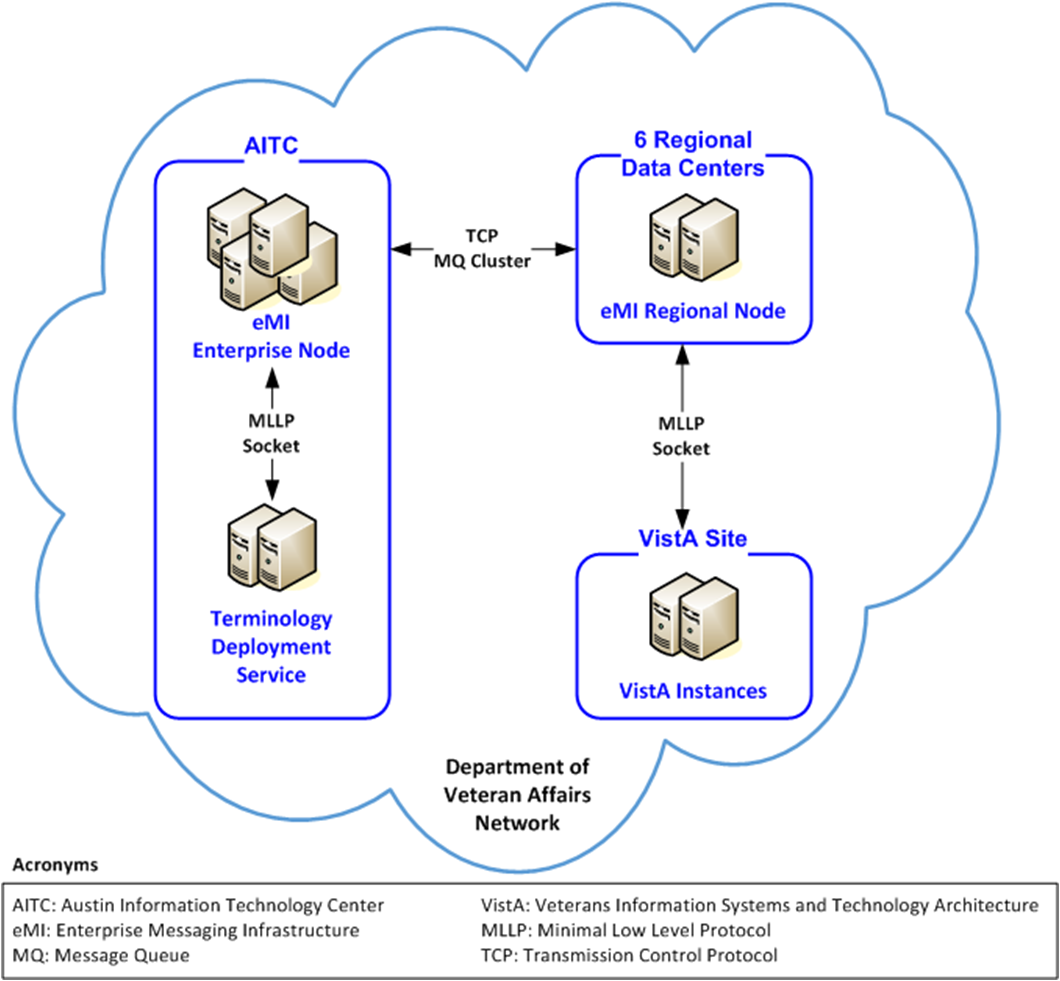


1. The VETS initiates the transfer of an HL7 message via HTTP.
2. The eMI AITC receives the HTTP message and sends the ACK as an HTTP response.
3. The eMI system identifies the routing information, RDC, and the VistA Internet Protocol (IP) and port.
4. The eMI forwards the message from the AITC enterprise system to the RDC.
5. The eMI RDC sends the received HL7 message to the VistA, and receives an Accept Acknowledgement [Commit Acknowledgement (CA)/Commit Rejection (CR)] from VistA.
6. The VistA initiates the transfer of an HL7 message via Socket.
7. The eMI validates the Message Header (MSH) and sends an Accept Acknowledgement (CA/CR) back to the VistA.
8. The eMI forwards the message from the RDC to the AITC enterprise system.
9. The eMI AITC sends the received HL7 message to the VETS, and receives an Accept Acknowledgement (CA/CR) from VETS.

Logical Deployment Overview

The eMI message broker hosts the message flow that listens on a configurable port for HL7 messages from the VA VistA over Transmission Control Protocol (TCP)/Minimal Lower Layer Protocol (MLLP) and routes these messages to the VETS systems. Figure 2 shows the boundaries, gateway, and locations of sending and receiving systems.

Figure 2 – STS – eMI Deployment Overview



STS – eMI Interface Requirements

1. The eMI system shall receive incoming HL7 messages via Socket from the VistA. The eMI system is expected to receive HL7 messages related to clinical activities.
2. The eMI system shall receive incoming HL7 messages via HTTP from the VETS. Once the message is received, the eMI system shall send an acknowledgement after validation as following:

Validates: *<Response><Subject>Successful Response from receiving site</Subject></Response>*

Error: *<Response><Subject>Error Response from the receiving site</Subject><ErrorDescription>{Error description from validation failure on header}</ErrorDescription></Response>*

1. The eMI system shall skip messages that are less than four bytes or messages that start with “SKIP.”
2. The eMI system shall validate the following fields of the MSH segment, as defined in Table 6.

Table 6 – Validation rule for MSH Segment

| MSH field | Description | Validation Rule |
| --- | --- | --- |
| MSH 01 | Field Separator | Is not empty |
| MSH 02.01 | Component Separator | Is not empty |
| MSH 02.02 | Repeat Separator | Is not empty |
| MSH 02.03 | Escape Character | Is not empty |
| MSH 02.04 | Sub Component Separator | Is not empty |
| MSH 03 | Sending Application | Is not empty |
| MSH 04 | Sending Facility | Is not empty |
| MSH 05 | Receiving Application | Is not empty |
| MSH 06 | Receiving Facility | Is not empty |
| MSH 07 | Date/Time of Message | Is not empty |
| MSH 09 | Message Type | Is not empty |
| MSH 10 | Message Control Id | Is not empty |
| MSH 11 | Processing Id | Is not empty |
| MSH 12 | Version Id | Is not empty |

1. The eMI system shall generate a CR and send the message back to the VistA if the validation of the MSH segment fails. The eMI system shall not send a CR to the STS.
2. The eMI system shall route the message received from VistA to the VETS system based on the MSH 05 (Receiving Application) and MSH 06 (Receiving Facility) of the incoming message.
3. The eMI system shall route the message received from the VETS system to the VistA based on the MSH 05 (Receiving Application) and MSH 06 (Receiving Facility) of the incoming message.
4. The eMI system shall externalize the mapping of the MSH 05, MSH06, receiving application’s host name, and receiving application’s port number (defined as routing rules) to the Operational Decision Management (ODM) so that they can be updated without requiring the eMI STS applications to redeploy to the IBM Integration Broker.

Table 7 – STS/VistA Routing Table (Sample)

| Receiving Facility | Receiving Application | Message Queue (MQ0 Queue Manager | MQ Queue Name | Host Name | Host Port |
| --- | --- | --- | --- | --- | --- |
| 200 | VETS UPDATE |  |  |  |  |
| VETS DATA |  |  |  |  |
| VETS MD5 |  |  |  |  |
| 500 | XUMF DATA |  | VIE.STS.VISTA.RDC1 |  |  |
| XUMF MD5 |  | VIE.STS.VISTA.RDC1 |  |  |
| XUMF UPDATE |  | VIE.STS.VISTA.RDC1 |  |  |
| 657 | XUMF DATA |  | VIE.STS.VISTA.RDC2 |  |  |
| XUMF MD5 |  | VIE.STS.VISTA.RDC2 |  |  |
| XUMF UPDATE |  | VIE.STS.VISTA.RDC2 |  |  |

1. In the event the eMI STS application is unable to determine the routing rules from ODM, an error message will be logged.
2. In the event the eMI STS application is unable to publish the message to either the VETS or the VistA, it will try for a configurable number of times and then notify an administrator via email.

Nominal VETS Message Flow

Figure 3 shows the message flow between the VETS and the VistA systems using the eMI.

Figure 3 – VETS to VISTA Sequence



1. The VETS initiates the transfer of a HL7 message over HTTP.
2. The eMI validates the MSH; if valid, it then pushes the message to the next flow to identify routing.
3. The eMI sends the ACK as an HTTP response.
4. The eMI identifies the routing information. It identifies the VistA instance for message delivery and the queue of the RDC for the VistA instance it’s associated with.
5. The eMI AITC pushes the event to the eMI RDC. This push occurs as part of the Message Queue (MQ) cluster.
6. The eMI delivers the HL7 message to the VistA via Socket.
7. The eMI receives a CA/CR from the VistA.

Error Flow: For any type of exception that the message flow encounters, the eMI sends an email with the exception details. If VistA is unavailable, then eMI will keep retrying until it is able to deliver the message. If eMI is unable to identify routing information, then it will send a notification to an administrator.

Figure 4 shows the message flow between the VistA and the VETS systems using the eMI.

Figure 4 – VistA to VETS Sequence



1. The VistA initiates the transfer of an HL7 message via Socket.
2. The eMI validates the MSH.
3. The eMI returns a CA if it does not encounter an error during validation; otherwise, it returns a CR.
4. The eMI RDC pushes the event to the eMI AITC. This push occurs as part of the MQ cluster.
5. The eMI delivers the HL7 message to the VETS via Socket.
6. The eMI receives a CA/CR from the VETS.

Error Flow: For any type of exception that the message flow encounters, the eMI sends an email with the exception details. If VETS is unavailable, then eMI will keep retrying until it is able to deliver the message.

STS Design

The following sections address STS design aspects.

Architecture Deviations

The STS message flow uses an HL7-to-HL7 Data Format Description Language (DFDL) pattern. Instead of using the HL7DFDLInput node, the STS message flow uses the source of the HL7DFDLInput node as a sub-flow; similarly, instead of using HL7DFDLOutput node, the STS message flow uses the source of the HL7DFDLOutput node as a sub-flow. Modifications ensure that the message flow meets VA-specific requirements for validation. There is no known impact on the performance or functionality.

Pattern

The STS message flow is an instance of an HL7-to-HL7 DFDL pattern. The following options are unchecked in the tool configuration due to the specific requirements of the STS interface.

* No sequencing
* No journaling
* Unchecked canonical feed, report remainders, check duplicates
* Publish to queue

Protocol

The STS integration service uses the protocols described in Table 8, Table 9, Table 10, and Table 11 to interface with the sending and receiving systems.

Table 8 – VistA to eMI Interface

| VistA to eMI Interface | |
| --- | --- |
| Protocol: | MLLP over TCP/IP |
| Message Type: | HL7 |
| VistA hostname: | VistA system |
| eMI hostname: | RDC Load balancer |

Table 9 – eMI Interface to VETS

| eMI Interface to VETS | |
| --- | --- |
| Protocol: | MLLP over TCP/IP |
| Message Type: | HL7 |
| eMI hostname: | AITC Message brokers |
| VETS hostname: | VETSHost |

Table 10 – VETS to eMI Interface

| VETS to eMI Interface | |
| --- | --- |
| Protocol: | HL7 over HTTP |
| Message Type: | HL7 |
| VETS hostname: | VETS system |
| eMI hostname: | AITC Load balancer |

Table 11 – eMI Interface to VistA

| eMI Interface to VistA | |
| --- | --- |
| Protocol: | MLLP over TCP/IP |
| Message Type: | HL7 |
| eMI hostname: | VistA system |
| VETS hostname: | VistAHost |

Message Routing

Content-based routing is utilized for routing messages from the VETS system to various VistA instances. Depending upon the value of the receiving facility (MSH-06) and receiving application (MSH-05), the messages are first routed to various RDCs, then the RDCs use the hostname and port information to route to the given VistA instance. Table 12 provides the sample mapping of the routing RDC queues and VistA hostname and ports. Once development is completed, this table will be updated.

Table 12 – STS Queue Routing Table (Sample)

| Application | Receiving Facility | Receiving Application | RDC Queue | VistA Host and Port |
| --- | --- | --- | --- | --- |
| VETS | 660 | XUMF\_UPDATE | VIE.STS.RDC1.VISTA |  |
| XUMF\_MD5 | VIE.STS.RDC1.VISTA |  |
| 565 | XUMF DATA | VIE.STS.RDC1.VISTA |  |
| XUMF MD5 | VIE.STS.RDC1.VISTA |  |

Transformation

The following sections document the various transformations for the STS message flow.

Protocol Transformation

Table 13 lists the various protocol transformations occurring in the STS message flow.

Table 13 – Protocol Transformation in STS

| Application | Protocol |
| --- | --- |
| VistA 🡪 eMI RDC | MLLP over TCP/IP |
| eMI RDC 🡪 eMI Enterprise | MQ Secure Sockets Layer (SSL) channel |
| eMI Enterprise 🡪 VETS | MLLP over TCP/IP |
| VETS 🡪 eMI Enterprise | HTTP |
| eMI Enterprise 🡪 eMI RDC | MQ SSL channel |
| eMI RDC 🡪 VistA | MLLP over TCP/IP |

Data Transformation

Data transformation is not applicable to the STS message flow.

Implementation Details

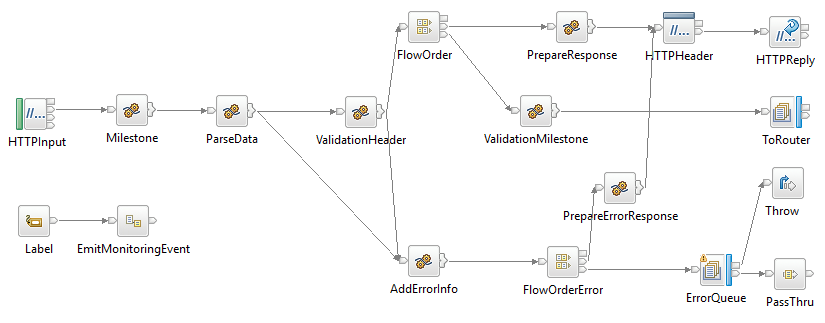
For the STS message flow, there are two components. One component runs on the RDC, which receives and sends messages to and from VistA, and the second component runs on the enterprise system, which sends and receives the messages to and from the VETS system.

STS

This component is deployed at the eMI AITC and has four sub flows: The STS\_HTTPReceiver, STS\_STS\_MLLPSender, STS\_MLLPRouterSTS and STS\_MLLPRouterVistA. The STS\_HTTPReceiver receives an HL7 message from the VETS system over HTTP. The STS\_STS\_MLLPSender subscribes to the queue for HL7 messages that are sent by various VistA sites at the RDCs, translates the messages to HL7 MLLP, and delivers them to STS over TCP/IP. The STS\_MLLPRouterVistAroutes the received message from VETS to the RDC to be delivered to VistA.

STS\_HTTPReceiver

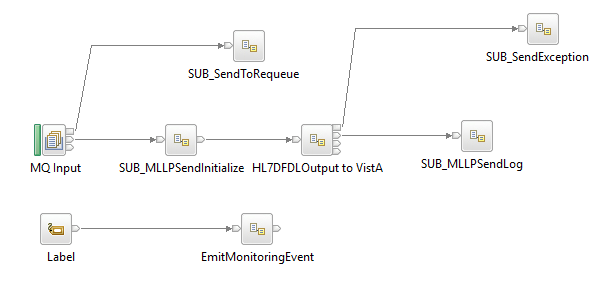
Figure 5 – STS Receiver Flow



The message flow receives HL7 messages from the VETS system over HTTP. The flow performs a basic validation of the MSH segment. The flow returns the ACK as an HTTP response. The flow translates the received message to an MQ DFDL HL7 message and publishes it to the queue.

STS\_STS\_MLLPSender

Figure 6 – STS Sender Flow



The message flow receives the message from eMI RDC and delivers it to the VETS system. The message payload is an HL7 message that is received from VistA. As Figure 6 shows, the receiver flow of the HL7-to-HL7 DFDL pattern is updated in the following manner to support the interface requirements.

* Update to log milestone information
* Replace HL7DFDLOutput node with HL7DFDLOutput message sub-flow to avoid treating the Accept Acknowledgement of type CR, Commit Error (CE), Application Rejection (AR), and Application Error (AE) as exceptions and prevent retries (Figure 12 shows the HL7DFDLOutput message sub-flow)

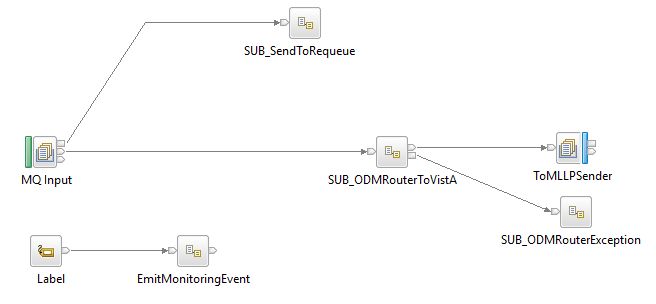
If there is an unexpected exception, then the message flow sends an email notification with the exception details to an administrator.

STS\_MLLPRouter

There are two message flows: STS\_MLLPRouterVistA and STS\_MLLPRouterSTS.

The STS\_MLLPRouterVistA flow determines the destination RDC, hostname, and IP for the VistA instance on the receiving facility (MSH-06) and receiving application (MSH-05). The routing rules are defined in the decision table in the Operational Decision Manager (ODM). The flow uses the SUB\_ODMRouterToVistA to connect to the ODM to retrieve and execute the rules. It also caches them in the memory. Figure 7 shows the message flow that is used to load the configuration.

Figure 7 – Identify VistA Routing Information Message Flow



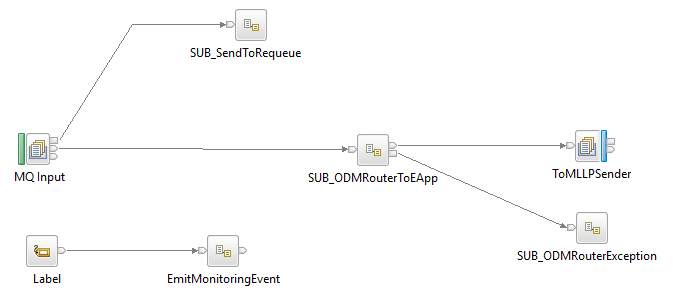
The decision table, Table 12, maps the receiving facility (MSH-06) and receiving application (MSH-05) to the destination queue for the RDC, host, and port for VistA. ODM defines and maintains this decision table.

If there is an unexpected exception, then the message flow sends an email notification with the exception details to an administrator.

Similarly, the STS\_MLLPRouterSTS flow determines the VETS hostname and IP based on the receiving facility (MSH-06) and receiving application (MSH-05). Figure 12 shows the message flow that is used to load the configuration.

If there is an unexpected exception, then the message flow sends an email notification with the exception details to an administrator.

Figure 8 – Identify STS Routing Information Message Flow

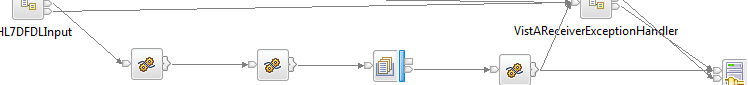


STS VistA Interface

This component gets deployed at all RDCs. It has a VistA Receiver that receives an HL7 message from VistA using MLLP over TCP/IP and a VistA Sender that receives a message from eMI AITC and sends it to VistA using MLLP over TCP/IP.

STS\_MLLPReceiver

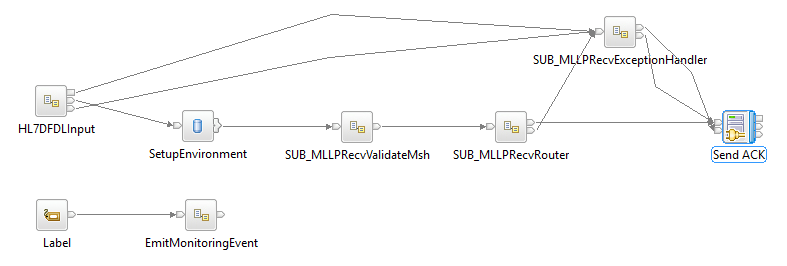
Figure 9 – VistA Receiver Flow



The message flow receives HL7 messages from VistA using MLLP over TCP/IP. The flow performs basic validation of the MSH segment and sends an Accept Acknowledgement (CA/CR) back to VistA. The flow translates the received message to an MQ DFDL HL7 message and publishes it to the enterprise STS queue. As Figure 8 shows, the receiver flow of the HL7 to HL7 DFDL pattern is updated in the following manner to support the interface requirements.

* Update ‘Receiver Flow’ of the HL7-to-HL7 DFDL pattern to:
* Update the MSH field validation
* Update to support return of Accept Acknowledgement as a CA and/or CR to VistA
* Update to log milestone information
* Replace HL7DFDLInput node with HL7DFDLInput message sub-flow to suppress validation of HL7 segments in received HL7 messages against the HL7v251 message definition set (Figure 10 shows the HL7DFDLInput message sub-flow)
* Update to handle invalid messages by sending an alert email to an administrator

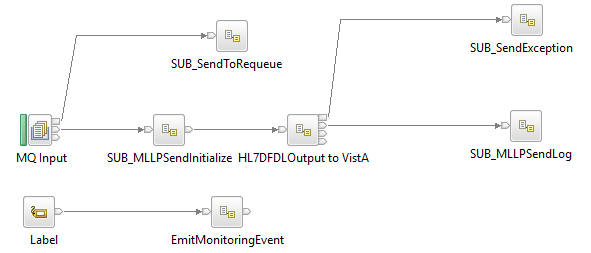
Figure 10 – HL7 Input Sub-Flow



If there is an unexpected exception, then the message flow sends an email notification with the exception details to an administrator.

STS\_VISTA\_MLLPSender

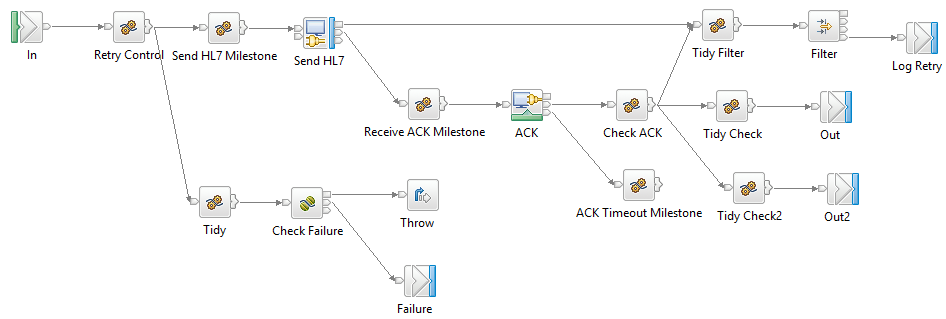
Figure 11 – VistA Sender Flow



The message flow receives the message from the eMI AITC and it is delivered to the respective VistA instance. The message payload is an HL7 message that is received from the VETS system along with the host and port information of the VistA instance. As Figure 11 shows, the receiver flow of the HL7-to-HL7 DFDL pattern is updated in the following manner to support the interface requirements.

* Update ‘Sender Flow’ of the HL7-to-HL7 DFDL pattern to:
* Dynamically set the hostname and port for the output node
* Update to log milestone information
* Replace HL7DFDLOutput node with HL7DFDLOutput message sub-flow to avoid treating the Accept Acknowledgement of type CR, CE, AR, and AE as exceptions and prevent retries (Figure 12 shows the HL7DFDLOutput message sub-flow)

Figure 12 – HL7 Output Sub-Flow



* Update to keep retrying to send messages when VistA is unavailable
* Update to handle generating an alert email to an administrator when the flow receives an Accept Acknowledgement (CR and CE)

If there is an unexpected exception, then the message flow sends an email notification with the exception details to an administrator.

Error Handling For STS/VistA Message Delivery

If either VistA or VETS is unavailable, then the flow uses the re-queuing mechanism. Undelivered messages would be published to the retry queue and would be retried for delivery at a configurable interval until they are delivered successfully.

Project Configuration File

Table 14 and Table 15 lists the project configuration file details that are either environment-specific or control the flow of messages.

Table 14 – Project Configurable Parameters eMI STS

| Property | Default Value | Purpose |
| --- | --- | --- |
| healthcare.STS\_MLLPReceiver#HL7DFDLInput.connectionDetails | localhost | Port on which eMI would listen for messages coming from VETS |
| router.RouterToSTS#RetryWaitTime | 3000 | Wait time in milliseconds between each retry to identify routing information for VETS messages |
| router.RouterToVistA#RetryWaitTime | 3000 | Wait time in milliseconds between each retry to identify routing information for VistA messages |
| requeue.Requeue#delay | 30 | Wait time in seconds between each retry for all the re-queued messages |
| requeue.Requeue#retryCount | 15 | Number of retries |
| sender.STSSender#RetryWaitTime | 3000 | Wait time in milliseconds between each retry to publish message to VETS |
| sender.STS\_VISTA\_MLLPSender#MQInput.queueName | VIE.ENTERPRISETOVISTA.CHANGEME | MQ Input Queue |
| sender.STS\_VISTA\_MLLPSender#ReplyQueue | VIE.ENTERPRISETOVISTA.CHANGEME | VistA reply queue |

Table 15 – Project Configurable Parameters eMI STSVistAInterface

| Property | Default Value | Purpose |
| --- | --- | --- |
| healthcare.STSVistAReceiver#HL7DFDLInput.connectionDetails | localhost | Port on which eMI would listen for messages coming from VA |
| sender.VistASender#MQ input.queueName | VIE.STS.VISTA.RDC1 | Queue to which STSVistAInterface subscribes to send messages to VistA |
| sender.VistASender#RetryWaitTime | 3000 | Wait time in milliseconds between each retry to publish message to VistA |
| requeue.Requeue#delay | 30 | Wait time in seconds between each retry for all the re-queued messages |
| requeue.Requeue#retryCount | 15 | Number of retries |

Queue Details

Table 16 lists the queues that the STS message flows use.

Table 16 – Queues

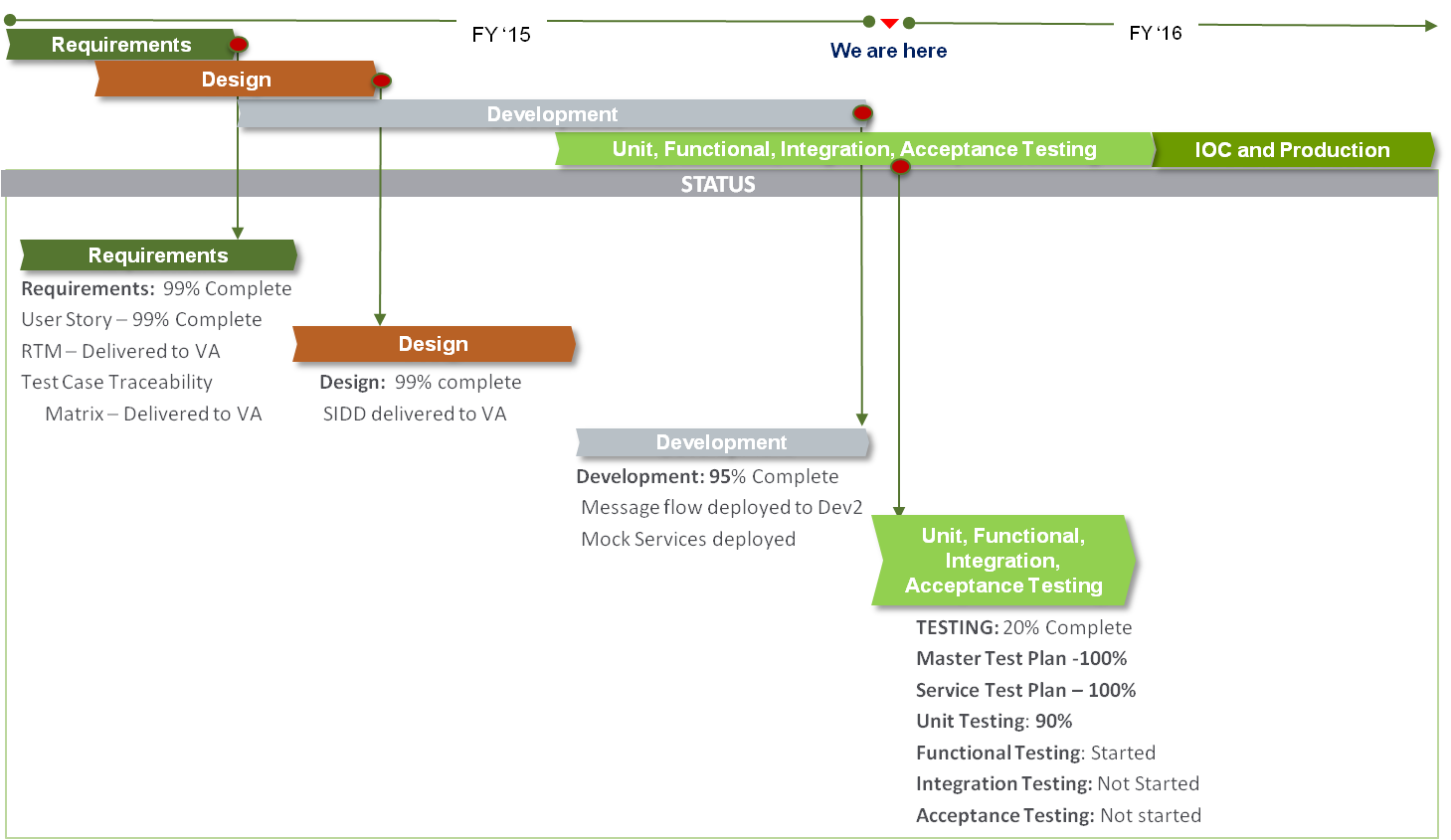
| Queue Name | Queue Location | Purpose |
| --- | --- | --- |
| EMI.SUBSCRIBE.EVENTS.ENT.L | AITC | Monitoring Subscription |
| VIE.STS.VISTATOSTS.SRC.ENT.C | CLUSTER  AITC | The queue for messages that are coming from VistA |
| VIE.STS.STSTOVISTA.SRC.ENT.L | AITC | The queue for messages that are coming from VETS |
| VIE.STS.STSRECEIVER.ERROR.ENT.L | AITC | The error queue that stores the messages that was not able to process in the STS receiver flow |
| VIE.STS.VISTATOSTS.TAR.ENT.L | AITC | The queue for messages to be sent to VETS |
| VIE.STS.REQUEUE.RETRY.ENT.L | AITC | The queue for messages to be retried |
| VIE.STS.REQUEUE.MAXRETRIED.ENTC.L | AITC | The queue for messages that have exhausted max retries |
| VIE.STS.REQUEUE.ERROR.ENT.L | AITC | The queue for messages placed for retry that cannot be processed correctly |
| EMI.SUBSCRIBE.EVENTS.RDC.L | RDC | Monitoring Subscription |
| VIE.STS.STSTOVISTA.RDC1.TAR.RDC.C | CLUSTER RDC | The queue for messages to be sent to RDC for VistA delivery |
| VIE.STS.VISTARECEIVER.ERROR.RDC.L | RDC | The error queue that stores the messages that was not able to process in the VistA receiver flow |
| VIE.STS.REQUEUE.RETRY.RDC.L | RDC | The queue for messages to be retried |
| VIE.STS.REQUEUE.MAXRETRIED.RDC.L | RDC | The queue for messages that have exhausted max retries |
| VIE.STS.REQUEUE.ERROR.RDC.L | RDC | The queue for messages placed for retry that cannot be processed correctly |

Timeline

Figure 13 shows the timeline for the STS Implementation.

The VIE implementation timeline is in the process of being updated. Once completed, a new diagram will be inserted.

Figure 13 - STS Timeline



Acronyms

Table 17 – Acronyms

| Acronym/Term | Definition |
| --- | --- |
| ACK | Acknowledgement |
| AE | Application Error |
| AITC | Austin Information Technology Center |
| AR | Application Rejection |
| CA | Commit Acknowledgement |
| CE | Commit Error |
| CPRS | Computerized Patient Record System |
| CR | Commit Rejection |
| DFDL | Data Format Description Language |
| eMI | Enterprise Messaging Infrastructure |
| ESB | Enterprise Service Bus |
| HL7 | Health Level Seven |
| IFCAP | Integrated Funds Distribution, Control Point Activity, Accounting, and Procurement |
| IP | Internet Protocol |
| kb | Kilobyte |
| MB | Megabyte |
| MLLP | Minimal Lower Layer Protocol |
| MQ | Message Queue |
| MSH | Message Header |
| ODM | Operational Decision Manager |
| POC | Point of Contact |
| RDC | Regional Data Center |
| SLA | Service Level Agreement |
| SSL | Secure Sockets Layer |
| STS | Standard & Terminology Service |
| TCP | Transmission Control Protocol |
| TDS | Terminology Deployment Service |
| VA | Department of Veterans Affairs |
| VAMC | VA Medical Center |
| VETS | VA Enterprise Terminology Services |
| VIE | VistA Interface Engine |
| VistA | Veterans Health Information System and Technology Architecture |

1. Architecture Design Decisions

The STS message flow has not deviated from any recommended or standard patterns defined by IBM or Enterprise Shared Services.

1. Messages
   1. Message Type

HL7 v2.4 messages

* 1. Sample Messages

**Sample Message 1:**

MSH^~|\&^VETS UPDATE^660INT^XUMF UPDATE^552^20060731085700.000-0600^^MFN~M01^20060731903543910^T^2.4^^^AL^AL^USA

MFI^Standard Terminology~~ERT^^MUP^20060731085700.000-0600^20060731085700.000-0600^NE

MFE^MUP^^^Reactants@4538714

ZRT^Term^SUCRALOSE

ZRT^Allergy\_Type^FOOD

ZRT^Search\_Term^SPLENDA

ZRT^has\_drug\_ingredient^""

ZRT^has\_drug\_class^""

ZRT^Status^1

MFE^MUP^^^Reactants@4693079

ZRT^Term^CHIGGERS

ZRT^Allergy\_Type^OTHER

ZRT^Search\_Term^JIGGERS

ZRT^has\_drug\_ingredient^""

ZRT^has\_drug\_class^""

ZRT^Status^1

MFE^MUP^^^Reactants@4693262

ZRT^Term^TREES

ZRT^Allergy\_Type^OTHER

ZRT^Search\_Term^""

ZRT^has\_drug\_ingredient^""

ZRT^has\_drug\_class^""

ZRT^Status^1

MFE^MUP^^^VERSION

ZRT^version^5

**Sample Message 2:**

MSH^~|\&^VETS MD5^660INT^XUMF MD5^552^20060731090000.000-0600^^MFQ~M01^20060731904415710^T^2.4^^^AL^AL^USA

QRD^20060731090000.000-0600^R^I^Standard Terminology Query^^^99999^ALL^Reactants^VA

**Sample Message 3:**

MSH^~|\&^XUMF MD5^552^VETS MD5^660INT^20060731131557-0500^^MFR~M01^55252377900^T^2.4^^^AL^NE^USA

MSA^AA^2006073111153958010^;CHECKSUM:7a96e54e332d06145f3ed8ebcd05e5d4;VERSION:23;

QRD^20060731111500.000-0600^R^I^Standard Terminology Query^^^99999^ALL^Nature of Order^VA

**AA Example:**

MSH^~|\&^XUMF UPDATE^442^VETS UPDATE^660INT^20060731134108-0400^^MFK~M01^44210936024^T^2.4^^^AL^NE^USA

MSA^AA^2006073111405085510^

**AE Example:**

MSH^~|\&^XUMF UPDATE^442^VETS UPDATE^660INT^20060731132002-0400^^MFK~M01^44210936012^T^2.4^^^AL^NE^USA

MSA^AE^2006073111194775010^data validation error

**CA Example:**

MSH^~|\&^VETS UPDATE^660INT^XUMF UPDATE^442^20060731120000.000-0600^^MFK^20060731114104^T^2.4

MSA^CA^44210936024

1. Message Mapping

None

1. Outstanding Issues

Closed PMR: The following PMR has been resolved.

* PMR #:30667.442.000
* Brief Description about the problem: Unable to secure IIB -> WODM database connection

1. Approval Signature

REVIEW DATE:

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Integrated Project Team (IPT) Chair Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Business Sponsor Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT Program Manager Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Manager Date