Enterprise Messaging Infrastructure

Veteran Interoperability and Integrations

Bidirectional Health Information Exchange

Message Flow

Service Integration Design Document



Department of Veterans Affairs

**April 2016**

Version 1.0

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 01/06/2015 | 1.0 | Tech Writer Review and Format |  |
| 02/24/2016 | 1.0 | Tech Writer Review and Format |  |
| 03/01/2016 | 1,0 | Updated contents and diagrams |  |
| 04/01/2016 | 1.0 | Tech Writer Review and Format |  |

Table of Contents

[1.0 Introduction 1](#_Toc447270685)

[1.1 Purpose 1](#_Toc447270686)

[1.2 Scope 1](#_Toc447270687)

[1.3 Audience 2](#_Toc447270688)

[1.4 References 2](#_Toc447270689)

[2.0 Interface Requirements 2](#_Toc447270690)

[2.1 Business Unit 2](#_Toc447270691)

[2.2 Service Level Agreement Metrics 3](#_Toc447270692)

[2.3 Message Type Metrics 3](#_Toc447270693)

[2.4 Logical System Overview 3](#_Toc447270694)

[2.5 Logical Deployment Overview 4](#_Toc447270695)

[2.6 BHIE - eMI Interface Requirements 5](#_Toc447270696)

[3.0 Nominal BHIE Message Flow 6](#_Toc447270697)

[4.0 BHIE Message Flow Design 6](#_Toc447270698)

[4.1 Architecture Deviations 6](#_Toc447270699)

[4.2 Pattern 7](#_Toc447270700)

[4.3 Protocol 7](#_Toc447270701)

[4.4 Message Routing 7](#_Toc447270702)

[4.5 Transformation 7](#_Toc447270703)

[4.5.1 Protocol Transformation 7](#_Toc447270704)

[4.5.2 Protocol Transformation 8](#_Toc447270705)

[5.0 Implementation Details 8](#_Toc447270706)

[5.1 BHIE Message Flow 8](#_Toc447270707)

[5.2 Receiver Flows 10](#_Toc447270708)

[5.3 Error Handling Flows 10](#_Toc447270709)

[5.4 Project Configuration File 11](#_Toc447270710)

[5.5 Queue Details 11](#_Toc447270711)

[6.0 Timeline 11](#_Toc447270712)

[7.0 Acronyms 12](#_Toc447270713)

[Appendix A. Architecture Design Decisions 13](#_Toc447270714)

[Appendix B. Message Flow Documentation 14](#_Toc447270715)

[Appendix C. Message Mapping 15](#_Toc447270716)

[Appendix D. Outstanding Issues 16](#_Toc447270717)

[Appendix E. Approval Signature 17](#_Toc447270718)

List of Tables

[Table 1 - Logical High-Level Message Transmission Flows 2](#_Toc447270719)

[Table 2 – Station 200 Business Unit 2](#_Toc447270720)

[Table 3 – DMIX DES Business Unit 3](#_Toc447270721)

[Table 4 – SLA Metrics 3](#_Toc447270722)

[Table 5 – Message Type Metrics 3](#_Toc447270723)

[Table 6 - VA Consumer to eMI Interface 7](#_Toc447270724)

[Table 7 - eMI Interface to DMIX/DES 7](#_Toc447270725)

[Table 8 - Project Configurable Parameters 11](#_Toc447270726)

[Table 9 - Acronyms 12](#_Toc447270727)

List of Figures

[Figure 1 – DMIX DES – eMI Logical System Overview 4](#_Toc447270728)

[Figure 2 – DMIX – eMI Deployment Overview 5](#_Toc447270729)

[Figure 3 - Nominal BHIE Message Flow 6](#_Toc447270730)

[Figure 4 - eMI BHIE Message Flow 8](#_Toc447270731)

[Figure 5 - Dispatch HTTP Subflow 9](#_Toc447270732)

[Figure 6 - Reply HTTP Subflow 10](#_Toc447270733)

[Figure 7 - Implementation Timeline 11](#_Toc447270734)

# Introduction

The Bidirectional Health Information Exchange (BHIE) Enterprise Messaging Infrastructure (eMI) application integrates the eMI with the Defense Medical Information Exchange (DMIX) Data Exchange Service (DES) to enable data sharing of crucial Department of Defense (DoD) patient records for use by doctors and other clinicians with VA systems through eMI gateway. These health records can be searched via patient identifier, Logical Observation Identifiers Names and Codes (LOINC), and desired information source(s). The DMIX/DES supports the exchange of patient clinical data with the DoD, the VA, and other federal and commercial entities using eHealth Exchange.

## Purpose

The purpose of this document is to describe the interface specifications between VA service consumers, the eMI service proxies for DMIX/DES, and the DoD DMIX/DES service provider. This document also provides the detailed design of the eMI service proxy message flows and includes protocol, transformation, broker patterns, happy path, and exception details. Detailed governance integration requirements, including SOA Service endpoints, required service registry artifacts, and governance policies, will be described as well.

## Scope

The scope of this Service Integration Design Document (SIDD) is to address interoperability between VA consumer applications (e.g., VistA Exchange, Station 200, etc.) and the DoD DMIX/DES service provider, by means of eMI proxies for the DMIX/DES v4 REST API.

The eMI BHIE application receives REST Web Service calls from VA consumer applications and calls DMIX/DES REST API version 4.0.

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. The common architectural specifications, such as security, logging, exception handling, and etc. are defined in the eMI Software Design Document; however, the definitions of the DMIS DES REST API, and 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 readers have a moderate knowledge of IBM Integration Bus (IIB) and RESTful web services.

## References

eMI Software Design Document is accessible on the VA eMI SharePoint site.

# Interface Requirements

Table 1 lists the software interfaces that are implemented.

Table 1 - Logical High-Level Message Transmission Flows

| Application | Interface |
| --- | --- |
| VA Service Consumer 🡪 eMI | Utilizes REST web service messaging endpoint to transmit a DMIX/DES GET request over HTTPS transport protocol to an eMI REST web service messaging endpoint. |
| eMI 🡪 DMIX/DES | Utilizes REST web service messaging endpoint to relay a the inbound DMIX/DES GET request over HTTPS transport protocol to a DMIX/DES service messaging endpoint. |
| DMIX/DES 🡪 eMI | DMIX/DES responds to the request from eMI over HTTPS transport protocol with a JSON message containing the DMIX/DES response payload. |
| eMI 🡪 VA Service Consumer | eMI relays the outbound DMIX/DES response payload over HTTPS transport protocol back to the VA Service Consumer. |

## Business Unit

Data moves from the VA consumer applications such as Station 200 to the DMIS DES. Table 2 and Table 3 list the point of contact (POC) information for those systems.

Table 2 – Station 200 Business Unit

| VA Business Unit | |
| --- | --- |
| Agency | VA |
| Sending Application | Station 200 |
| POC Name |  |
| Title | BHIE Interagency Systems Manager |
| Address |  |

Table 3 – DMIX DES Business Unit

| VETS Business Unit | |
| --- | --- |
| Agency | DoD |
| Receiving Application | DMIX DES |
| POC Name |  |
| Title | DMIX Program Manager |
| Contact Info |  |

## Service Level Agreement Metrics

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

Table 4 – SLA Metrics

| SLA Type | SLA Data |
| --- | --- |
| Number of messages/day | TBD |
| Average Message size | 50 kilobytes (kb) to 10 megabytes (MB) |
| Data Type | REST Web Service |
| Throughput | TBD |

## Message Type Metrics

Table 5 lists the message metrics based on message types.

Table 5 – Message Type Metrics

| Message Type | Estimated Message Size |
| --- | --- |
| REST Web Service | 50 kb to 10 MB |

## Logical System Overview

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

Figure 1 – DMIX DES – eMI Logical System Overview



1. The VA service consumer system initiates a REST Web Service call to CA API Gateway hosted on eMI instance at AITC
2. CA API Gateway forwards request to eMI BHIE message flow on IIB.
3. The eMI BHIE message flow mediates input data if necessary to conform to DMIX DES REST API v4.0.
4. The eMI BHIE message flow invokes routing rules if necessary to route to DMIX DES system.
5. The eMI BHIE message flow queries WebSphere Service Registry and Repository (WSRR) for DMIX DES REST API endpoint which includes server name and port.
6. The eMI BHIE message flow forward a REST Web Service call to DMIX DES service.

## Logical Deployment Overview

The eMI message broker hosts the message flow that listens on a configurable port for REST Web Service call from VA consumer application over Hyper Text Transmission Protocol (HTTP) and routes the call to DMIX DES. Figure 2 shows the boundaries, gateway, and locations of sending and receiving systems.

Figure 2 – DMIX – eMI Deployment Overview

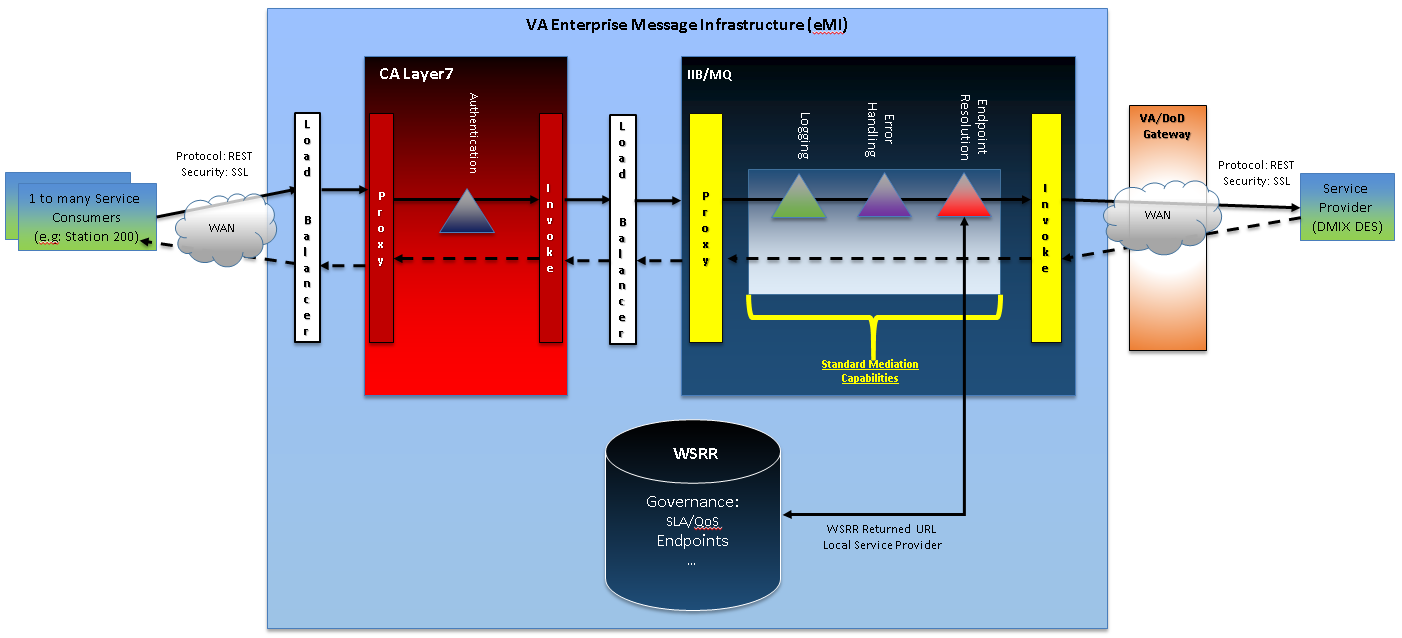


## BHIE - eMI Interface Requirements

1. The eMI system shall use the DoD’s DMIX/DES system as the data source for its BHIE message flow.
2. The eMI system shall receive outbound data request via HTTPS from VA consumer application.
3. The eMI system shall receive inbound data response via HTTPS from DMIX DES system.
4. The eMI system shall interface with DMIX DES system through its BHIE Message Flow using DMIX DES REST API v4.0.
5. The eMI system shall ensure a secured connection with DMIS DES system by means of two-way Secure Sockets Layer (SSL) encryption.

# Nominal BHIE Message Flow

Figure - Nominal BHIE Message Flow



1. The VA service consumer issues a REST request for DMIX/DES data to the eMI proxy, via the CA Layer 7 gateway.
2. The VA service consumer issues a REST request for DMIX/DES data to the eMI proxy, via the CA Layer 7 gateway.
3. CA Layer 7 assigns an internal eMI identifier to the request and forwards to IIB for message routing.
4. IIB obtains the DMIX/DES service provider REST endpoint from WSRR. WSRR REST service endpoint values may be cached in the IIB runtime to ensure performance requirements are met.
5. IIB relays the request to the DMIX/DES service provider REST endpoint. IIB does not modify the request payload.
6. IIB relays the DMIX/DES service provider response to the VA service consumer, via the CA Layer 7 gateway. IIB does not modify the response payload.

Errors originating within the DMIX/DES service provider system are relayed unchanged to the VA service consumer. Errors originating within the eMI system are sent to the VA service consumer, either appended to or in place of the response payload, as appropriate.

# BHIE Message Flow Design

The following section refers to the design of the eMI message flows which proxy requests to the DMIX/DES service provider. Further information about request and response formats can be found in the service provider’s ICD, DMIX-External-ICD-v3-draft-11162015.pdf.

## Architecture Deviations

The DMIX/DES proxy message flows do not contain any architectural deviations.

## Pattern

The DMIX/DES proxy message flows are loosely based on the IBM Service Proxy pattern. See section 5 for a detailed explanation of the flow.

## Protocol

The DMIX/DES proxy implementation uses the protocols described in Table 6 and Table 7 to interface with the sending and receiving systems.

Table 6 - VA Consumer to eMI Interface

| VA Consumer to eMI | |
| --- | --- |
| Protocol | HTTPS |
| Message Type | Messages are in JSON format and conform to the DMIX/DES v4 REST API. |
| VA Consumer Hostname | Multiple |
| eMI Hostname | Austin Information Technology Center (AITC) Load balancer. |

Table 7 - eMI Interface to DMIX/DES

| eMI To DMIX/DES | |
| --- | --- |
| Protocol | HTTPS |
| Message Type | Messages are in JSON format and conform to the DMIX/DES v4 REST API. |
| eMI Hostname | AITC Message brokers |
| DMIX/DES Hostname | Determined dynamically by means of WSRR REST Service endpoint lookup. |

## Message Routing

Requests to the eMI BHIE Message Flow are routed by means of dynamic WSRR REST Service endpoint lookup. The destination URL for the DMIX/DES service provider is obtained from the WSRR governance repository. Subsequently, the message flow uses the destination URL to route the inbound request to the proper REST endpoint within the DMIX/DES service provider REST API.

## Transformation

The following sections document the various transformations for the eMI BHIE message flow.

### Protocol Transformation

Protocol transformation is not applicable to the eMI BHIE message flow.

### Protocol Transformation

Data transformation is not applicable to the eMI BHIE message flow.

# Implementation Details

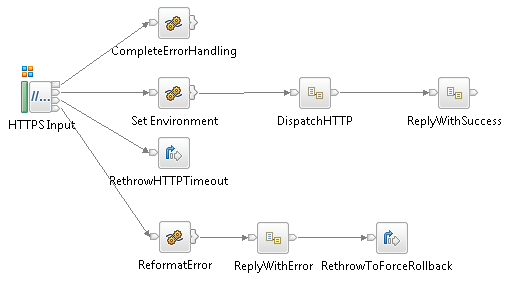
For the eMI BHIE message flow, there is only one component that runs on the eMI Enterprise Node at AITC which receives outbound data requests from VA systems inside the VA network, sends data requests to DMIS DES system outside the VA network, and receives inbound data respond from DMIX DES system.

## BHIE Message Flow

The eMI BHIE Message Flow is implemented using one primary message flow and several subflows from the eMI Services Library.

The eMI BHIE Message Flow is a common message flow pattern that can proxy any RESTful (GET) API. It leverages the eMI Services Library to look up endpoints and consume the target service, and it leverages common error handling and logging mechanisms. It preserves the global, parent and local transaction ids, and query string, which enables the detailed level auditing of transactions. Additionally, it is the central place where all the errors are reformatted and sent back to the consumer based on the error code dictionary. The flow is described in detail below.

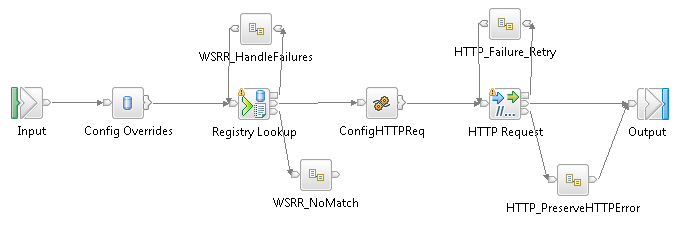
Figure - eMI BHIE Message Flow



1. HTTPS Input - Receives an HTTPS REST request for DMIX/DES data at URL https://<eMI host>:<port>/dmix/dataservice/v4.0/mhs/\* from a VA service consumer (e.g., Station 200, VistA Exchange), etc. Processing continues on the out terminal flow.
2. Set Environment – Extracts routing and other information from request and stores in eMI environment schema for subsequent processing. For example, this information includes:
   1. Global Transaction ID created by Layer 7
   2. HTTP request identifier
   3. HTTP method
   4. Query string parameters
   5. URL suffix (i.e., removes host and port, preserves path and path parameters)
3. DispatchHTTP – Delegates processing to the DispatchHTTP subflow, to lookup provider endpoint(s) in WSRR and relay the request to DMIX/DES.
4. ReplyWithSuccess – Delegates processing to the ReplyHTTP subflow, to relay the DMIX/DES provider response to the consumer.
5. RethrowHTTPTimeout – When replying to the consumer, if the connection with the consumer times out (i.e., the consumer fails to respond), this node throws an exception so that the error may be processed.
6. ReformatError – All exceptions occurring downstream will be propagated to the HTTP Input node’s catch terminal. The ReformatError node ensures that the current exception’s error information is captured properly for further processing and relay to the consumer.
7. ReplyWithError – Delegates processing to the ReplyHTTP subflow, to relay error information to the consumer.
8. RethrowToForceRollback – Rolls back the current transaction by throwing an eMI exception.
9. CompleteErrorHandling – Performs final error handling after the HTTP Input node’s catch terminal flow executes RethrowToForceRollback.

The Dispatch HTTP subflow is used to retrieve the target service endpoint URL from WSRR and consume the associated service. Should an error occur while connecting to WSRR or the target service, the subflow is designed to perform a retry based on user defined properties. Any error that occurs after successful connection to WSRR or the target service is re-thrown to the main flow, formatted to the canonical model, and sent to consumer. The flow is described in detail below.

Figure - Dispatch HTTP Subflow



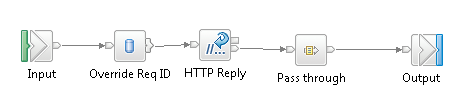
1. Input – Processing begins here when invoked by the calling message flow.
2. Config Overrides – Specifies configuration data for use in downstream processing.
3. Registry Lookup – Retrieves DMIX/DES provider endpoint from WSRR.
   1. WSRR\_HandleFailures – In the event WSRR cannot be reached, this node will extract the associated error message and initiate a retry.
   2. WSRR\_NoMatch – In the event a matching endpoint cannot be found, this node will extract the associated error message and throw an exception to be processed in the parent flow.
4. ConfigHTTPReq – Builds the final URL which the flow will use to forward the request to the DMIX/DES provider.
5. HTTP Request – Forwards the request to the DMIX/DES provider.
   1. HTTP\_Failure\_Retry - In the event the DMIX/DES provider cannot be reached, this node will extract the associated error message and initiate a retry.
   2. HTTP\_PreserveHTTPError – If an error occurs, this node will extract the error information for processing in the parent flow.
6. Output – Processing returns to the calling message flow.

The Reply HTTP subflow is used to:

* Relay successful responses to the consumer based on the response from the target service.
* Relay error messages to the consumer. Errors are formatted to a canonical model using common procedures.

The flow is described in detail below.

Figure - Reply HTTP Subflow



Input – Processing begins here when invoked by the calling message flow.

1. Override Req ID – Establishes the proper request identifier.
2. HTTP Reply – Forwards the response to the consumer.
3. Pass through – A placeholder node which allows for additional logging.
4. Output – Processing returns to the calling message flow.

## Receiver Flows

There is no receiver flow within eMI due to the DMIX/DES service provider implementation using synchronous REST web service endpoints. Each response from the DMIX/DES service provider is processed within the same HTTP request/response cycle as the associated request.

## Error Handling Flows

The DMIX/DES Proxy Service leverages common error handling mechanisms defined in the eMI Services Library. Details of the eMI Services Libraries are documented in the eMI SDD.

## Project Configuration File

The DMIX/DES Proxy Service will leverage a project configuration file for project configurable parameters. Table 5 lists the project configuration file details that are either environment specific or control the flow of messages.

Table 8 - Project Configurable Parameters

| Property | Default Value | Purpose |
| --- | --- | --- |
| Number of Retries | 3 | Number of retries to service provider before returning error to service consumer. |
| Retry delay | 1000 | Length of delay between retries. |

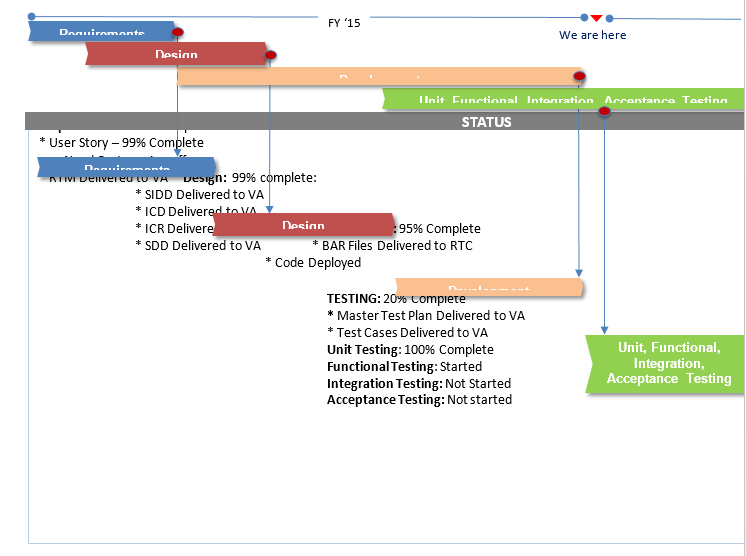
## Queue Details

N/A.

# Timeline

Figure 7 shows the timeline for the eMI BHIE Message Flow Implementation.

Figure - Implementation Timeline



# Acronyms

Table - Acronyms

| Abbreviation/Term | Definition |
| --- | --- |
| AITC | Austin Information Technology Center |
| DES | Data Exchange Service |
| DMIX | Defense Medical Information Exchange |
| DoD | Department of Defense |
| eMI | Enterprise Messaging Infrastructure |
| ESS | Enterprise Shared Service |
| IIB | IBM Integration Bus |
| LOINC | Logical Observation Identifiers Names and Codes |
| QoS | Quality of Service |
| SDD | System Design Document |
| SIDD | Service Integration Design Document |
| SLA | Service Level Agreement |
| SOA | Service Oriented Architecture |
| SSL | Secure Sockets Layer |
| VA | Veterans Affairs |
| VII | Veterans Interoperability and Integration |
| WSRR | WebSphere Service Registry Repository |

1. Architecture Design Decisions

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

1. Message Flow Documentation

The following PDF attachment contains the documentation generated by the IBM Integration Bus Toolkit for the DMIX/DES proxy message flow:



1. Message Mapping

The DoD DMIX/DES Provider External ICD v0.3 document attached here describes the v4 REST API which the eMI DMIX/DES Proxy Services support.

1. Outstanding Issues

The eMI BHIE Message Flow does not have any outstanding issue

1. Approval Signature

REVIEW DATE:

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

Integrated Project Team (IPT) Chair Date

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

Business Sponsor Date

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

IT Program Manager Date

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

Project Manager Date