DATS

Development Applications and Technical Services

Centers for Medicare & Medicaid Services

CMS eXpedited Life Cycle (XLC)

Electronic Submission of Medical Documentation (esMD)

Review Contractor (RC) Client Java User Guide and Installation Handbook

Version 8.1

01/21/2021

Document Number: RC Implementation Guide Java_AR2021.04.0

Contract Number: HHSM-500-75FCMC-20C0035

Document Change History

Table 1: Record of Changes

Version Number	Date	Author/Owner	Description of Change
8.1	1/21/2021	Steven Prest	1. Replaced broken image in Figure 19
8.0	1/15/2021	Karthik Srinivasan	2. Updated document to reflect changes from EIDM to IDM. 3. Updated section 1.4.3 4. Updated Table 82 – Request Level UTN: Changed tracking number from “1-50 Alphanumeric Characters” to “14 Alphanumeric Characters”. 5. Replaced all “EIDM” links with “IDM” links
7.2	10/13/2020	Vijayalakshmi Muthukrishnan	6. Updated Table 78
7.1	08/11/2020	Karthik Srinivasan	7. Updated document to reflect changes in comments from TOSS
7.0	08/07/2020	Karthik Srinivasan	1. Included November release functionality in section 1.4 2. Removed all references to Coversheet and FFR 3. Updated sections 12.1, 12.1.2, 12.1.4.1, 12.1.3, 12.1.5, 12.2, 14.2, 14.3, 19.1, 19.1.2.1, 4. Updated Figure 24 5. Updated Tables 11, Table 78* (Error codes sent from esMD to RC), Table 79*, Table 81*, 84*, 88*, 89* 6. Added Appendix G - FAQs

Version Number	Date	Author/Owner	Description of Change
6.1	04/20/2020	Karthik Srinivasan, Vijayalakshmi Muthukrishnan	1. Added a. Sections 12.1.26, 12.2.18, 12.2.19, 12.2.20, 14.5, 15 b. Figure 18 c. Table 10 2. Updated a. Section 1.4 – to include AR2020.07.0 release details b. Section 1.4.1 – to include Document Code File Request and pickup notification for RC client functionality c. Section 12.1 & 12.2 – added Document code file d. Section e. Table 9 – Included ICDT Directory f. Table 79 – Included Document Code Flat File Validation Error g. Table 80 – Included Error Document Codes Validate File h. Table 81 i. Table 87 – Included CTC 8.5 & 17 j. Table 88 – Included CTC 17
6.0	09/30/2019	Vijayalakshmi Muthukrishnan	• Updated for AR2020.01.0: a. Sections 1.4, 9.2, 11.1, and 11.2 b. Tables 10, 74, 77, and 83. • Added: a. Sections 11.1.24, 11.1.25, 11.2.15 through 11.2.16, 13.5, and 15.8 b. Figures 20 and 21; and c. Tables 33, 34, 47 through 49, 54, 55, and 71.
5.1	04/09/2019	Vijayalakshmi Muthukrishnan	Resolved review comments. Updated: 1. Section 1.4; and 2. Table 72.

Version Number	Date	Author/Owner	Description of Change
5.0	03/28/2019	Vijayalakshmi Muthukrishnan	1. Updated for AR2019.07.0: a. Sections 1.4, 1.4.1, 3 through 6, 8.2, 9, 11.1.1, 11.1.4, 11.1.7, 11.1.8, 13.8.1, 16.2.2, 17.2, and Appendix A; b. Figures 10 through 13; and c. Tables 12, 14,1 5, 17 through 35, 37 through 44, 55, 64, 66, 69, 70, and 72 through 75. 2. Added: a. Sections 9.1, 9.2, 10, 13.4, 15, and 18; b. Figures 7 through 9 and 19; and c. Tables 10, 11, 16, 36, 46, 54, and 67.
4.1	7/16/2018	Vijayalakshmi Muthukrishnan	Resolved CMS review comments. Updated sections 1.4.1 and 1.5
4.0	06/27/2018	Vijayalakshmi Muthukrishnan	Updated for AR2018.10.0: 1. Updated: a. Sections 1.4, 11.1, and 11.2; b. Tables 11, 84, and 93 through 97. 2. Added: a. 11.1.28, 11.1.29, 11.2.22, 11.2.23, 13.5, and 14.2.18; b. Figure 17; and c. Tables 39, 40, 61, 63, and 81.
3.2	4/6/2018	Vijayalakshmi Muthukrishnan	1. Updated section 15.1 2. Added sections 11.1.26 and 11.1.27
3.1	04/02/2018	Jim Runser	Resolved CMS review comments. Updated: 1. Section 1.4 paragraph 14; and 2. Table 88.
3.0	03/13/2018	Vijayalakshmi Muthukrishnan	Updates for AR2018.07.0: 1. Updated: a. Sections 1.4, 1.4.1, 11.1, 11.1.5, and 11.2; and b. Tables 10, 11, 78, 79, 85, and 88 through 90. 2. Added: a. Sections 11.1.25, 11.2.20, 11.2.21, 13.4, and 14.2.17; b. Tables 36, 55 through 57, and 74; and c. Figure 16.

Version Number	Date	Author/Owner	Description of Change
2.1	01/29/2018	Vijayalakshmi Muthukrishnan	Resolved CMS comments: 1. Updated Section 1.4 paragraph 13; 2. Moved: a. Former Sections 14.1 and 14.2 to Section 14.2.9 and 14.2.10, respectively, and b. Former Tables 49 and 54 to Tables 62 and 63, respectively. 3. Deleted Section 14.
2.0	01/10/2018	Vijayalakshmi Muthukrishnan	Original update for AR2018.04.0: 1. Updated: a. Sections 1.4, 1.4.1, 11.1, 11.1.3, 11.1.4, 11.2, and 11.2.6; b. Tables 1, 72, and 81 through 84; and c. Figures 7 through 10. 2. Added: a. Sections 11.1.1, 11.1.22, 11.1.23, 11.1.24, 11.2.18, 11.2.19, 13.3, and 15.2.14; b. Tables 33 through 35, 53, and 69; and c. Figure 15.
1.0	10/04/2017	Vijayalakshmi Muthukrishnan	No substantive changes for Release AR2018.01.0 added.

Table of Contents

List of Figures

List of Tables

1. Introduction

The Centers for Medicare & Medicaid Services (CMS) is a federal agency that ensures health care coverage for more than 100 million Americans. The CMS administers Medicare and provides funds and guidance for all the 50 states in the nation, for their Medicaid programs and Children's Health Insurance Program (CHIP). The CMS works together with the CMS community and organizations in delivering improved and better coordinated care.

1.1 Overview of the esMD System

Each year, the Medicare Fee-For-Service (FFS) Program makes billions of dollars in estimated improper payments. The CMS employs several types of Review Contractors (RC) to measure, prevent, identify, and correct these improper payments. RCs find improper payments and manually review claims against medical documentation obtained to verify the providers' compliance with Medicare rules. The RCs request medical documentation by sending a paper letter to the provider. In the past, medical documentation providers had only two options for delivering the medical documentation requested by sending it by letter or fax.

The Electronic Submission of Medical Documentation (esMD) system gives providers the option of sending medical documentation electronically to a requesting RC, instead of sending the documentation by letter or fax.

Many providers use a Health Information Handler (HIH) organization to perform tasks, such as submitting claims and providing electronic health record systems. Any organization that handles health information on behalf of a provider is an HIH. Some HIHs are beginning to offer esMD gateway services; Claim Clearinghouses, Release of Information vendors, Health Information Exchanges, and Electronic Health Record vendors are often referred to as HIHs.

The esMD system allows providers and HIHs to use gateway services to send responses for requests for additional documentation electronically to an RC during the claims review process.

1.1.1 The esMD Claim Review Contractors

Under the authority of the Social Security Act, CMS employs a variety of contractors to process and review claims in accordance with Medicare rules and regulations. Table 2: Medicare Contractors, Responsibilities, and Contact Information lists the review contractors referenced in this implementation guide.

Table 2: Medicare Contractors, Responsibilities, and Contact Information

Type of Contractor	Responsibilities	Contact Information
Medicare Administrative Contractors (MAC)	Process claims submitted by physicians, hospitals, and other health care professionals, and submit payment to those providers in accordance with Medicare rules and regulations. This includes identifying and correcting underpayments and overpayments.	http://www.cms.gov/ Research-Statistics-Data- and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/Review-Contractor-Directory- Interactive-Map
Zone Program Integrity Contractors (ZPIC), formerly Program Safeguard Contractors (PSC)	Identify cases of suspected fraud and take appropriate corrective actions.	http://www.cms.gov/ Research-Statistics-Data- and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/Review-Contractor-Directory- Interactive-Map
Supplemental Medical Review Contractor (SMRC)	Conduct nationwide medical review, as directed by CMS. This includes identifying underpayments and overpayments.	http://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/Medical-Review/SMRC.html
Contractor (CERT DC), CERT Review Contractor (CERT RC), and CERT Statistical Contractor (CERT SC)	Collect documentation and perform reviews on a statistically valid random sample of Medicare FFS claims to produce an annual improper payment rate.	https://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/CERT/index.html?redirect=/cert
Recovery Auditors	Identify underpayments and overpayments, as part of the Recovery Audit Program.	http://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/Recovery-Audit-Program/
Qualified Independent Contractor (QIC)	A party to the redetermination may request a reconsideration if dissatisfied with the redetermination decision. QIC conducts the reconsideration.	https://www.cms.gov/medicare/appeals-and-grievances/orgmedffsappeals/reconsiderationbyaqualifiedindependentcontractor.html

1.2 System Overview

The esMD system provides a mechanism for exchanging medical documentation and responses for Cross-Enterprise Document Reliable Interchange (XDR) and X12N 278

requests between the Medicare Provider community and the Medicare RC community. The purpose is to enable the electronic transmission of information between HIHs who represent Providers and the Medicare RCs, replacing paper documents where possible.

The RC Client is a utility that enables RCs to communicate with esMD by exchanging files via TIBCO® Managed File Transfer (MFT) server.

Note: The esMD system identifies submissions and requests sent from HIHs to RCs as inbound files and identifies transactions and responses for XDR and X12N 278 sent from RCs to HIHs as outbound files.

1.3 System Requirements

See Section 7, System Requirements for the system requirements for installing the Java version of the RC Client.

Section 7, System Requirements provides the requirements needed for the computer system where the RC Client will be installed, including the computer system's processor, amount of disk space and free memory needed, permissions, minimum internet connectivity Kilobits Per Second (Kbps) transfer speeds, and the Microsoft Java Framework version needed to run the RC Client properly.

Refer to the Identity Management (IDM) Instructions in the link below for details on how to obtain an IDM login:
http://www.cms.gov/Research-Statistics-Data-and-Systems/Computer-Data-and-Systems/ESMD/Downloads/IDMInstructions.docx

Refer to Section 1.4.3 RC Client Operation Overview for Enterprise File Transfer (EFT) Password requirements per IDM policy for logging in to the internal server.

https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/EnterpriseIdentityManagement/Guides-and-Documentation

1.4 RC Client Overview

In September 2011, CMS implemented the esMD program for providers to submit medical documentation in response to requests from Medicare RCs and also enhanced the esMD Gateway to support electronic responses to requests.

In January 2013, CMS expanded the CMS esMD Gateway to allow Durable Medical Equipment, Prosthetics/Orthotics & Services (DMEPOS) suppliers and providers to send electronic Prior Authorization (PA) Requests to Medicare RCs.

In June 2013, CMS enabled automated "Prior Authorization Review Results Responses" from Medicare RCs to HIHs via the esMD Gateway.

In June 2014, the “RC Client” application was implemented to allow data exchanges between HIHs and facilitate Medicare RCs electronically receiving PA requests to the RC’s computer system and allow RCs to electronically enter a decision on a PA request.

In June 2015, the “RC Client” application was extended to allow RCs to enter a Reject Error Code for a PA request electronically received, or electronically submit that there was an error in receiving the PA request’s response that was transmitted to the RC Client installed on the Medicare RC’s computer system or network. The RC is able to submit responses for PA programs, such as the Ambulance and Hyperbaric Oxygen (HBO).

In July 2016, the esMD application was updated to allow “RC Client” application to receive DMEPOS PA requests and Pre-Claim Review Demonstration for Home Health Services Pre-Claim Review (HHPCR) requests as X12N transactions and was able to send Review Result Responses for these programs. The “RC Client” application also began receiving Second Level Appeal Requests via the esMD system.

In October 2016, the “RC Client” application began receiving HHPCR Request as XDR transactions as well as additional information (ParentUniqueId and SplitNumber value in optional metadata element tags) in the RC Metadata Extensible Markup Language (XML) for matching/grouping the split payloads submitted by the HIH because of file size limitation in the esMD System.

In January 2017, the “RC Client” application started accepting structured medical documentation using the Health Level 7 (HL7) standard, Consolidated Clinical Document Architecture (C-CDA) from the HIHs. The structured medical documentation is sent to the “RC Client” application in XML format in the RC Package in addition to the existing Portable Document Format (PDF) format of the Payload files. The MFT folders for the RCs were moved from the Baltimore Data Center (BDC) to the Virtual Data Center (VDC).

In April 2017, the “RC Client” application programming interface was provided to the RCs to perform and support the Inter Contractor Document Transfer (ICDT) functionality. ICDT allows RCs to exchange files/documents from one RC to another RC, as needed, through the esMD system using Content Type Code 15.1 (ICDT Request), Content Type Code 15.2 (ICDT Solicited Response), and Content Type Code 15.3 (ICDT Unsolicited Response).

In July 2017, the “RC Client” application began receiving DME Phone Discussion Requests in XDR format and DMEPOS PA program in the XDR and X12 format. DMEPOS PA program in the XDR and X12 format was activated in the esMD system.

In October 2017 with AR2017.10.0, HIHs can send additional documentation for X12N 278 requests through X12N 275 transactions to esMD in addition to the previously existing XDR transactions. There was no impact to RCs because the existing Content Type Code 13 is referenced in the RC package for X12 requests.

esMD release AR2018.04.0 in April 2018 continued to support existing functionality and Lines of Business (LOB) while adding the following new capabilities. The RC Client application began sending the Additional Documentation Request (ADR) Review Results Letter (RRL) electronically through the esMD system using Content Type Code 1.3. Updates were made to the existing ADR Review Responses functionality. HIHs began sending Structured Documentation in the Clinical Documents for Payers (CDP) Set 1 format in addition to the C-CDA format. Also, HIHs began sending multiple services for XDR and X12 HHPCR requests.

esMD release AR2018.07.0 in July 2018 continued to support existing functionality and LOBs while adding the following new capabilities:

- The RC Client application will begin receiving Unsolicited Paperwork (PWK) claim documentation in XDR format using Content Type Code 7.

esMD release AR2018.10.0 in October 2018 added the following new capabilities:

- The RC Client application will begin sending PA/Pre-Claim Review (PCR) decision letters electronically through the esMD system using the Content Type Code 1.4; and
- Updates are made to the label of an Electronic Medical Documentation Request (eMDR) data element (Health Insurance Claim (HIC) Number (HICN)) to support the Medicare Beneficiary Identifier (MBI) transition.

esMD release AR2019.07.0 in July 2019 added the following new capabilities:

1. The RC Client User Interface (UI) screens and Application Programming Interface (API) are updated to support the new format of the esMD Transaction ID. Refer to section 16 Java Client API for more details.
2. The “Review Response to PA Request” tab is now disabled and all the PA responses for the decisions A, N and M has to be sent as Workload responses. Refer to section 11 Review Responses Through RC Client for more details
3. The RC Client application will begin receiving the Service Registration Requests batch file from esMD.

esMD release AR2020.01.0, in January 2020, supported existing functionality and LOBS while adding the following new capabilities:

1. The RC Client application will begin sending Pre-Pay eMDR ADR letters electronically through the esMD system using the Content Type Code 1.5; and
2. The RC Client application will begin sending Post-Pay eMDR ADR letters electronically through the esMD system using the Content Type Code 1.6;

esMD release AR2020.07.0, scheduled for July 2020, will continue to support existing functionality and LOBS while adding the following new capabilities:

1. The RC Client application will begin receiving Document Codes flat file from the esMD system using Content Type Code 17.

2. The RC Client application will begin receiving Hospital OutPatient Department XDR multiple services PA program using the Content type code 8.5

esMD release AR2020.11.0, scheduled for November 2020, will continue to support existing functionality and LOBS while adding the following capabilities:

1. The RC Client application began receiving X12 packages without FFR and CoverSheet from the esMD system using Content Type Code 13.

esMD release AR2021.04.0 in April 2021, does not have any changes to RC Client API. Updated the document for the EIDM to IDM migration.

1.4.1 RC Client Pull/Push Functionality

The RC Client provides the following functionality:

- Pull:
    - Inbound documents (submitted by HIHs) from the TIBCO MFT server;
    - HIH acknowledgements indicating receipt of pick up notifications, PA Review Result Responses, Administrative Error Response, PA PCR Decision Letters, and ADR Review Result Letters;
    - Data Element Validation results for the outbound process;
    - ICDT Request;
    - ICDT Solicited Response;
    - ICDT Unsolicited Response;
    - ICDT Batch Notifications (Acknowledgement/Pickup Notifications);
    - ICDT Validation Failures/Error Notifications;
    - ICDT Administrative Errors; and
    - eMDR Service Registration Request.
    - Document Code File Request.

- Push:
    - PA review decision responses to PA Requests for XDR and X12N 278 to esMD;
    - Error responses to PA Requests for XDR and X12N 278 to esMD;
    - Administrative Error response for XDR and X12N 278 to esMD;
    - Error messages generated due to file decompression and checksum verification;
    - Acknowledgement messages for receipt of documents and authorization requests;
    - Site-Specific Configuration settings:
        - Push frequency/Pull frequency; and

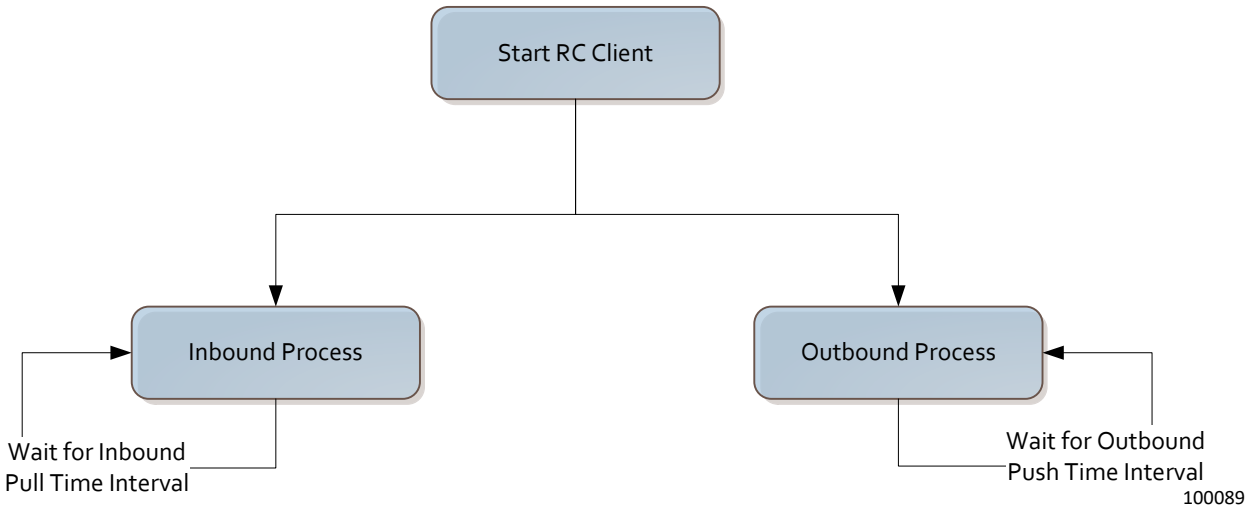
        - Folder locations for both Inbound and Outbound files.
    - ICDT to esMD:
        - ICDT Request;
        - ICDT Solicited Response; and
        - ICDT Unsolicited Response.
    - ICDT Pickup Notifications/Error Notifications to esMD;
    - ICDT Administrative Error Response to esMD;
    - ADR RRL to esMD;
    - eMDR Service Registration Request Pickup Notification;
    - Document Codes Request Pickup Notification;
    - PA PCR Decision Letters to esMD; and
    - PA/PCR Decision letters to esMD.

1.4.2 RC Client Application Overview

The esMD RC Java Client is a standalone Java Windows desktop application that runs outside the CMS network on the RC's machine, computer, or server. The purpose of the RC Java Client is to connect to the TIBCO MFT server at the VDC and push and pull files. The RC Java Client uses the IDM System login credentials to authenticate with the TIBCO MFT server. The RC Client users (at the RC site) provide their login credentials when they start the RC Client on their machines.

Users enter their login credentials only once at the program startup. When the RC Client starts, it initiates and then continuously runs two parallel threads as shown in Figure 1: RC Client Inbound and Outbound Process. When a user starts the RC Client, it will run continuously and will push and pull files automatically without continual user intervention, based on the frequencies set by the RC.

Figure 1: RC Client Inbound and Outbound Process



In the inbound process, when the RC Client connects to the TIBCO MFT server, the RC Client immediately executes a pull cycle. The documents are pulled into the RC's inbound user directory for the authenticated user, and then the RC Client disconnects and waits for the next cycle, as determined by the Inbound Pull Time Interval setting.

In the outbound process when the RC Client connects to the TIBCO MFT server, the RC Client executes a push cycle. The documents are pushed from the RC's outbound user directory to the TIBCO MFT server, and then the RC Client disconnects and waits for the next cycle as determined by the Outbound Push Time Interval setting.

The inbound pull frequency is independent of the outbound push frequency. After each successful push or pull process, the RC Client thread disconnects from the TIBCO MFT server. To ensure continuous operation of the RC Client, it must preserve each user's IDM login credentials during the program execution.

Note: Running multiple instances of the Java RC Client for the same jurisdiction could result in errors while pulling the files.

The RC Client was updated as part of esMD Release 4.0 to allow RCs to submit review responses for the new PA programs using the Graphical User Interface (GUI). The RC does not need to login to the TIBCO MFT Server in order to create Review Responses, Error Responses and Administrative Error Responses. The login is necessary only to pull or push files from or to TIBCO MFT Server.

1.4.3 RC Client Operation Overview

The RC Client runs in a cyclical manner sleeping for a specified time interval between the operating cycles. The sleep intervals are configured in the "checkFrequency" parameter for the Inbound Process and the "pushFrequency" parameter for the Outbound process. The RC is advised to use the default of 240 minutes (4 hours) for the Inbound process and 15 minutes for the Outbound process.

The RC Client operation is interrupted in two events:

1. IDM passwords that have expired (**Note:** IDM passwords expire every 60 days, if not changed); and
2. A Virus Scan error notification is received from the esMD.

In the first scenario, when the IDM password expires, the RC Client suspends its operation and is terminated. The RC must restart the RC Client and the user must provide the right credentials to login to the TIBCO MFT server. The IDM notifies the user 15 days prior to the password expiring. For more information on the IDM User Credentials and how to reset the password, please refer to the IDM Instructions document in the esMD Downloads section, using the link below:

http://www.cms.gov/Research-Statistics-Data-and-Systems/Computer-Data-and-Systems/ESMD/Information_for_Review-Contractors.html

The password setup in the portal must meet the following IDM for users to be able to log into Internet Server:

PASSWORD POLICY

1. Passwords must be at least 8 characters in length. [1]
2. Passwords must include an uppercase letter.
3. Passwords must include a lowercase letter
4. Passwords must include a number (0 - 9).
5. Passwords must include one of one special character.
6. Passwords must not contain a space.
7. Passwords must not be one of the user's last 24 passwords.
8. Passwords must not contain parts of the user's First Name, Last Name, or User ID.
9. 24 hours must have elapsed since the last password change.

Note: After the password reset, update the password to the new password in the configuration or script file if it is being stored and used by RC Client.

In the second scenario, when a Virus Scan error notification is received from esMD, all of the RC Client processes are suspended, and the RC Client is terminated. In addition, the RC Client is locked and cannot pull/push files even if the RC Client is restarted. The RC is advised to contact the esMD Service Desk (refer to Section 21. Contacts for more details) to unlock the RC Client.

1.5 ICDT Overview

ICDT functionality enables RCs to route ICDT Requests and ICDT Solicited/Unsolicited Responses to other RCs.

esMD supports the following types of ICDT Requests and ICDT Solicited/UnSolicited Responses as part of the initial pilot program:

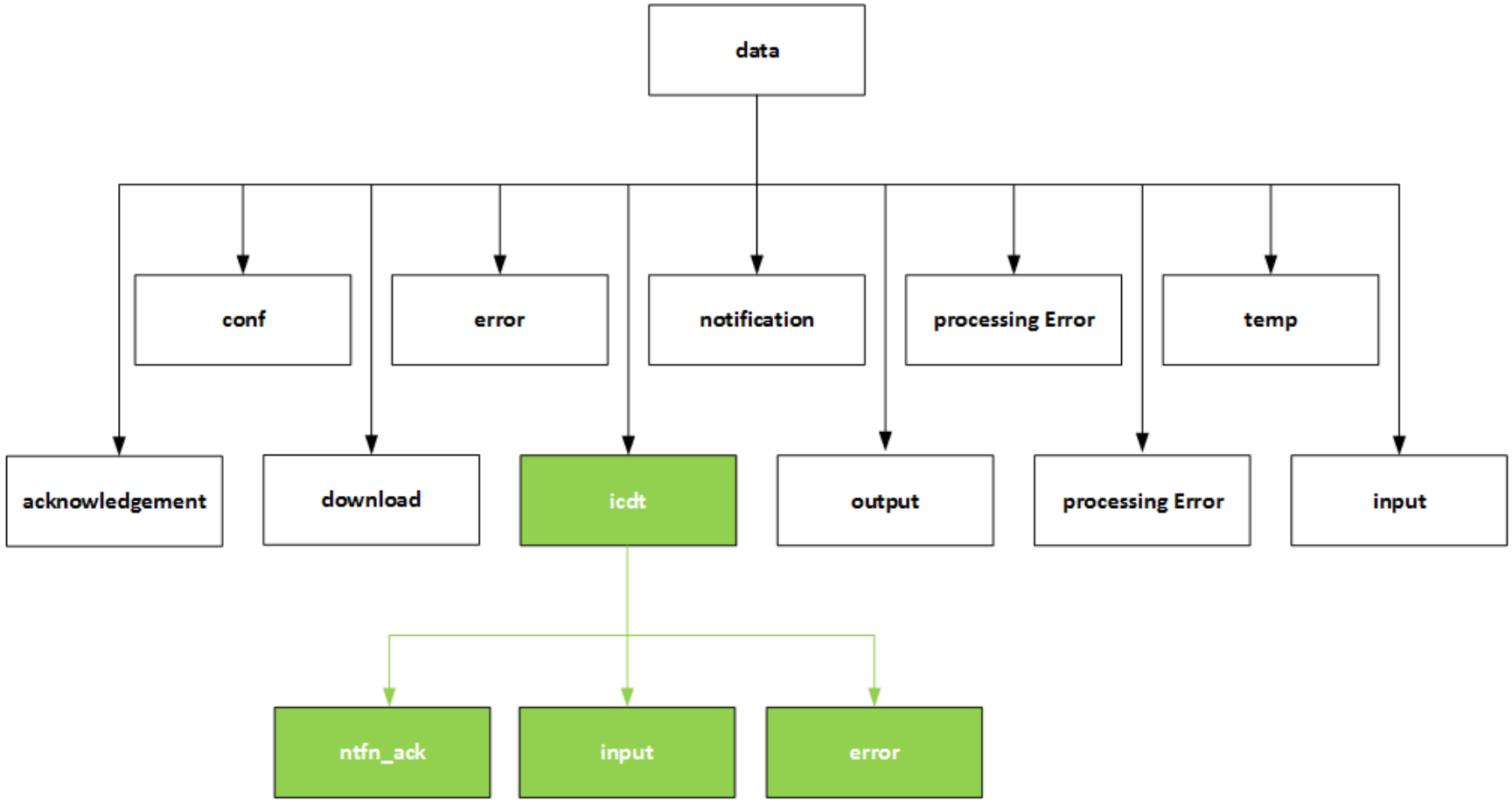
1. **ICDT Request/Solicited Response:** RC-A sends an ICDT Request to RC-B requesting certain documentation for a claim or a case, and RC-B responds (ICDT Solicited Response) to RC-A with the requested attachments; and
2. **ICDT Unsolicited Response:** RC-A sends the ICDT UnSolicited Response documentation bundle to RC-B (e.g., misdirected documentation).

[1] HARP user passwords must currently contain a minimum of 12 characters.

1.5.1 RC Client ICDT Folder Structure

A separate folder structure is used for placing the ICDT Request/Solicited Response, Unsolicited Response, Notifications, errors, and Acknowledgments files. The older ‘icdt’ is under the ‘data’ folder as shown Figure 2: RC Client ICDT Folder Structure. The ‘icdt’ folder contains three folders. The ICDT Request and ICDT Response files are moved to the ‘input’ folder. All of the notifications and acknowledgments are placed in the ‘ntfn_ack’ folder. Any validation errors and admin errors received from the esMD system are moved to the ‘error’ folder.

Figure 2: RC Client ICDT Folder Structure



2. Overview of How This Document is Structured

This document is structured into the following two primary sections.

1. First primary section of this document provides the following:

   - How to start and log into the RC Client;
   - How to enter a Review Response decision;
   - How to enter an error code for a PA request;
   - How to submit Inbound Submissions errors; and
   - Advanced debugging, which shows how to test to see if your RC Client application can connect to the TIBCO MFT server and if you have any inbound files ready for downloading.

   ❖ The audience for this first section is the **RC business users**.

2. How to install and configure a Java version of RC Client.

   ❖ The audience for this second section is the **person(s) installing the RC Client application**.

   This section provides the technical specifications for installing and configuring RC Client on a computer system or network and includes the following:

   - Overview of the installation process;
   - Systems Requirements for a Java installation;

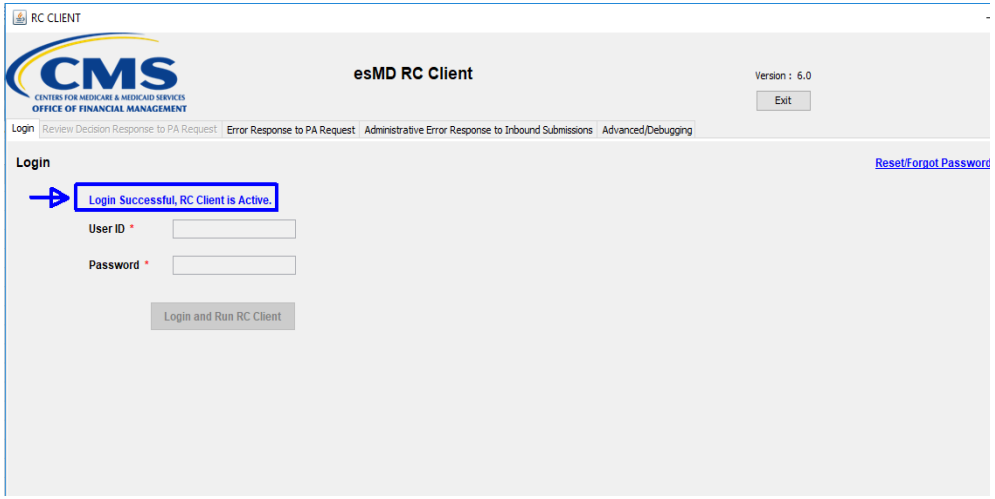
- Installing an Out-of-Box Java version of the RC Client application;
- TIBCO MFT file transfers;
- XML Messages, including Outbound, Inbound, and Error messages;
- Inbound Processes and Files;
- Outbound Processes and Files;
- Configuring the RC Client application;
- RC Client Components;
- RC Client Workflow;
- RC Client application Utilities, Components, Schedulers, and Encryption;
- Changes to the API;
- Using API;
- Configuring the RC Client application for notifications;
- Processing and pulling in documents; and
- Security.

3. How to Start the RC Client and Log In

The following are the step-by-step instructions for starting the RC Client and logging in.

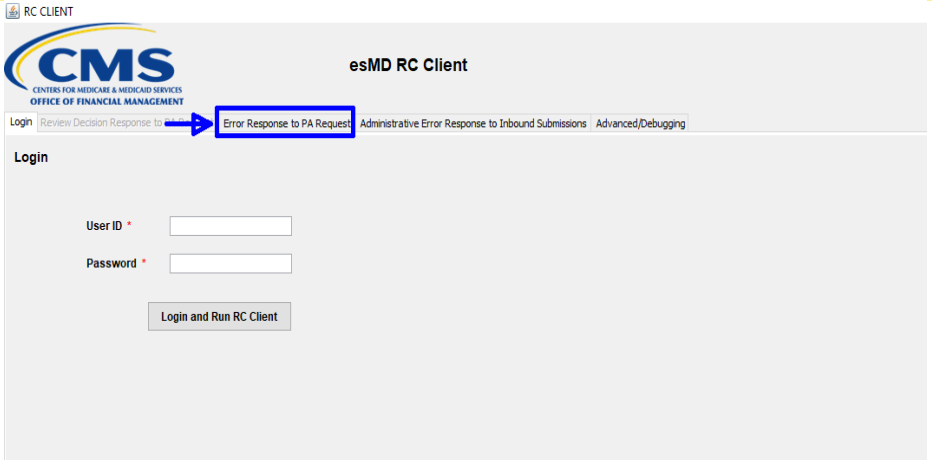
Step	Action
Step 1. Starting the RC Client and Logging In	Start the RC Client by selecting the **rcclient.bat** in the RC Installation folder or directory.

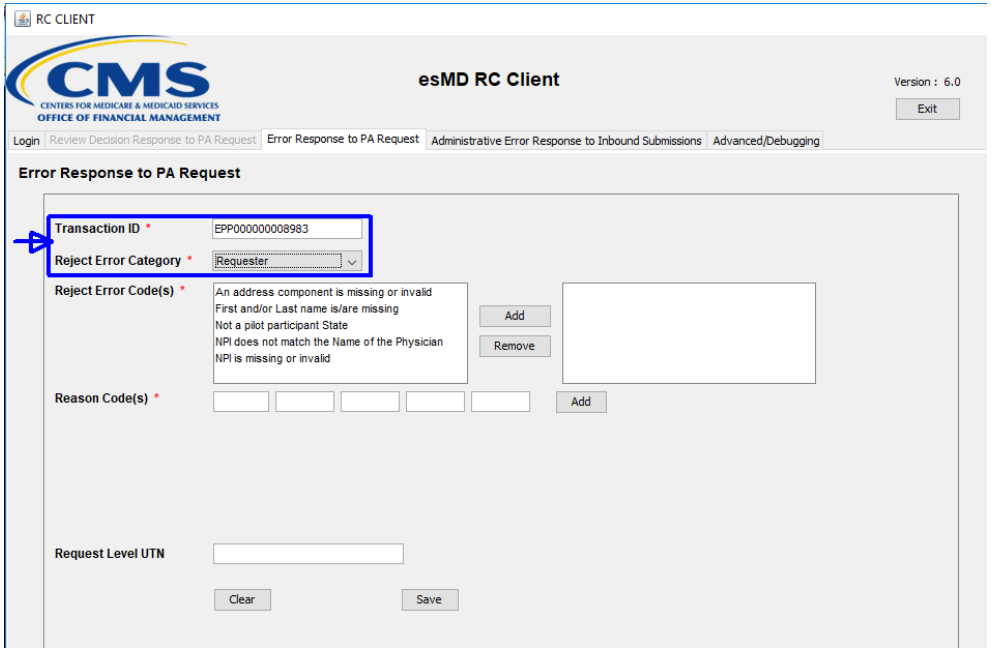
Step	Action
Step 2. Starting the RC Client and Logging In	The **Login** screen is displayed. Enter your IDM **User ID** and **password**, then select **Login and Run RC Client**. The IDM user name should use all uppercase letters. Example: XXXX **Note**: The IDM login credentials are confidential and should not be shared with others. (For more information on IDM login credentials, see IDM's User Guide in IDM confluence page (https://confluenceent.cms.gov/pages/viewpage.action?spaceKey=IDM&title=IDM+User+Guides) 

Step	Action
Step 3. Starting the RC Client and Logging In	After a successful log in, the **Login Successful, RC Client is Active.** message is displayed. 

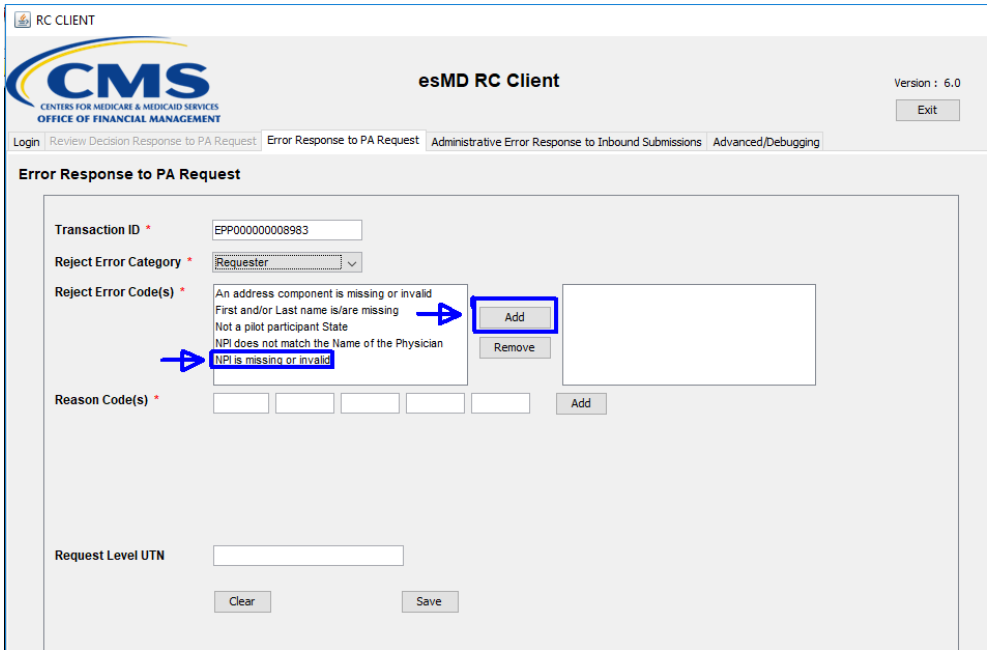
4. How to Enter an Error Code on the Error Response to PA Request Tab

This section provides step-by-step instructions on how to enter an error code on the **Error Response to PA Request** tab.

Step	Action
Step 1. Entering an **Error Code**	Select the **Error Response to PA Request** tab. ❖ After a successful log in, another log in is not required to navigate to and use the Error Response to PA Request tab. 

Step	Action
Step 2. Entering an **Error Code**	The fields for the **Error Response to PA Request** tab are displayed. ❖ Before You Begin: If you need a brief description of any of the fields on the tabs, see Appendix A: Description of Fields on RC Client Tabs. Enter the **Transaction ID** and select a **Reject Error Category**. 

Step	Action
Step 3. Entering an **Error Code**	Select a **Reject Error Code** and then **Add** to add the Reject Error Code. For information on how to access an up-to-date list of Reject Error Codes, see Appendix B: Reject Error Codes.



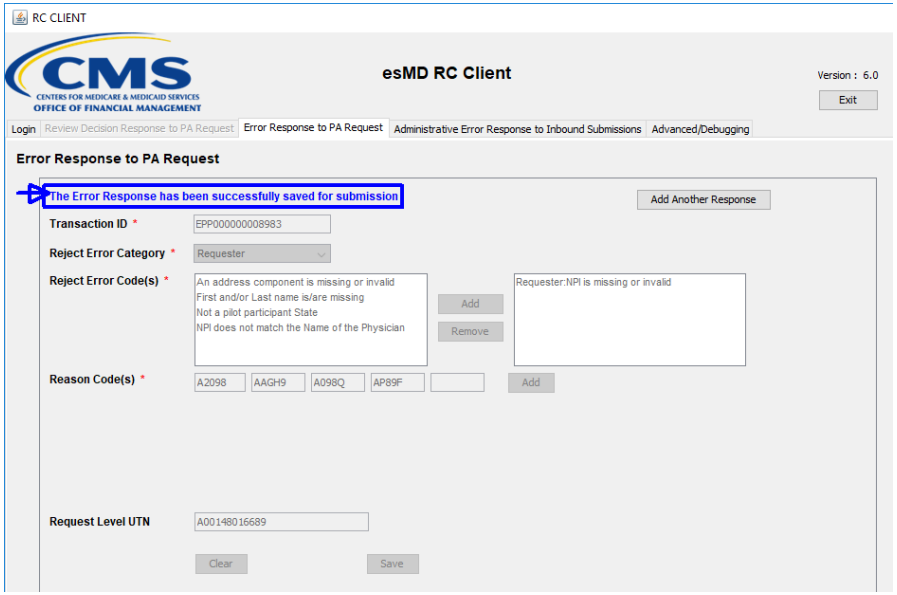
Step	Action
Step 4. Entering an **Error Code**	Enter the **Reason Code** or **Reason Codes**. Select Add at the end of the row of Reason Code fields to add additional rows of Reason Codes, as needed. Enter the **Request Level UTN** and then select **Save** to submit the Error Code for submission. Technical Note: After selecting Save, an XML message will be created to be sent to the esMD system and will be packaged into a compressed zip file. The zip file will be placed in a directory specified in the OutboundConfig/outputDirectory of the esmd-rc-client-config.xml. The outbound thread running on the RC Client will push the file to the TIBCO MFT server.

Step	Action
Step 5. Entering an **Error Code**	After selecting Save, the "**The Error Response has been successfully saved for submission**" message is displayed.

Technical Note: After selecting Save, the RC Client validates the data entered and displays errors messages, as applicable.

If the data validation is successful, the Error Code is created, and the "The Error Response has been successfully saved for submission" message is displayed.

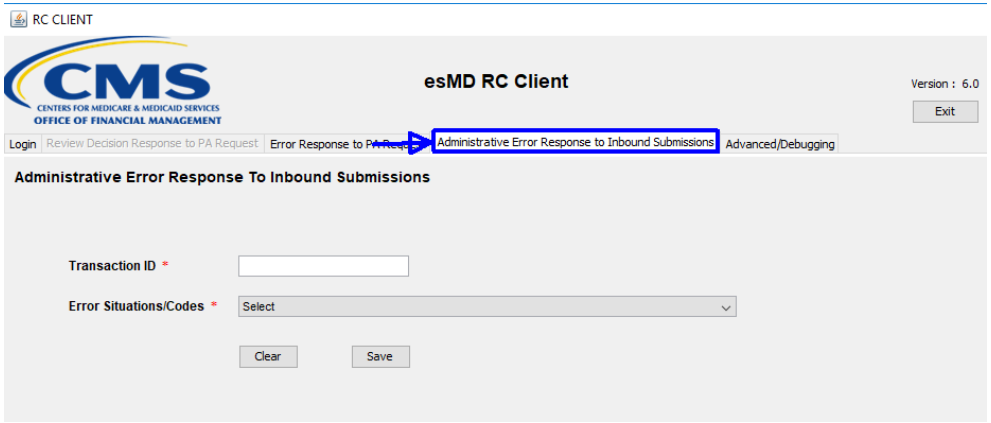
Note: After successfully saving a decision for submission, all information in the fields are cleared and another response may be entered.



5. How to Submit an Inbound Submission Error on the Administrative Error Response to Inbound Submissions Tab

This section provides step-by-step instructions on how to enter an inbound submission error on the **Administrative Error Response to Inbound Submissions** tab.

Step	Action
Step 1. Entering an **Inbound Submissions Error**	Select the **Administrative Error Response to Inbound Submissions** tab. ❖ After a successful log in, another log in is not required to navigate to and use the Administrative Error Response to Inbound Submissions tab.



Step 2.

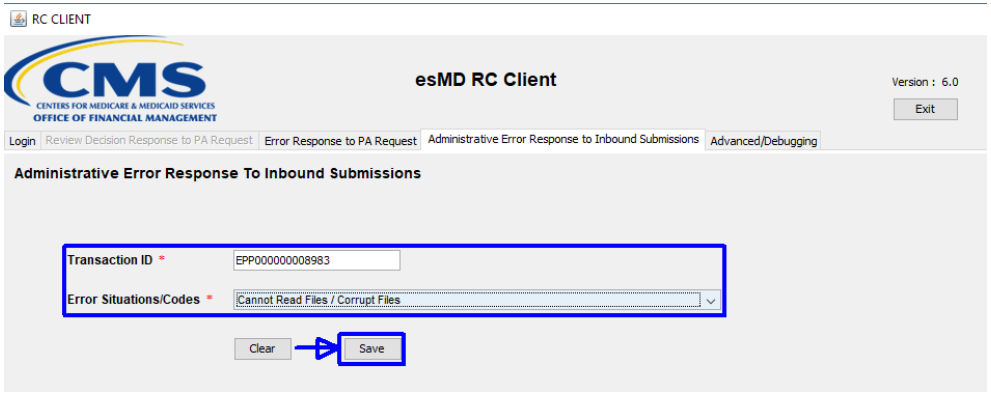
Entering an **Inbound Submissions Error**

The fields for the **Administrative Error Response to Inbound Submissions** tab are displayed.

❖ Before You Begin: If you need a brief description of any of the fields on the tabs, see Appendix A: Description of Fields on RC Client Tabs.

Enter the **Transaction ID**, select an **Error Situation** or **Error Code** from the Error Situations/Codes drop down menu, and then select **Save** to submit the Inbound Submissions error for submission.

Technical Note: After selecting Save, an XML message will be created to be sent to the esMD system and will be packaged into a compressed zip file. The zip file will be placed in a directory specified in the OutboundConfig/outputDirectory of the esmd-rc-client-config.xml. The outbound thread running on the RC Client will push the file to the TIBCO MFT server.

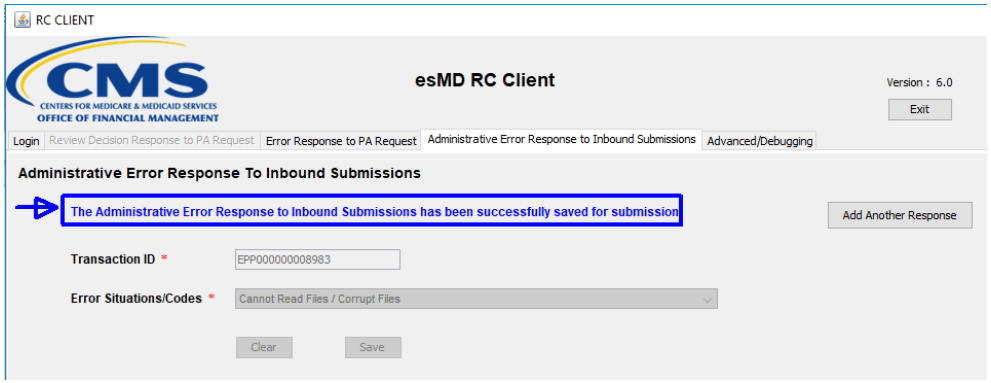


Step	Action
Step 3. Entering an **Inbound Submissions Error**	After selecting Save, the "**The Administrative Error Response to Inbound Submissions has been successfully saved for submission**" message is displayed.

Technical Note: After selecting Save, the RC Client validates the data entered and displays errors messages, as applicable.

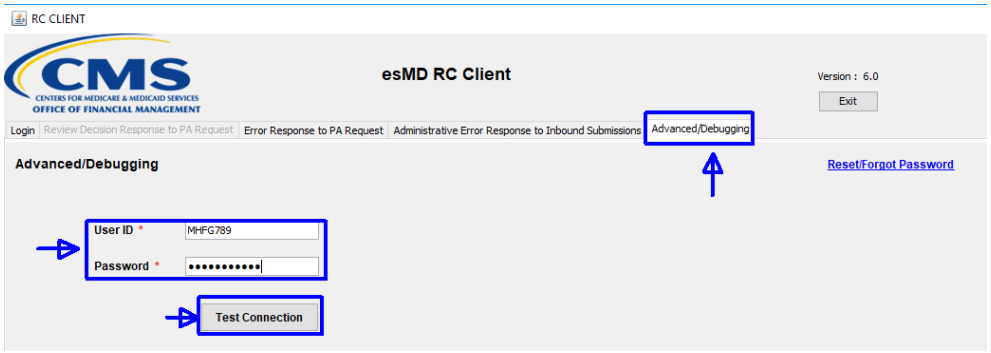
If the data validation is successful, the Inbound Submissions Error is created, and the "The Administrative Error Response to Inbound Submission has been successfully saved for submission" message is displayed.

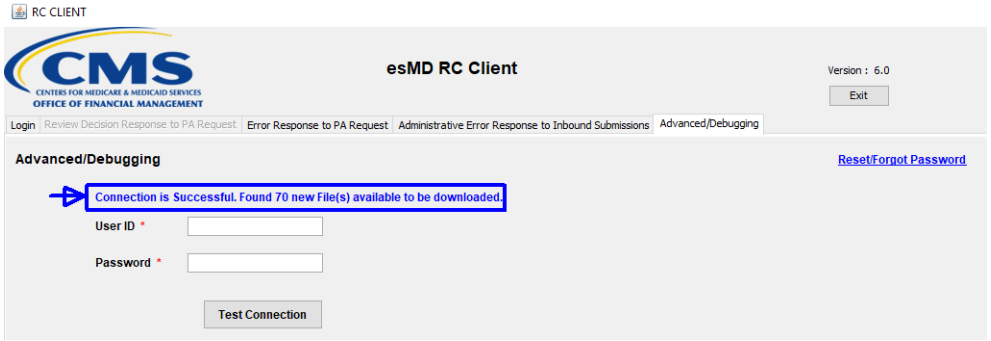
Note: After successfully saving a decision for submission, all information in the fields are cleared and another response may be entered.



6. How to Verify Connection to TIBCO MFT server, Using the Advanced/Debugging Tab

This section provides step-by-step instructions on how to verify connection to the TIBCO MFT server, using the **Advanced/Debugging** tab.

Step	Action
Step 1. Checking Connection to TIBCO MFT Server	Select the **Advanced/Debugging** tab. The Advanced/Debugging tab fields are displayed. On the Advanced/Debugging tab, enter your IDM **User ID** and **password**. (This is required on the Advanced/Debugging tab.) Select **Test Connection**. 

Step	Action
Step 2. Checking Connection to TIBCO MFT Server	After selecting Test Connection, the "**Connection is Successful. Found 3 new File(s) available to be downloaded.**" message is displayed. **Note**: After successfully testing your connection, you may select another tab. 

7. System Requirements

The following are the system requirements for installing a Java version of the RC Client.

7.1 Processor

The RC Client requires a Pentium 2 266-Megahertz (MHz) processor or greater.

7.2 Disk Space

The disk requirement for the RC Java Client is 10 Megabytes (MB). The documents that the RC Client pulls from the TIBCO MFT server may require additional disk space.

7.3 Memory

The RC Java Client requires a minimum of 50 MB of free memory.

7.4 Permissions

The RC Client must have read, write, and execute permissions on all the directories under the installation home.

7.5 Network

The RC Client requires internet connectivity that supports more than 32-Kbps transfer speeds.

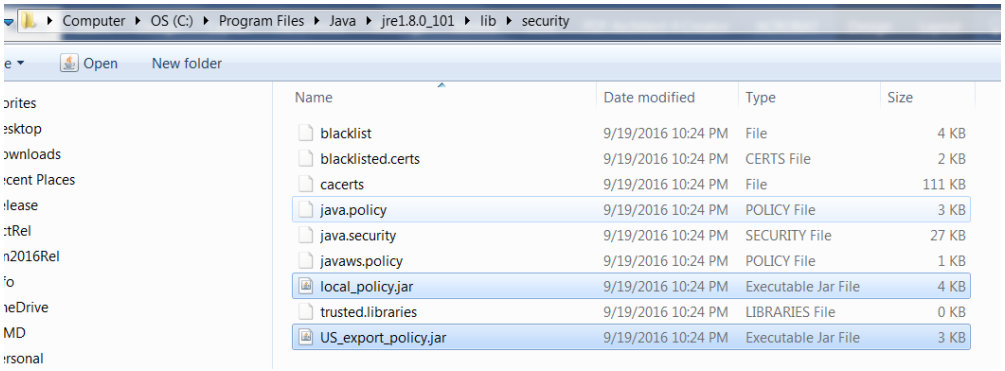
7.6 Java Framework

The RC Client requires Java Runtime Environment (JRE) 1.8 or greater. Section 7.6.1 describes how to download and configure the Java RC Client for upgrading the Java version to 1.8.

7.6.1 How to Configure the RC Client for Java 1.8 Version from an Earlier Version

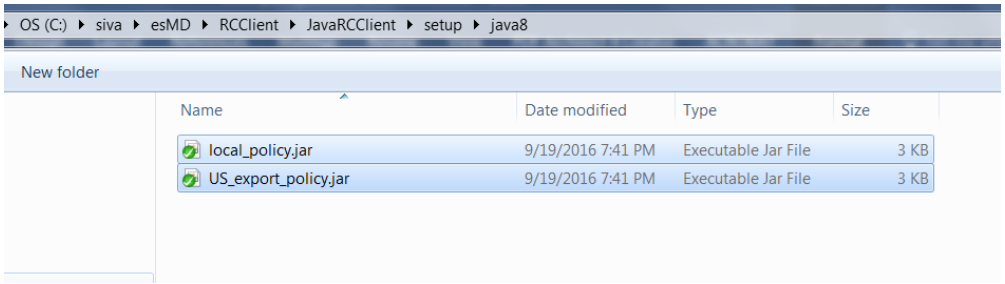
Step	Action
Step 1.	Download Java 1.8 (Java SE Development Kit 8u101) from the Oracle website, and install it in the machine where RC Client has been set up.
Step 2.	Backup the following Java Cryptography Encryption (JCE) Policy Files from the "...\Java\jre1.8.0_101\lib\security" path (refer to Figure 3: Directory Structure for the Policy Files): 1. local_policy.jar 2. US_export_policy.jar Delete these two files after backing them up.

Figure 3: Directory Structure for the Policy Files



Step	Action
Step 3.	Copy the JCE Policy Files (local_policy.jar, and US_export_policy.jar) provided in the RC Client installation package from "...\esMDRCClientAPISampleClient\setup\java8" as shown in Figure 4: Setup Directory for Java, and place it in the same path mentioned in step 2 ("...\Java\jre1.8.0_101\lib\security").

Figure 4: Setup Directory for Java



Step 4.	Run the RC Client and login.

7.7 Libraries

The Table 3: Libraries lists all the third-party libraries used by the RC Client along with their corresponding versions and a brief description of how the RC Client uses them.

Table 3: Libraries

Library	Version	Description
commons-codec	0.1.51	Used for Encoding and Decoding
commons-compress	1.2.17	Used for Extraction and compression of the packages.
commons-io	2.4	Used for reading and writing files to the Filesystem.
commons-lang	1.7	Used by the Java Secure Channel for Helper Utilities
commons-logging	1.7	Logging Framework used by the Jsch.
Jcalendar	1.0	Used for the popup Calendar in the GUI
Jsch	1.1	Java Secure Channel for Secure Shell (SSH) File Transfer Protocol (SFTP) Connection
log4j	4.0	Logging Framework

8. How to Install and Configure a Java Version of RC Client

Review the System Requirements in Section 7. System Requirements to make sure the machine that will host the RC Client meets the necessary requirements.

You can install the RC Client in two ways:

1. Out of the box; or
2. Custom RC Client (Java).

8.1 Out-of-the-Box

The RC Java Client API comes packaged with a sample client. To run this sample client out-of-the-box, the RCs need to follow the procedures in the following sections.

8.1.1 KeyStore Set Up

Important: The RC Client uses asymmetric encryption to store the IDM user credentials securely. For this encryption to work, you will need a secure Java KeyStore (JKS) with Public and Private keys of 2048 length. If you already have a JKS, you only need to update the configuration file with this information. Please refer to Section 16.1 Security for more details on the Security framework used by the RC Client.

1. If you do not have a JKS, create one for the RC Client to use. (Required.)
2. Type the following command to create a new keystore for the RC Client.

keytool -genkey -keyalg RSA -keystore <keystore> -alias <*alias*> -storepass <*storepassword*> -keypass <*keypassword*> -dname "CN=<commonName>, OU=<organizationalUnit>, O=<*organizationName*>, L=<localityName>, S=<*stateName*>, C=<*country*>" -keysize 2048 -validity 360

Note: Replace <parameter> with the value of the parameter from the list in Table 4: Keystore Creation Parameters.

This command creates the Public and Private keys, using the Rivest, Shamir & Adleman (RSA) Algorithm with a key size of 2048 and validity of one year.

Important: After the Public and Private keys have expired, you must re-create both keys to continue to use the RC Client.

Table 4: Keystore Creation Parameters

Where	Means
<keystore>	The keystore is the home location. If you do not specify the <keystore> option, the default keystore file named *keystore.jks* in the user's home directory will be created, if it does not already exist. For example, the config/keystore.jks will be created.
<alias>	The certificate chain and the private key are stored in a new keystore entry, identified by *alias.*
<storepassword>	The store password is used to protect the integrity of the keystore. It must be at least six characters long.
<keypassword>	The key password is used to protect the private key of the generated key pair. If a password is not provided by the user, the user is prompted to provide it. If you press Enter at the prompt for the key password, the key password is set to the same password that was used for the keystore. The <keypassword> must be at least six characters long.
<commonName>	The common name is the name for any entity, such as the name of a person (for example, Susan Jones) or the name of your company.
<organizationalUnit>	The organizational unit can be used for a small organization, department, or division of an organization (for example, Purchasing).
<organizationName>	The organization name is for a large organization or company (for example, ABC Systems, Inc.).
<localityName>	The locality name can be for a city (for example, Palo Alto).
<stateName>	The state name can be for a U.S. state or province of another country (for example, California or Ontario in Canada).
<country>	The country is a two-letter code (for example, US).

8.1.2 Integrity Verification

The following command will print the public key from the keystore and verify the keystore integrity.

1. Type the following command:

   keytool -list -v -keystore <*keystore*> -storepass <*storepassword*> -alias <*alias*>

Note: Replace <parameter> with the value for the parameter listed in Table 4: Keystore Creation Parameters.

8.1.3 Java Cryptography Extension (JCE) Policy Update

In addition to creating and providing the keystore, you may need to override the JCE security policy files if these files were not already overridden.

8.1.3.1 Understanding the JCE Security Policy Files

Due to import control restrictions, the version of the JCE security policy files bundled in the Java Development Kit™ (JDK) environment allows "strong" but limited cryptography to be used. To run the RC Client, this security policy must be overridden with the "unlimited strength" policy files that contain no restrictions on cryptographic strengths. If the RC Client is run with the default JCE security policy files, it will cause an error similar to the following:

```
java.security.InvalidKeyException: Illegal key size at
javax.crypto.Cipher.a(DashoA13*...)
```

New JCE security policy files are packaged along with the RC Client and are in the “setup” subdirectory of the installation directory.

Note: These files do not contain additional encryption functionality because such functionality is supported in Oracle’s JDK.

8.1.3.2 Understanding the Export/Import Issues

JCE for JDK has been through the U.S. export review process. The JCE framework, along with the Oracle JCE provider that comes standard with it, is exportable. The JCE architecture allows flexible cryptographic strength to be configured via jurisdiction policy files. Due to the import restrictions of some countries, the jurisdiction policy files distributed with the JDK software have built-in restrictions on available cryptographic strength.

8.1.3.3 JCE Policy Files

The setup directory in the RC Client installation contains the policy files listed in Table 5: JCE Policy Files.

Table 5: JCE Policy Files

Policy File	Description
local_policy.jar	Unlimited strength local policy file
US_export_policy.jar	Unlimited strength U.S. export policy file

8.1.3.4 Installation Locations for Windows and UNIX

<java-home> refers to the directory where the JRE was installed. It is determined based on whether you are running JCE on a JRE with the JDK installed. The JDK contains the JRE, but at a different level in the file hierarchy. Table 6: Java Development Kit and Table 7: Java Runtime Environment show examples of the installation for Java version 1.8.

Table 6: Java Development Kit

Environment	Example JDK Installation Directory	JAVA_HOME
Windows	C:\jdk1.8.0	C:\jdk1.8.0\jre
Unix	/home/user1/jdk1.8.0	/home/user1/jdk1.8.0/jre

Table 7: Java Runtime Environment

Environment	Example JRE Installation Directory	JAVA_HOME
Windows	C:\jre1.8.0	C:\jre1.8.0
Unix	/home/user1/jre1.8.0	/home/user1/jre1.8.0

Notes:

1. UNIX (Solaris/Linux) and Windows use different path name separators; use the appropriate one ("\" or "/") for your environment; and

2. On Windows, for each JDK installation, there may be an additional JRE installed under the "Program Files" directory. Ensure you install the unlimited strength policy Java Archive (JAR) files for all JREs that you plan to use.

8.1.3.5 Setting Up Encryption/Decryption without Limitation

To use the encryption/decryption functionalities of the JCE framework without any limitation:

1. Make a copy of the original JCE policy files (US_export_policy.jar and local_policy.jar in the standard place for JCE jurisdiction policy JAR files) in case you later decide to revert to these “strong” versions; and

2. Copy the policy files with the unlimited strength versions from the “setup” directory per the version of Java to be used under the installation directory to the security directory shown in Table 8: Security Directory.

Table 8: Security Directory

Environment	Installation Directory
Windows	<java-home>/lib/security
Unix	<java-home>\lib\security

8.1.4 Configuring the RC Client

Once the keystore is created and the policy files are installed, the RC Client is ready to be configured to use the keystore.

1. Update the keystore information in the configuration file (required); and

   Important: The XML configuration file (i.e., config/esmd-rc-client-config.xml) is used by the RC Client to retrieve important configuration parameters necessary for its operation.

2. Use the comments for each configuration parameter shown in Table 9: Sample RC Client Configuration File as a guide in entering your data.

Table 9: Sample RC Client Configuration File

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<ns2:ESMDConfig xmlns:ns2="http://esmd.ois.cms.hhs.gov/v1/rc/config"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://esmd.ois.cms.hhs.gov/v1/rc/config esmd-config.xsd ">

   <!--The TIBCO MFT Server Configuration-->
   <ESMDSFTPServer>
      <!--TIBCO MFT Sever host name or IP -->
      <host>eft1.feps.cms.gov</host>
      <!--The TIBCO MFT SFTP PORT-->
      <port>9022</port>
      <!-- Update: Use T for VAL, P for PROD-->
      <environmentId>T</environmentId>
      <!--The EFT File Name Prefix-->
      <eftFilePrefix>#EFT</eftFilePrefix>
   </ESMDSFTPServer>
   <!--The Keystore information for Encryption and Security-->
   <KeyStoreInfo>
      <!-- Update: The JKS Keystore Path-->
      <keyStoreLocation>/RCClient/config/keystore.jks</keyStoreLocation>
      <!-- Update: The Encrypted Keystore Password-->
      <encKeyInfo>ltwdafsdviaZNpvV54aRM9ZzQiw==</encKeyInfo>
      <!-- Update: The Encrypted Private Key Password-->
      <encKeyInfoExt>srs8adsfasRtLEB2I=</encKeyInfoExt>
      <!-- Update: The Certificate Alias-->
      <certAlias>selfsigned</certAlias>
   </KeyStoreInfo>
   <!--The Inbound Process Configuration-->
   <InboundConfig>
      <!-- Update: Enable the Inbound Process? true/false-->
      <enabled>true</enabled>
      <!--The Pull Frequency for the Inbound Process in minutes; the default is 240 minutes i.e., 4 hours--
>
      <checkFrequency>30</checkFrequency>
      <!-- Update: The RC Client installation/home directory-->
      <rcHomeDirectory>/RCClient</rcHomeDirectory>
```

```
        <!-- Update: The target directory to extract the downloaded inbound files before routing-->
        <targetDirectory>/RCClient/data/download</targetDirectory>
        <!-- Update: The input directory where the inbound payloads and the metadata will be routed after
the extraction-->
        <inputDirectory>/RCClient/data/input</inputDirectory>
        <!-- Update: The temp directory where the files are pulled from TIBCO MFT-->
        <tempDirectory>/RCClient/data/temp</tempDirectory>
        <!-- Update: The Error directory for routing the inbound error notifications from esMD/HIH-->
        <errorDirectory>/RCClient/data/error</errorDirectory>
        <!-- Update: The configuration directory for RC Client-->
        <configDirectory>/RCClient/data/conf</configDirectory>
        <!-- Update: The acknowledgments directory for routing the inbound notifications from esMD/HIH-->
        <acknowledgmentsDirectory>/RCClient/data/acknowledgment</acknowledgmentsDirectory>
        <!-- Update: The notifications directory for routing the inbound notifications from esMD/HIH-->
        <notificationsDirectory>/RCClient/data/notification</notificationsDirectory>
        <!-- Update: The Processing Error directory for routing only the unprocessed notifications from
esMD/HIH-->
        <processingErrorDirectory>/RCClient/data/processingError</processingErrorDirectory>
        <!-- Update: The Remote Inbound Directory path on the TIBCO MFT Server. IMPORTANT: Replace
ES#### with your own mail box number-->
        <remoteInboundDir>/ES####</remoteInboundDir>
        <!--Update: The mail box number for the inbound files used to pick the inbound files to pull-->
        <inboundRoutingId>ES####</inboundRoutingId>
    <ICDTDirectoryStructure>
            <inputDirectory>/data/icdt/input</inputDirectory>
            <errorDirectory>/data/icdt/error</errorDirectory>
            <notificationsDirectory>/data/icdt/ntfn_ack</notificationsDirectory>
    </ICDTDirectoryStructure>
   <eMDRRegistrationDirectory>/data/eMDRRegistration</eMDRRegistrationDirectory>
     </InboundConfig>
     <!--The Outbound Process Configuration-->
     <OutboundConfig>
        <!-- Update: Enable the Outbound Process? true/false-->
        <enabled>true</enabled>
        <!--The push frequency for the Outbound process in minutes default is 15 minutes-->
        <pushFrequency>15</pushFrequency>
        <!-- Update: The temp directory to use for the outbound process for creating the PMPDA/Notification
files-->
        <tempDirectory>/RCClient/data/temp</tempDirectory>
```

```
        <!-- Update: The local outbound directory to push the outbound files from-->
        <outputDirectory>/RCClient/data/output</outputDirectory>
        <!-- Update: The Remote Outbound directory to push files. IMPORTANT: Replace ES#### with your
own mail box number-->
        <remoteOutboundDir>/ES####_UPLOAD</remoteOutboundDir>
        <!--The Remote Outbound mail box number to push files onto esMD servers via TIBCO MFT.
LEAVE IT AS IT-->
        <outboundRoutingId>ESMD2</outboundRoutingId>
        <!--The Outbound File name prefix-->
        <outboundFilePrefix>ON</outboundFilePrefix>
    </OutboundConfig>
</ns2:ESMDConfig>
```

8.1.4.1 Configuring Your Password Encryption

1. Run the encryptConfig.bat script to update the KeystoreInfo section with the encrypted keystore and private key password;
2. When the script prompts, enter your keystore and private key passwords, as shown in Figure 5: Keystore Password Encryption and Figure 6: Private Key Password Encryption, and click OK in each Input window; and

Figure 5: Keystore Password Encryption

Figure 6: Private Key Password Encryption

3. Update the XML configuration file parameter "certAlias" with the alias of the certificate you created in Section 8.1.1 KeyStore Set Up.

   The KeystoreInfo section of the XML Configuration file is now updated with the encrypted passwords and the certificate information required for the RC Client operation.

8.1.5 Running the RC Client

Before you, as the RC, run the sample RC Client, you must double-check all the configuration parameters in the XML configuration file, especially the ones with the "Update" prefix in the comments of the sample XML configuration file, as shown in Table 9: Sample RC Client Configuration File.

1. To run the sample RC Client, run the "rcclient.bat" utility provided in the distribution package.
2. Start the RC Client by providing login credentials for the Login tab and select the “Login and Run RC Client” button.

8.2 Custom RC Client

The RC Java Client provides an API, so the RC can extend the RC Client to fit the RC’s environmental needs. The API enables the RC to perform the following functions:

1. Log in to the TIBCO MFT server (See Section 1.4.3 RC Client Operation Overview);
2. Get Notifications from the TIBCO MFT server using the SFTP. (Refer to Section 16.2.2 Inbound);
3. Decrypt/encrypt and store the login credentials using a secure RSA algorithm. (Refer to Section 16.2.6 Utilities - Encryption);
4. Pull medical documentation from the TIBCO MFT server. (Refer to Section 16.2.2 Inbound);
5. Extract the downloaded packages. (Refer to Section 16.2.2 Inbound);
6. Check the payloads using checksums in the metadata. (Refer to Section 16.2.2 Inbound); and
7. Push the outbound files from the “output” directory. (Refer to Section 16.2.3 Outbound).

Note: The procedures for customizing the RC Client API are beyond the scope of this document. (The source code that will be packaged along with the RC Client contains the documentation needed for integrating the API.)

9. TIBCO® MFT File Transfers

Table 10: Inbound and Outbound Files Format lists the zip/XML files that will be transferred between esMD and the RCs

Notes:

1. ES0001 is a sample mailbox number that the TIBCO MFT server uses to identify the RC, and “8NF” is a sample three-character alphanumeric value; and
2. The esMD transaction ID will not be included in the zip file name and will be included in the RC metadata XML file.

Table 10: Inbound and Outbound Files Format

Type	Example File Name	Description
Inbound	T.<<ReceiverRoutingId>>.L<<CTC>>.E<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	Submissions received from esMD to the RC: • T – Environment ID. T is for User Acceptance Testing (UAT) and P is for Production (PROD); • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – Content Type Code (CTC) of the program; • E – Delivery type of the inbound request; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS

Type	Example File Name	Description
Inbound	T.<<ReceiverRoutingId>>.L<<CTC>>.A<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	Acknowledgments received from esMD to the RC: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program; • A – Delivery type of the Acknowledgments; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.
Inbound	T.<<ReceiverRoutingId>>.L<<CTC>>.N<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	HIH delivery notification from esMD to RC: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program; • N – Delivery type of the HIH Notifications; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.

Type	Example File Name	Description
Inbound	T.<<ReceiverRoutingId>>.L<<CTC>>.F<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	Any validation failures from esMD: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program; • F – Delivery type of the esMD validation failures; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.
Inbound	T.<<ReceiverRoutingId>>.MISC.Y<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	Virus scan failure errors from esMD to RC: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • MISC– Miscellaneous Type; • Y – Delivery type of the virus scan failures; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.
Inbound	T.<<ReceiverRoutingId>>.MISC.X<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	Virus found or infected file sent by RC to esMD: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • MISC– Miscellaneous Type; • X – Delivery type of the virus-scan-infected errors; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.

Type	Example File Name	Description
Inbound/ Inbound	T.<<ReceiverRoutingId>>.L<<CTC>>.Q<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	ICDT Solicited Request from RC to esMD to RC: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program; • Q – Delivery type of the ICDT Solicited request; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.
Inbound/ Outbound	T.<<ReceiverRoutingId>>.L<<CTC>>.R<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	ICDT Solicited and Unsolicited Response from RC to esMD to RC. • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program; • R – Delivery type of the ICDT Solicited and Unsolicited Response; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.
Inbound/ Outbound	T.<<ReceiverRoutingId>>.ADM.C<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	ICDT Administrative error Response from RC to esMD to RC: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • ADM – Administrative error response; • C – Delivery type of the Administrative error response; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.

Type	Example File Name	Description
Inbound/ Outbound	T.<<ReceiverRoutingId>>.ICDT.B<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	ICDT Pickup Notification from RC to esMD to RC and ICDT acknowledgments from esMD to RC: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • ICDT – Type of response; • B – Delivery type of the ICDT Pickup Notification; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.
Inbound	T.<<ReceiverRoutingId>>.L<<CTC>>.V<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	esMD validation errors from esMD to RC for ICDT Request or Solicited or Unsolicited Response. • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program; • V – Delivery type of the ICDT Solicited and Unsolicited Response; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in DMMDDYY; and • THHmmssS – Time format in THHmmssS.

Type	Example File Name	Description
Outbound	T.<<ReceiverRoutingId>>.L<<CTC>>.U<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	RRLs, decision letters, and Pre-Pay and Post-Pay eMDR ADR letters from RC to esMD: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program; • U – Delivery type; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.
Inbound	T. <<ReceiverRoutingId>>.L<<CTC>>.E<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	Service Registration flat file from esMD to RC: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program which is “5”; • E – Delivery type of the Service Registration Request; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.

Type	Example File Name	Description
Outbound	T#EFT.ON.<<ReceiverRoutingId>>.L<<CTC>>.P<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	Pickup notification file name from RC to esMD for the Service Registration: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program which is “5”; • P – Delivery type of the Pickup Notification for Service Registration Request; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and • THHmmssS – Time format in THHmmssS.
Inbound	T.<<ReceiverRoutingId>>.L<<CTC>>.E<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	Document Code flat file from esMD to RC: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program which is “17”; • E – Delivery type of the DCF Request; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and THHmmssS – Time format in THHmmssS.

Type	Example File Name	Description
Outbound	T#EFT.ON. <<ReceiverRoutingId>>.L<<CTC>>.P<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	Pickup notification file name from RC to esMD for the document Code file: • T – Environment ID. T is for UAT and P is for PROD; • <<ReceiverRoutingId>> – RC Routing ID; • L – The Line of Business; • <<CTC>> – CTC of the program which is “17”; • P – Delivery type of the Pickup Notification for Document Code file; • <<3CharRandom>> – 3-character random number included by esMD; • <<SenderRoutingID>> – Sender Routing ID which is ESMD2; • DMMddyy – Date format in MMDDYY; and THHmmssS – Time format in THHmmssS.

10. XML Schema Definitions

The following schema definitions are updated for the new format of the esMD Transaction ID. They are available at the following Universal Resource Locator (URL): https://www.cms.gov/Research-Statistics-Data-and-Systems/Computer-Data-and-Systems/ESMD/Information_for_Review-Contractors.html.

1. esMD-businesstypes.xsd;
2. esMD-config.xsd;
3. esMD-rc.xsd;
4. esMDProcessMetadata.xsd;
5. emdr-rcprocessmetadata.xsd; and
6. emdr-postpay.xsd

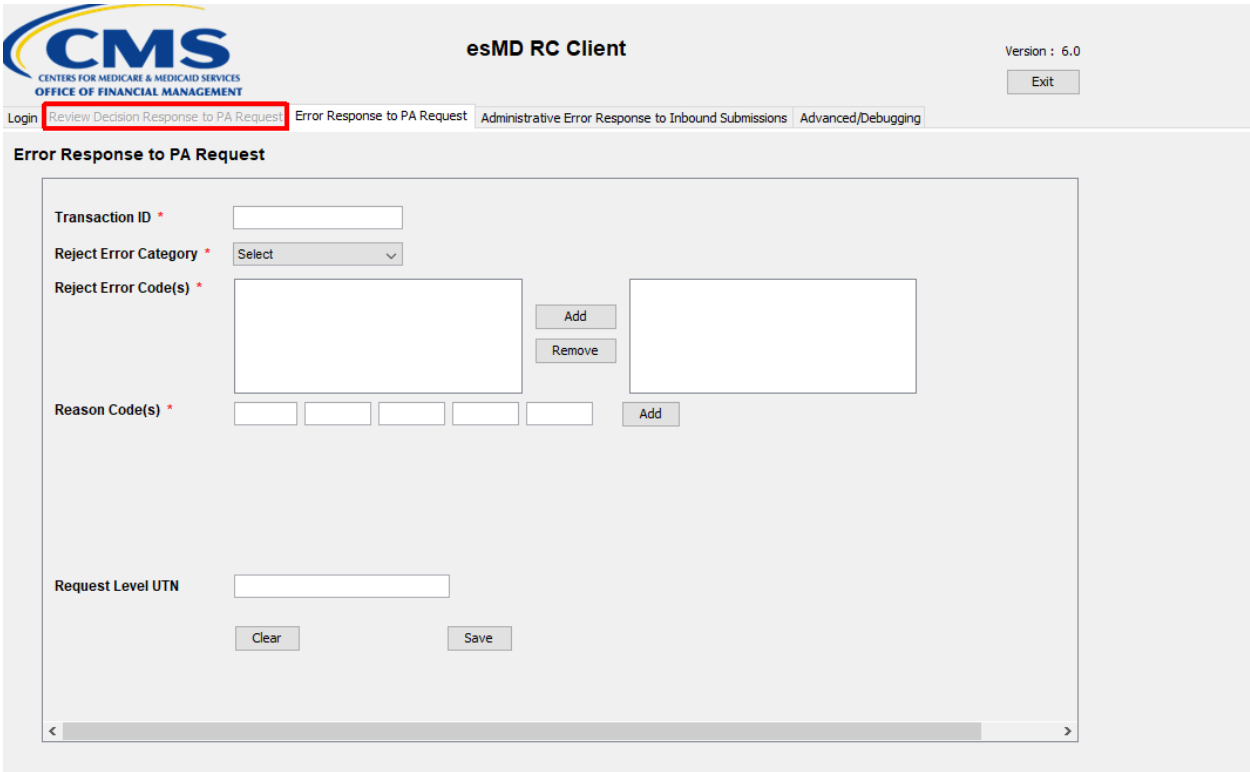
11. Review Responses Through RC Client

11.1 Review Decision Response to PA Request

As part of the July 2019 release, the Review Decision Response to PA Request tab, as shown in Figure 7: Review Response To PA Request, was disabled to restrict RCs from sending PA Responses for decision indicators A, N, and M. The esMD system rejects any PA responses for decision indicators A, N, and M from RCs if the decision is sent via the RC Client. The RCs continue to send decision indicators A, N, and M to esMD via the Workload Response files.

Note: RCs can continue to send Error Responses to PA Requests through the RC Client.

Figure 7: Review Response To PA Request



11.2 Multiple Review Responses

Starting in the July 2019 release, RCs have the ability to send multiple review responses to HIHs for the same transaction. esMD accepts and processes multiple review responses for the same transaction.

12. XML Messages

This section describes the various XML messages transferred during the inbound and outbound processes.

12.1 Inbound

Note: Please refer to the Appendix A: Description of Fields on RC Client Tabs for details on how RC Client routes the inbound files once they are successfully processed into the data directories.

The RC Client transfers the following files during the inbound process:

1. Payload Files in PDF and XML formats;
2. Metadata File;
3. Pickup HIH Status Response;
4. Pickup Validation Error Response;
5. Administrative Error HIH Status Response;
6. Administrative Error Response Validation Error;
7. Virus Scan Error Response;
8. PA Review Result HIH Status Response;
9. PA Review Result Validation Error Response;
10. esMD Acknowledgement Response for the ADR Response/eMDR Request;
11. esMD Validation Error Response for the ADR Response/eMDR Request;
12. HIH Delivery Notification Response for the ADR Response/eMDR Request;
13. ICDT Request XML;
14. ICDT Solicited Response XML;
15. ICDT Unsolicited Response XML;
16. ICDT Pickup Notification/Acknowledgement Response (as a batch process);
17. ICDT Pickup Error Notification;
18. ICDT Validation Error Notification;
19. ICDT Acknowledgement Notification;
20. ICDT Admin Error Response;
21. esMD Acknowledgment Response for ADR RRL;
22. esMD Validation Error Response for ADR RRL;
23. HIH Delivery Notification Response for ADR RRL;
24. Service Registration Request;
25. HIH Delivery Notification for Service Registration Response;
26. esMD Validation Error Response for Service Registration Response;
27. esMD Acknowledgment Response for PA/PCR Decision letters;

28. HIH Delivery Notification Response for PA/PCR Decision letters;
29. esMD validation Error Response for Pre-Pay eMDR letters;
30. esMD validation Error Response for Post-Pay eMDR letters; and
31. Document Code File.

12.1.1 Payload Files in PDF and XML Formats

The RC Client will receive PDF files as payloads in the inbound documents with delivery type “E”. Examples of payload file names are E_JIT000000008418-AA569852P2545215519840365951551984040625_0.pdf or E_JIT000000008418-AA569852P2545215519840365951551984040625_0.xml

12.1.2 Metadata File

The metadata file accompanies the payload files as part of inbound documents sent to RC Client. These documents name will always start delivery type “E”, followed by content type code and global unique ID. The metadata file contains information about the payloads like the Object Identifier (OID), Transaction ID, Submission metadata (includes Attachment Control Number and other information), and optional metadata. The Content Type Code will change for each line of business. See Table 11: E_L13_TNZ000007047921_metadata.xml.

Note: The metadata file will remain the same for all lines of business. For all list of line of business, please refer Table 86: Content Type Codes and Business Types.

Note: The Claim ID is optional for First Level Appeal Requests and Second Level Appeal Requests.

Note: HIHs send new Claim ID updates for the acceptance of 8 numeric characters or the current ClaimId validations.

For more information on the Content Type Codes, refer to Appendix D: Content Type Codes.

Table 11: E_L13_TNZ000007047921_metadata.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<ns0:RetrieveMedicalDocumentationResponse xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc"
returnCode="1" serviceSuccessful="true">

   <statusDescription>The RetrieveMedicalDocumentationRequest processed
successfully.</statusDescription>

   <NumberOfDocuments>1</NumberOfDocuments>

   <ESMDPackage>
```

```
<ESMDTransaction TransactionId="TNZ000007047921" DeliveryType="E"/>
<SendingOID>urn:oid:123.456.657.126</SendingOID>
<TargetOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</TargetOID>
<CompleteSubmission>true</CompleteSubmission>
<RequestType>X12-XDR</RequestType>
<SubmissionMetadata>
  <BusinessType>XDR X12</BusinessType>
  <CreationTime>2020-07-21T11:17:08.894-04:00</CreationTime>
  <SubmissionTime>2020-07-21T11:17:08.894-04:00</SubmissionTime>
  <EFTSubmissionTime>2020-07-21T11:17:08.894-04:00</EFTSubmissionTime>
  <ContentTypeCode>13</ContentTypeCode>
  <NPI>1111111112</NPI>
</SubmissionMetadata>
<Documentation DocumentUniqueIdentifier="E_TNZ000007047921-
SK783159P6721415953446261791595344628735_0" MimeType="application/pdf"
FileName="E_TNZ000007047921-SK783159P6721415953446261791595344628735_0.pdf">
  <OptionalMetadata>
    <FieldName>FileName</FieldName>
    <FieldValue>E_TNZ000007047921-
SK783159P6721415953446261791595344628735_0.pdf</FieldValue>
  </OptionalMetadata>
  <OptionalMetadata>
    <FieldName>Description</FieldName>
    <FieldValue>From esMD</FieldValue>
  </OptionalMetadata>
  <OptionalMetadata>
    <FieldName>CheckSum</FieldName>
    <FieldValue>572d8369be300056270a41f070abb63a903fb440</FieldValue>
```

```
        </OptionalMetadata>
        <OptionalMetadata>
          <FieldName>AttachmentControlNumber</FieldName>
          <FieldValue>SK783159P1230909045</FieldValue>
        </OptionalMetadata>
      </Documentation>
    </ESMDPackage>
</ns0:RetrieveMedicalDocumentationResponse>
```

12.1.2.1 Split Payload Transactions

There is an optional functionality provided for HIHs to split the payloads when sending files are larger than 200 MB in size. Payloads that are larger than 200 MB in size are sent in multiple transactions by HIHs. In case of HIHs splitting the payloads when the sending files are larger than 200 MB in size, RCs will match/group the payloads using the additional information (ParentUniqueId and SplitNumber value set in the OptionalMetadata tag) in the RC metadata XML file. The same ParentUniqueId and a different SplitNumber (e.g., 1-5) value are passed in the RC Metadata XML file for all the transactions that are intended for a single submission by the HIH. RCs might receive duplicate split numbers or additional split numbers or missing split numbers for the same ParentUniqueID when HIHs are sending them.

Note: This is an optional functionality for HIHs to use when submitting payloads larger than 200 MB in size. Not all transactions received from HIHs have this additional information for RCs.

12.1.3 Pickup HIH Status Response

When the RC Client sends a pickup notification to esMD, the esMD application processes the notification and sends the response to the HIH. Once the esMD application receives the acknowledgement for the pickup notification from HIH, then it generates the Pickup Status Response and sends it to the RC, indicating the response was sent to the HIH, as detailed in the code in Table 12: N_L8_1_KBW000000006908_Delivery_Acknowledgement.xml.

Note: The metadata file will remain the same for all lines of business. For all list of line of business, please refer Table 86: Content Type Codes and Business Types.

Table 12: N_L8_1_KBW000000006908_Delivery_Acknowledgement.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ns0:RCPickupNotificationResponse xmlns:ns0="http://esmd.ois.cms.hhs.gov/v1/rc/config">
   <ESMDTransactionId>KBW000000006908</ESMDTransactionId>
   <ErrorInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
   <Status>Success</Status>
   <StatusDesc>SENT PICKUP STATUS TO HIH</StatusDesc>
</ns0:RCPickupNotificationResponse>
```

12.1.4 Pickup Validation Error Response

When the RC Client sends a Pickup Notification to esMD, the esMD application processes and sends the Pickup Notification to the HIH. If there is an error in processing the Pickup Notification submitted by the RC, the esMD application generates the Pickup Validation Error Response, as detailed in Table 13: F_L13_PDW000000007903_Pickup_Validation_Error.xml, and sends it to the RC. The RC will correct the pickup notification and resubmit it to esMD.

Table 13: F_L13_PDW000000007903_Pickup_Validation_Error.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:RCPickupNotificationResponse xmlns:ns0="http://esmd.ois.cms.hhs.gov/v1/rc/config">
   <ESMDTransactionId>PDW000000007903</ESMDTransactionId>
   <ErrorInfo>
      <ErrorCode>520</ErrorCode>
      <ErrorName> </ErrorName>
      <ErrorDescription>Review Contractor Response Transaction ID does not
exist</ErrorDescription>
   </ErrorInfo>
   <Status>FAILED</Status>
   <StatusDesc>esMD validation error. Please correct and resubmit.</StatusDesc>
   <fileName>D.ESMD2.L13.PQNM.ESD002.D030619.T1234460</fileName>
</ns0:RCPickupNotificationResponse>
```

12.1.5 Administrative Error HIH Status Response

When the RC Client sends an administrative error for an inbound submission to esMD, the esMD application processes the administrative error and sends the response to the HIH. Once the esMD application receives the acknowledgement for the administrative error from HIH, then it generates the Administrative Error HIH Status Response and sends it to the RC, indicating the error was sent to the HIH, as detailed in the code in Table 14: N_L1_IUC000000006217_Delivery_Acknowledgement.xml.

Note: The Administrative Error HIH Status Response will remain the same for all lines of business. For all list of line of business, please refer Table 86: Content Type Codes and Business Types.

Table 14: N_L1_IUC000000006217_Delivery_Acknowledgement.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ns0:SubmitPADeterminationResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc"
returnCode="1" serviceSuccessful="true">
   <statusDescription>Sent administrative error response delivery to HIH</statusDescription>
   <ESMDTransaction TransactionId="IUC000000006217" DeliveryType="N"/>
</ns0:SubmitPADeterminationResponseResult>
```

12.1.6 Administrative Error Response Validation Error

When the RC Client sends an Administrative Error Response to esMD, the esMD application processes and sends the Administrative Error Response to the HIH. If there is an error in processing the Administrative Error Response submitted by the RC, the esMD application generates the Administrative Error Response Validation Error, as detailed in Table 15: F_ADMIN_123456788912345_Validation_Error.xml, and sends it to the RC. The RC will correct the administrative error response and resubmits it.

Table 15: F_ADMIN_123456788912345_Validation_Error.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:SubmitPADeterminationResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc"
returnCode="1" serviceSuccessful="true">
   <statusDescription>esMD validation error. Please correct and resubmit.</statusDescription>
   <ESMDTransaction TransactionId="123456788912345" DeliveryType="F"
ParentTransactionId="123456788912345" RoutingId="ESD002"/>
   <ReceivedFileName>D.ESMD2.ADM.DWMX.ESD002.D011619.T1337020</ReceivedFileName>
   <ValidationFailure>
      <FailureCode>633</FailureCode>
      <FailureReason>esMD validation error: Either HIH is not active or agreement has expired to
receive the response.</FailureReason>
   </ValidationFailure>
   <ValidationFailure>
      <FailureCode>613</FailureCode>
      <FailureReason>esMD validation error : Administrative error code is invalid. Correct and
resubmit</FailureReason>
   </ValidationFailure>
</ns0:SubmitPADeterminationResponseResult>
```

12.1.7 esMD Virus Scanning Service Down Error Response

When the RC Client sends any outbound file to esMD, the esMD application sends it to the esMD Virus Scanning Service for virus scanning. If the service is down and esMD fails to perform virus scanning for any outbound files from RCs, then an error response is sent back to the RC.

The esMD application sends the error response message detailed in Table 16: Y_L151Q5DNES9996D030819T1405310_Virus_Scan_Gateway_Failure.xml to the RC if the esMD Virus Scanning Service is unavailable or down while performing virus scanning of all responses sent from RCs.

Table 16:
Y_L151Q5DNES9996D030819T1405310_Virus_Scan_Gateway_Failure.xml

```
<ns0:RCPickupNotificationResponse xmlns:ns0="http://esmd.ois.cms.hhs.gov/v1/rc/config">
   <ESMDTransactionId> L151Q5DNES9996D030819T1405310</ESMDTransactionId>
   <ErrorInfo>
      <ErrorCode>609</ErrorCode>
      <ErrorName/>
      <ErrorDescription>esMD Virus Scanning service is unavailable</ErrorDescription>
   </ErrorInfo>
   <Status>FAILED</Status>
   <StatusDesc>esMD internal system error - esMD Virus Scanning service is unavailable.So the
response is rejected.</StatusDesc>
</ns0:RCPickupNotificationResponse>
```

12.1.8 Virus Scan Error Response

When the RC Client sends any outbound file to esMD, the esMD application sends it to the Virus Scan Gateway for virus scan. If the file is found to be virus infected, the esMD application sends the message detailed in Table 17: X_L151Q5DNES9996D030819T1405310_Virus_Scan_Error.xml to the RC. The RC Client will then pull this Virus Scan Error, stop the inbound and outbound processes, and lock down the RC Client to prevent RC Client from interacting with esMD. In this situation, the RC Client does not enable recovery, and the RC will contact esMD Service Desk.

Table 17: X_L151Q5DNES9996D030819T1405310_Virus_Scan_Error.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:RCPickupNotificationResponse xmlns:tns="http://esmd.ois.cms.hhs.gov/v1/rc/config"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://esmd.ois.cms.hhs.gov/v1/rc/config esmd-config.xsd">
  <ESMDTransactionId> L151Q5DNES9996D030819T1405310</ESMDTransactionId>
  <ErrorInfo>
    <ErrorCode>560</ErrorCode>
    <ErrorName>VirusFound</ErrorName>
    <ErrorDescription>ESMD validation error:  Submission is infected with virus</ErrorDescription>
  </ErrorInfo>
  <Status>FAILED</Status>
  <StatusDesc>Outbound Response File contains virus and so the response is
rejected.</StatusDesc>
</tns:RCPickupNotificationResponse>
```

12.1.9 PA Review Result HIH Status Response

When the RC Client sends a PA Review Result to esMD, the esMD application processes the file and sends the PA Review Result to the HIH. The esMD application submits the PA Review Result HIH Status Response, detailed in Table 18:

N_L8_1_ENS000000004289_Delivery_Acknowledgement.xml, and sends it to the RC, indicating the result was sent to the HIH.

Table 18: N_L8_1_ENS000000004289_Delivery_Acknowledgement.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:SubmitPADeterminationResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc"
returnCode="1" serviceSuccessful="true">
   <statusDescription>Sent PA Review result response delivery to HIH</statusDescription>
   <ESMDTransaction TransactionId="ENS000000004289" DeliveryType="N"/>
</ns0:SubmitPADeterminationResponseResult>
```

12.1.10 PA Review Result Validation Error Response

When the RC Client sends a PA Review Result to esMD, the esMD application processes and sends the PA Review Result to the HIH. If there is an error in processing the PA Review Result submitted by the RC, the esMD application generates the PA Results Response Error, as detailed in Table 19: F_PA_321312313112312_Review_Response_Validation_Error.xml, and sends it to the RC. The RC will correct the response and resubmits the PA Review Result.

Table 19: F_PA_321312313112312_Review_Response_Validation_Error.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<ns0:SubmitPADeterminationResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc"
returnCode="1" serviceSuccessful="true">
   <statusDescription>esMD validation error. Please correct and resubmit.</statusDescription>
   <ESMDTransaction TransactionId="321312313112312" DeliveryType="F"
ParentTransactionId="321312313112312" RoutingId="ESD002"/>
   <ReceivedFileName>D.ESMD2.PA.RUDN.ESD002.D011619.T1342310</ReceivedFileName>
   <ValidationFailure>
      <FailureCode>631</FailureCode>
      <FailureReason>esMD validation error: A Review or Error Response is not allowed for this
transaction.</FailureReason>
   </ValidationFailure>
</ns0:SubmitPADeterminationResponseResult>
```

12.1.11 ICDT Request XML

The RCs send the ICDT Request to another RC via esMD in XML format as part of the ICDT Request Package with delivery type “Q”. The file name of the ICDT Request should contain only alphanumeric characters and underscores (i.e., “_”).

Table 20: Q_L1518DMESD0020315191038490_ICDTSolicitedRequest.xml shows the XML message generated for an ICDT Request XML from RCs.

Table 20: Q_L1518DMESD0020315191038490_ICDTSolicitedRequest.xml

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<ICDTRequest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://cms.hhs.gov/esmd/icdt">
   <receiverOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.50</receiverOID>
   <receiverID>01232</receiverID>
   <senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</senderOID>
   <senderID>01231</senderID>
   <requestID>L1518DMESD0020315191038490</requestID>
   <contentType>15.1</contentType>
   <TransactionType transType="Claim">
      <OptionalMetadata>
         <FieldName>CLAIM_ID</FieldName>
         <FieldValue>12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>CASE_ID</FieldName>
         <FieldValue>12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>NPI</FieldName>
         <FieldValue>1234567890</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>HICN</FieldName>
         <FieldValue>1234567</FieldValue>
      </OptionalMetadata>
   </TransactionType>
</ICDTRequest>
```

12.1.12 ICDT Solicited Response XML

The RCs send the ICDT Response Document to another RC via esMD in any file format **except executable files** as part of the ICDT Response Package with delivery type “R”. The ICDT Solicited Response is sent for the Request from another RC. The file name of the ICDT Response Document should contain only alphanumeric characters and underscore (i.e., “_”).

Table 21: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml shows the sample XML Message of the ICDT Response sent from the RCs.

Table 21: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ICDTResponse xmlns="http://cms.hhs.gov/esmd/icdt">
   <receiverOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.4</receiverOID>
   <receiverID>01232</receiverID>
   <senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.3</senderOID>
   <senderID>01231</senderID>
   <requestID>QAC5K9XTESD0011210162133560</requestID>
   <contentType>15.2</contentType>
```

```
   <responseID>L1521R7ESD0020130191302560</responseID>
   <TransactionType transType="Claim">
      <OptionalMetadata>
         <FieldName>CLAIM_ID</FieldName>
         <FieldValue>Claim ID 12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>CASE_ID</FieldName>
         <FieldValue>CASE ID 12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>NPI</FieldName>
         <FieldValue>1234567890</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>HICN</FieldName>
         <FieldValue>HICN123456</FieldValue>
      </OptionalMetadata>
   </TransactionType>
   <Documentation FileName="pdf-sample_1.pdf" MimeType="application/pdf" DocUniqueID="pdf-
sample_1">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>fc80d59877b4ae21911591b53664b2da1324cf25</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>7945</FieldValue>
      </OptionalMetadata>
   </Documentation>
   <numberOfDocuments>1</numberOfDocuments>
</ICDTResponse>
```

12.1.13 ICDT Unsolicited Response XML

The RCs send the ICDT UnSolicited Response Document to another RC via esMD in any file format **except executable files** as part of the ICDT Response Package with delivery type “O”. The file name of the ICDT Response Document should contain only alphanumeric characters and underscore (i.e., “_”).

Table 22: R_L153PQQES99960308191418450_ICDTUnSolicitedResponse.xml shows the sample XML Message of the ICDT UnSolicited Response sent from the RCs.

Table 22: R_L153PQQES99960308191418450_ICDTUnSolicitedResponse.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ICDTResponse xmlns="http://cms.hhs.gov/esmd/icdt">
   <receiverOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.4</receiverOID>
   <receiverID>01232</receiverID>
   <senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.3</senderOID>
   <senderID>01231</senderID>
   <contentType>15.3</contentType>
   <responseID>L153PQQES99960308191418450</responseID>
```

```
   <TransactionType transType="SMRC-Misroute">
      <OptionalMetadata>
         <FieldName>CLAIM_ID</FieldName>
         <FieldValue>Claim ID 12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>CASE_ID</FieldName>
         <FieldValue>CASE ID 12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>NPI</FieldName>
         <FieldValue>1234567890</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>HICN</FieldName>
         <FieldValue>HICN12345</FieldValue>
      </OptionalMetadata>
   </TransactionType>
   <Documentation FileName="pdf-sample_1.pdf" MimeType="application/pdf" DocUniqueID="pdf-
sample_1">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>fc80d59877b4ae21911591b53664b2da1324cf25</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>7945</FieldValue>
      </OptionalMetadata>
   </Documentation>
   <Documentation FileName="pdf-sample_2.pdf" MimeType="application/pdf" DocUniqueID="pdf-
sample_2">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>fc80d59877b4ae21911591b53664b2da1324cf25</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>7945</FieldValue>
      </OptionalMetadata>
   </Documentation>
   <numberOfDocuments>2</numberOfDocuments>
</ICDTResponse>
```

12.1.14 ICDT Pickup Notification/Acknowledgement Response (as a Batch Process)

When the RC Client sends an ICDT Request or ICDT Solicited/Unsolicited Response to esMD, the esMD application validates the ICDT Request or ICDT Solicited/Unsolicited Response and generates the acknowledgement response as a batch to the RC. The Recipient RC downloads the package and sends the successful pickup notification to esMD. The pickup notification/acknowledgement response is generated in a batch file

in a single XML. Each RC receives the multiple batch acknowledgements in a day as per the Batch schedule of the esMD system.

The esMD system sends the Successful Pickup Notifications and Acknowledgments in a batch response XML, i.e., the notifications and acknowledgments from esMD are not delivered to the RC in real time. The Batch process at the esMD system is scheduled to run multiple times in a day. One batch file is generated for each RC in an esMD batch schedule with all of the pickup notifications and acknowledgments that were received during the particular time duration. The pickup notifications and acknowledgments are delivered to the RC in XML format and not in the zip file format.

The RequestType element in the XML indicates whether it is a notification or acknowledgment for the particular transaction.

The esMD system generates the batch acknowledgement response to the RC as shown in Table 23: T.ESD002.ICDT.BJ99.ESMD2.D031519.T1201000, and sends it to the RC.

Each request has an ICDTNotification block with all of the details pertaining to the request. The number of requests in the XML file is identified by the ID value (highlighted).

Table 23: T.ESD002.ICDT.BJ99.ESMD2.D031519.T1201000

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:ICDTCommunication xmlns:ns0="http://cms.hhs.gov/esmd/icdt">
   <ns0:ICDTNotification RequestType="SOLIC_REQ_ACK" id="1">
      <ns0:ICDTMetaData>
         <ns0:receiverOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.50</ns0:receiverOID>
         <ns0:receiverID>01232</ns0:receiverID>
         <ns0:senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</ns0:senderOID>
         <ns0:senderID>01231</ns0:senderID>
         <ns0:requestID>L1518DMESD0020315191038490</ns0:requestID>
         <ns0:contentType>15.1</ns0:contentType>
      </ns0:ICDTMetaData>
      <ns0:creationTime>2019-03-15T12:01:00.335-04:00</ns0:creationTime>
      <ns0:fileName>D#EFT.ON.ESMD2.L15_1.Q8DM.ESD002.D031519.T1038490</ns0:fileName>
      <ns0:Status>
         <ns0:serviceSuccessful>true</ns0:serviceSuccessful>
      </ns0:Status>
   </ns0:ICDTNotification>
   <ns0:ICDTNotification RequestType="SOLIC_REQ_ACK" id="2">
      <ns0:ICDTMetaData>
         <ns0:receiverOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.50</ns0:receiverOID>
         <ns0:receiverID>01232</ns0:receiverID>
         <ns0:senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</ns0:senderOID>
         <ns0:senderID>01231</ns0:senderID>
         <ns0:requestID>L151C2WESD0020315191038580</ns0:requestID>
         <ns0:contentType>15.1</ns0:contentType>
      </ns0:ICDTMetaData>
      <ns0:creationTime>2019-03-15T12:01:00.335-04:00</ns0:creationTime>
      <ns0:fileName>D#EFT.ON.ESMD2.L15_1.QC2W.ESD002.D031519.T1038580</ns0:fileName>
      <ns0:Status>
```

```
        <ns0:serviceSuccessful>true</ns0:serviceSuccessful>
      </ns0:Status>
   </ns0:ICDTNotification>
</ns0:ICDTCommunication>
```

12.1.15 ICDT Validation Error/Pickup Error Notification

When the RC Client sends an ICDT Request or ICDT Solicited/Unsolicited Response to esMD, the esMD application processes and sends the ICDT Request/Response to another RC. If there is an error processing the ICDT Request or ICDT Solicited/Unsolicited Response submitted by the RC at the esMD system, the esMD application generates the Validation Error Notification as detailed in Table 24: V_L153RLELESD002D031519T1040260_Validation_Error.xml, and sends it to the RC.

Table 24: V_L153RLELESD002D031519T1040260_Validation_Error.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:ICDTCommunication xmlns:ns0="http://cms.hhs.gov/esmd/icdt">
   <ns0:ICDTNotificationFailure RequestType="RESP_VALDTN_ERR" id="1">
      <ns0:ICDTMetaData>
         <ns0:receiverOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.50</ns0:receiverOID>
         <ns0:receiverID>01232</ns0:receiverID>
         <ns0:senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</ns0:senderOID>
         <ns0:senderID>01231</ns0:senderID>
         <ns0:requestID> </ns0:requestID>
         <ns0:contentType>15.3</ns0:contentType>
      </ns0:ICDTMetaData>
      <ns0:creationTime>2019-03-15T11:25:18.457-04:00</ns0:creationTime>
      <ns0:responseID>L153LELESD0020315191040260</ns0:responseID>
      <ns0:fileName>D.ESMD2.L15_3.RLEL.ESD002.D031519.T1040260</ns0:fileName>
      <ns0:Status>
         <ns0:description>esMD validation error. Please correct and resubmit.</ns0:description>
         <ns0:serviceSuccessful>false</ns0:serviceSuccessful>
      </ns0:Status>
      <ns0:ValidationFailure>
         <ns0:FailureCode>969</ns0:FailureCode>
         <ns0:FailureReason>esMD Validation Error: The documentation type received in the ICDT
UNSOLICITED RESPONSE XML is invalid. Correct and Resubmit.</ns0:FailureReason>
      </ns0:ValidationFailure>
   </ns0:ICDTNotificationFailure>
</ns0:ICDTCommunication>
```

12.1.16 ICDT Administrative Error Response

The RCs can send the following Administrative Error responses for the ICDT Request/Solicited response and Unsolicited responses:

1. The file is corrupt and/or cannot be read; and
2. A virus was found.

Table 25: T.ESD002.ADM.CL1N.ESMD2.D031919.T1334570 shows the sample Administrative error response XML file.

Table 25: T.ESD002.ADM.CL1N.ESMD2.D031919.T1334570

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<ICDTCommunication xmlns="http://cms.hhs.gov/esmd/icdt">
   <ICDTNotificationFailure RequestType="ADMIN_ERROR">
      <creationTime>2019-03-19T13:34:57.665-04:00</creationTime>
      <fileName>D.ESD002.L15_2.V1W5.ESMD2.D030719.T1309450</fileName>
      <Status>
         <description>ERROR: Admin error notification</description>
         <serviceSuccessful>false</serviceSuccessful>
      </Status>
      <ValidationFailure>
         <FailureCode>ESMD_410</FailureCode>
         <FailureReason>File is corrupt and/or cannot be read</FailureReason>
      </ValidationFailure>
   </ICDTNotificationFailure>
</ICDTCommunication>
```

12.1.17 esMD Acknowledgment Response for ADR RRL

When the RC Client sends an ADR RRL to esMD, the esMD application processes the response and sends the acknowledgement response to the RC after successfully validating the response in the esMD system. The esMD system generates the acknowledgement response to RC as shown in Table 26: A_L1_3_MPD000000006788_Receipt_Acknowledgement.xml.

Table 26: A_L1_3_MPD000000006788_Receipt_Acknowledgement.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<ns0:SubmitOutboundRequestOrResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc">
   <ESMDTransaction TransactionId="MPD000000006788" DeliveryType="A" RoutingId="ESD002"/>
   <UniqueID>L13NA3ESD0020227191324310</UniqueID>
   <submissionMetadata>
      <RCOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</RCOID>
      <RCName>Test Review Contractor</RCName>
      <ContentTypeCode>1.3</ContentTypeCode>
      <CaseId>ABC123</CaseId>
      <LetterId>letteridsmp112</LetterId>
      <ReceivedFileName>D.ESMD2.L1_3.UNA3.ESD002.D022719.T1324310</ReceivedFileName>
   </submissionMetadata>
   <Status>
      <description>ESMD accepted ADR Review Result Letter</description>
      <serviceSuccessful>true</serviceSuccessful>
   </Status>
</ns0:SubmitOutboundRequestOrResponseResult>
```

12.1.18 esMD Validation Error Response for ADR RRL

When the RC Client sends an ADR RRL to esMD, the esMD application processes and sends the ADR RRL to the HIH. If there is an error processing the ADR RRL submitted by the RC, the esMD application generates the Validation Error Response as detailed in Table 27: F_L1_3_XSU000000004277_Validation_Error.xml.

Table 27: F_L1_3_XSU000000004277_Validation_Error.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<ns0:SubmitOutboundRequestOrResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc">
   <ESMDTransaction TransactionId="XSU000000004277" DeliveryType="F" RoutingId="ESD002"/>
   <UniqueID>L1312GESD0020116191416520</UniqueID>
   <submissionMetadata>
      <ContentTypeCode>1.3</ContentTypeCode>
      <CaseId>Case12345</CaseId>
      <LetterId>Letter1234</LetterId>
      <ReceivedFileName>D.ESMD2.L1_3.U12G.ESD002.D011619.T1416520</ReceivedFileName>
   </submissionMetadata>
   <Status>
      <description>esMD validation error. Please correct and resubmit.</description>
      <serviceSuccessful>true</serviceSuccessful>
   </Status>
   <ValidationFailure>
      <FailureCode>907</FailureCode>
      <FailureReason>esMD validation error: The combination of Review Contractor OID and the
Content type code received in the request from RC is incorrect. Correct and
resubmit.</FailureReason>
   </ValidationFailure>
</ns0:SubmitOutboundRequestOrResponseResult>
```

12.1.19 HIH Delivery Notification Response for ADR RRL

When the RC Client sends an ADR RRL to esMD, the esMD application processes and sends it to the HIH. Based on the response from the HIH upon delivery of the ADR RRL, the esMD application generates the HIH Delivery Notification (with either success or failure message) as shown in Table 28: N_L1_3_YAU000000006003_Delivery_Acknowledgement.xml

Table 28: N_L1_3_YAU000000006003_Delivery_Acknowledgement.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<ns0:SubmitOutboundRequestOrResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc">
   <ESMDTransaction TransactionId="YAU000000006003" DeliveryType="N"/>
   <UniqueID>L13TPTESD0020215191200390</UniqueID>
   <submissionMetadata>
      <HIHOID>urn:oid:123.456.657.126</HIHOID>
      <HIHName>Test HIH for DBR</HIHName>
      <RCOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</RCOID>
      <RCName>Test Review Contractor</RCName>
```

```
      <ContentTypeCode>1.3</ContentTypeCode>
      <CaseId>ABC123</CaseId>
      <LetterId>letteridsmp112</LetterId>
   </submissionMetadata>
   <Status>
      <description>ADR Review Result Letter - Successfully delivered to HIH</description>
      <serviceSuccessful>true</serviceSuccessful>
   </Status>
</ns0:SubmitOutboundRequestOrResponseResult>
```

12.1.20 esMD Validation Error Response for PA/PCR Decision Letters

When the RC Client sends a PA/PCR Decision Letters package to esMD, the esMD application processes and sends the PA/PCR Decision Letters to the HIH. If there is an error processing the decision letters package submitted by the RC, the esMD application generates the Validation Error Response as detailed in Table 29: R_UID_Validation_Error_Response.xml.

Table 29: R_UID_Validation_Error_Response.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:SubmitOutboundRequestOrResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc">
   <ESMDTransaction TransactionId="XSU000000004277" DeliveryType="F" RoutingId="ESD002"/>
   <UniqueID>L1412GESD0020116191416520</UniqueID>
   <submissionMetadata>
      <ContentTypeCode>1.3</ContentTypeCode>
      <CaseId>Case12345</CaseId>
      <LetterId>Letter1234</LetterId>
      <ReceivedFileName>D.ESMD2.L1_4.U12G.ESD002.D011619.T1416520</ReceivedFileName>
   </submissionMetadata>
   <Status>
      <description>esMD validation error. Please correct and resubmit.</description>
      <serviceSuccessful>true</serviceSuccessful>
   </Status>
   <ValidationFailure>
      <FailureCode>907</FailureCode>
      <FailureReason>esMD validation error: The combination of Review Contractor OID and the
Content type code received in the request from RC is incorrect. Correct and
resubmit.</FailureReason>
   </ValidationFailure>
</ns0:SubmitOutboundRequestOrResponseResult>
```

12.1.21 HIH Delivery Notification Response for PA/PCR Decision Letters

When the RC Client sends a PA/PCR Decision Letters package to esMD, the esMD application processes and sends it to the HIH. Based on the response from the HIH upon delivery of the PA/PCR decision letters, the esMD application generates the HIH Delivery Notification (with either success or failure message) as shown in Table 30: N_L1_4_PBB000000004805_Delivery_Acknowledgement.xml

Table 30: N_L1_4_PBB000000004805_Delivery_Acknowledgement.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:SubmitOutboundRequestOrResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc">
   <ESMDTransaction TransactionId="PBB000000004805" DeliveryType="N"/>
   <UniqueID>L14GV3ESD0020125191605530</UniqueID>
   <submissionMetadata>
      <HIHOID>urn:oid:123.456.657.126</HIHOID>
      <HIHName>Test HIH for DBR</HIHName>
      <RCOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</RCOID>
      <RCName>Test Review Contractor</RCName>
      <ContentTypeCode>1.4</ContentTypeCode>
      <CaseId>Case12345</CaseId>
      <LetterId>Letter1234</LetterId>
   </submissionMetadata>
   <Status>
      <description>PAPCR Decision Letter  -Successfully delivered to HIH</description>
      <serviceSuccessful>true</serviceSuccessful>
   </Status>
</ns0:SubmitOutboundRequestOrResponseResult>
```

12.1.22 esMD Validation Error Response for Pre-Pay eMDR Letters

When the RC Client sends a Pre-Pay eMDR package to esMD, and if there is an error processing the eMDR letters package submitted by the RC, the esMD application generates the Validation Error Response as detailed in Table 31: esMD Validation Error Response for Pre-Pay eMDR Letters.

Table 31: esMD Validation Error Response for Pre-Pay eMDR Letters

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:SubmitOutboundRequestOrResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc">
   <ESMDTransaction TransactionId="IYM000006820412" DeliveryType="F" RoutingId="ESD002"/>
   <UniqueID>IYM000006820412</UniqueID>
   <submissionMetadata>
      <ContentTypeCode>1.5</ContentTypeCode>
      <ReceivedFileName>D.ESMD2.L1_5.U06B.ESD002.D040819.T1607070</ReceivedFileName>
   </submissionMetadata>
   <Status>
      <description>esMD validation error. Please correct and resubmit.</description>
      <serviceSuccessful>true</serviceSuccessful>
   </Status>
   <ValidationFailure>
      <FailureCode>1041</FailureCode>
      <FailureReason>eMDR Process Metadata XML File is missing.</FailureReason>
   </ValidationFailure>
</ns0:SubmitOutboundRequestOrResponseResult>
```

12.1.23 esMD Validation Error Response for Post-Pay eMDR Letters

When the RC Client sends a Post-Pay eMDR package to esMD, and if there is an error processing the eMDR letters package submitted by the RC, the esMD application

generates the Validation Error Response as detailed in Table 32: esMD Validation Error Response for Post-Pay eMDR Letters.

Table 32: esMD Validation Error Response for Post-Pay eMDR Letters

```
<?xml version="1.0" encoding="UTF-8"?>

<ns0:SubmitOutboundRequestOrResponseResult xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc">
   <ESMDTransaction TransactionId="HCH000006829669" DeliveryType="F" RoutingId="ESD002"/>
   <UniqueID>HCH000006829669</UniqueID>
   <submissionMetadata>
      <ContentTypeCode>1.6</ContentTypeCode>
      <ReceivedFileName>D.ESMD2.L1_6.U6RC.ESD002.D070219.T1610570</ReceivedFileName>
   </submissionMetadata>
   <Status>
      <description>esMD validation error. Please correct and resubmit.</description>
      <serviceSuccessful>true</serviceSuccessful>
   </Status>
   <ValidationFailure>
      <FailureCode>1029</FailureCode>
      <FailureReason>esMD validation error: eMDR Process Metadata XML File for ADR EMDR
POSTPAY LETTERS is missing. Please resubmit.</FailureReason>
   </ValidationFailure>
</ns0:SubmitOutboundRequestOrResponseResult>
```

12.1.24 Document Code File

Document code file is a flat file sent by esMD on a Quarterly basis. Table 33: Document Code File shows the sample Document Code File.

Table 33: Document Code File

```
U20200131143015DOCUCODE

V0000001UBPDDJVThe long description of the document which is getting requested

V0000001UBPDDJVThe long description of the document which is getting requested

V0000001UBPDDJVThe long description of the document which is getting requested

V0000001UBPDDJVThe long description of the document which is getting requested

V0000001UBPDDJVThe long description of the document which is getting requested

V0000001UBPDDJVThe long description of the document which is getting requested

V0000001UBPDDJVThe long description of the document which is getting requested

W0000007
```

12.2 Outbound

Note for API users: Please refer to the properties files packaged with the source code for more details on the reference data needed to populate the outbound XMLs described in this section.

The RC Client transfers the following messages during the outbound process:

1. Pickup Notification;
2. Error Pickup Notification;
3. Review Decision Response to PA Request;
4. Error Response to PA request;
5. Administrative Error Response to Inbound Submissions;
6. esMD Process Metadata (eMDR Request);
7. ADR Review Response XML;
8. esMD Process Metadata (ADR Review Response)
9. ICDT Request;
10. ICDT Solicited Response;
11. ICDT Unsolicited Response;
12. ICDT Pickup/ Pickup Error Notification;
13. ICDT Administrative Error Notification;
14. ADR RRL;
15. esMD Process Metadata (ADR RRL);
16. Pickup Notification for Service Registration;
17. Pickup Notification for Document Codes;
18. PA/PCR Decision Letters;
19. eMDR Process Metadata;
20. eMDR Structured File for Post-Pay ADR letters; and
21. API Error Messages for Pre-Pay and Post-Pay.
22. DCF Pickup Notification
23. DCF Error Pickup Notification
24. HOPD Pickup Notification

12.2.1 Pickup Notification

The RC Client generates pickup notifications for all inbound files with delivery type “E” pulled from TIBCO MFT server and processed successfully, as detailed in Table 34: P_L1_L23_Pickup.xml.

Table 34: P_L1_L23_Pickup.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RCPickupNotification xmlns:ns2="http://esmd.ois.cms.hhs.gov/v1/rc/config">
   <ESMDTransactionId> DLL00000000952</ESMDTransactionId>
   <RoutingId>ESD002</RoutingId>
   <PickupTime>2019-03-15T13:02:23.218-04:00</PickupTime>
   <SubmissionTime>2019-03-15T13:02:23.218-04:00</SubmissionTime>
   <fileName>D.ESD002.L1.EL23.ESMD2.D190315.T1202000</fileName>
</ns2:RCPickupNotification>
```

12.2.2 Error Pickup Notification

The RC Client generates pickup error notifications for all inbound files pulled from TIBCO MFT and processed unsuccessfully, as detailed in Table 35: P_L1_L23_Pickup_Error.xml The processing errors are generated in two scenarios:

1. Checksum verification failed (i.e., the payload file received by the RC client does not match the file sent by esMD); and
2. Extraction was unsuccessful (i.e., the RC client could not successfully unzip the file received from the server).

Table 35: P_L1_L23_Pickup_Error.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RCPickupNotification xmlns:ns2="http://esmd.ois.cms.hhs.gov/v1/rc/config">
   <ESMDTransactionId>L23</ESMDTransactionId>
   <RoutingId>ES9999</RoutingId>
   <PickupTime>2019-03-15T13:02:23.218-04:00</PickupTime>
   <SubmissionTime>2019-03-15T13:02:23.218-04:00</SubmissionTime>
   <fileName>D.ESD002.L1.EL23.ESMD2.D190315.T1202000</fileName>
   <ErrorInfo>
      <ErrorCode>535</ErrorCode>
      <ErrorName> ERROR VERIFYING PAYLOAD CHECKSUM </ErrorName>
      <ErrorDescription> ESMD_535 - RC Client processing error (Checksum issue).  Please
resubmit.</ErrorDescription>
   </ErrorInfo>
</ns2:RCPickupNotification>
```

12.2.3 Error Response to PA Request

The Error Response to PA Request is the XML message from the RC to the HIH, to inform the HIH of the review result response with decision as “Rejected” as detailed in Table 36: R_PA_LQA000000006905_Response.xml.

Please refer to the Appendix B: Reject Error Codes for more information on the error codes used in the Error Review Response for a PA Request.

Table 36: R_PA_LQA000000006905_Response.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:SubmitPADeterminationResponse xmlns:ns2="http://esmd.ois.cms.hhs.gov/v2/rc"
xmlns:ns3="http://esmd.ois.cms.hhs.gov/v2/rc/cmsbt">
   <ESMDTransaction TransactionId="LQA000000006905" DeliveryType="R" RoutingId="ESD002"/>
   <PAReviewResponse>
      <CreationTime>2019-03-19T14:09:27.299-04:00</CreationTime>
      <SubmissionTime>2019-03-19T14:09:46.313-04:00</SubmissionTime>
      <ErrorResponseDetail>
         <DecisionIndicator>R</DecisionIndicator>
         <RejectErrorCodeRecordList>
            <RejectErrorCodeRecord>
               <ErrorCategoryName>Medical-Info</ErrorCategoryName>
               <ErrorCodeRecordList>
                  <ErrorCodeRecord>
                     <ErrorCode>AG</ErrorCode>
                     <ErrorCodeDescription>Incorrect Modifier for the Procedure
Code</ErrorCodeDescription>
                  </ErrorCodeRecord>
               </ErrorCodeRecordList>
            </RejectErrorCodeRecord>
         </RejectErrorCodeRecordList>
         <ReasonCodeList>
            <ReasonCode>AF891</ReasonCode>
         </ReasonCodeList>
         <RequestLevelUniqueTrackingNumber>testUtn2356</RequestLevelUniqueTrackingNumber>
      </ErrorResponseDetail>
   </PAReviewResponse>
</ns2:SubmitPADeterminationResponse>
```

12.2.4 Administrative Error Response to Inbound Submissions

The Administrative Error Response is the XML message from the RC to the HIH to inform the HIH of the administrative error response to inbound submissions.

Note: Section 5 How to Submit an Inbound Submission Error on the Administrative Error Response to Inbound Submissions Tab describes the process of creating an XML message, using the RC Client.

Table 37: D_ADM_EPP000000008983_AdminResponse.xml provides the PA Review Decision Response XML.

Table 37: D_ADM_EPP000000008983_AdminResponse.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:SubmitAdministrativeErrorResponse xmlns:ns2="http://esmd.ois.cms.hhs.gov/v2/rc"
xmlns:ns3="http://esmd.ois.cms.hhs.gov/v2/rc/cmsbt">
   <ESMDTransaction TransactionId="EPP000000008983" DeliveryType="D" RoutingId="ESD002"/>
   <AdministrativeErrorResponse>
      <CreationTime>2019-03-19T14:15:47.133-04:00</CreationTime>
```

```
      <SubmissionTime>2019-03-19T14:16:18.847-04:00</SubmissionTime>
      <ErrorResponseList>
         <ErrorResponseRecord>
            <ErrorCode>ESMD_414</ErrorCode>
            <ErrorName>Incomplete File</ErrorName>
         </ErrorResponseRecord>
      </ErrorResponseList>
   </AdministrativeErrorResponse>
</ns2:SubmitAdministrativeErrorResponse>
```

12.2.5 ICDT Request

The RCs send the ICDT Request to another RC via esMD in XML format as part of the ICDT Request Package with delivery type “Q”. The file name of the ICDT Request should contain only alphanumeric characters and underscores (i.e., “_”).

Table 38: Q_L1518DMESD0020315191038490_ICDTSolicitedRequest.xml shows the XML message generated for an ICDT Request XML from the RCs.

Table 38: Q_L1518DMESD0020315191038490_ICDTSolicitedRequest.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ICDTRequest xmlns="http://cms.hhs.gov/esmd/icdt">
   <receiverOID>urn:oid:126.543.321.123</receiverOID>
   <receiverID>01232</receiverID>
   <senderOID>urn:oid:126.543.321.121</senderOID>
   <senderID>01231</senderID>
   <requestID> L1518DMESD0020315191038490</requestID>
   <contentType>15.1</contentType>
   <TransactionType transType="Claim">
      <OptionalMetadata>
         <FieldName>CLAIM_ID</FieldName>
         <FieldValue>Claim ID 12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>CASE_ID</FieldName>
         <FieldValue>CASE ID 12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>NPI</FieldName>
         <FieldValue>1234567890</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>HICN</FieldName>
         <FieldValue>HICN12345678910</FieldValue>
      </OptionalMetadata>
   </TransactionType>
</ICDTRequest>
```

12.2.6 ICDT Solicited Response

The RCs send the ICDT Response Document to another RC via esMD in any file format **except executable files** as part of the ICDT Response Package with delivery type “R”. The ICDT Solicited Response is sent for the Request from another RC. The file name of the ICDT Response Document should contain only alphanumeric characters and underscores (i.e., “_”).

Table 39: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml shows the sample XML Message of the ICDT Response sent from the RCs.

Table 39: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ICDTResponse xmlns="http://cms.hhs.gov/esmd/icdt">
   <receiverOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.4</receiverOID>
   <receiverID>01232</receiverID>
   <senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.3</senderOID>
   <senderID>01231</senderID>
   <requestID>QAC5K9XTESD0011210162133560</requestID>
   <contentType>15.2</contentType>
   <responseID>L1521R7ESD0020130191302560</responseID>
   <TransactionType transType="Claim">
      <OptionalMetadata>
         <FieldName>CLAIM_ID</FieldName>
         <FieldValue>Claim ID 12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>CASE_ID</FieldName>
         <FieldValue>CASE ID 12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>NPI</FieldName>
         <FieldValue>1234567890</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>HICN</FieldName>
         <FieldValue>HICN123456</FieldValue>
      </OptionalMetadata>
   </TransactionType>
   <Documentation FileName="pdf-sample_1.pdf" MimeType="application/pdf" DocUniqueID="pdf-
sample_1">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>fc80d59877b4ae21911591b53664b2da1324cf25</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>7945</FieldValue>
      </OptionalMetadata>
   </Documentation>
   <numberOfDocuments>1</numberOfDocuments>
</ICDTResponse>
```

12.2.7 ICDT Unsolicited Response

The RCs send the ICDT Unsolicited Response Document to another RC via esMD in any file format **except executable files** as part of the ICDT Response Package with delivery type “R”. The file name of the ICDT Response Document should contain only alphanumeric characters and underscores (i.e., “_”).

Table 40: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml shows the sample XML Message of the ICDT Unsolicited Response sent from the RCs.

Table 40: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ICDTResponse xmlns="http://cms.hhs.gov/esmd/icdt">
   <receiverOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.4</receiverOID>
   <receiverID>01232</receiverID>
   <senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.3</senderOID>
   <senderID>01231</senderID>
   <contentType>15.3</contentType>
   <responseID>L1535RJES99960308191422450</responseID>
   <TransactionType transType="SMRC-Misroute">
      <OptionalMetadata>
         <FieldName>CLAIM_ID</FieldName>
         <FieldValue>Claim ID 12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>CASE_ID</FieldName>
         <FieldValue>CASE ID 12345678910</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>NPI</FieldName>
         <FieldValue>1234567890</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>HICN</FieldName>
         <FieldValue>HICN12345</FieldValue>
      </OptionalMetadata>
   </TransactionType>
   <Documentation FileName="pdf-sample_1.pdf" MimeType="application/pdf" DocUniqueID="pdf-
sample_1">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>fc80d59877b4ae21911591b53664b2da1324cf25</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>7945</FieldValue>
      </OptionalMetadata>
   </Documentation>
   <Documentation FileName="pdf-sample_2.pdf" MimeType="application/pdf" DocUniqueID="pdf-
sample_2">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>fc80d59877b4ae21911591b53664b2da1324cf25</FieldValue>
```

```
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>7945</FieldValue>
      </OptionalMetadata>
   </Documentation>
   <numberOfDocuments>2</numberOfDocuments>
</ICDTResponse>
```

12.2.8 ICDT Pickup/Pickup Error Notification

When the RC Client sends an ICDT Request or ICDT Response to esMD, the esMD application processes the response and sends the acknowledgement response to the RC after successfully validating the response in the esMD system. The Recipient RC downloads the package and sends the successful pickup notification to esMD.

The esMD system generates the acknowledgement response to RC as shown in Table 41: T.ESD002.ICDT.BJ99.ESMD2.D031519.T1201000, and sends it to the RC.

Table 41: T.ESD002.ICDT.BJ99.ESMD2.D031519.T1201000

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:ICDTCommunication xmlns:ns0="http://cms.hhs.gov/esmd/icdt">
   <ns0:ICDTNotification RequestType="SOLIC_REQ_ACK" id="1">
      <ns0:ICDTMetaData>
         <ns0:receiverOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.50</ns0:receiverOID>
         <ns0:receiverID>01232</ns0:receiverID>
         <ns0:senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</ns0:senderOID>
         <ns0:senderID>01231</ns0:senderID>
         <ns0:requestID>L1518DMESD0020315191038490</ns0:requestID>
         <ns0:contentType>15.1</ns0:contentType>
      </ns0:ICDTMetaData>
      <ns0:creationTime>2019-03-15T12:01:00.335-04:00</ns0:creationTime>
      <ns0:fileName>D#EFT.ON.ESMD2.L15_1.Q8DM.ESD002.D031519.T1038490</ns0:fileName>
      <ns0:Status>
         <ns0:serviceSuccessful>true</ns0:serviceSuccessful>
      </ns0:Status>
   </ns0:ICDTNotification>
   <ns0:ICDTNotification RequestType="SOLIC_REQ_ACK" id="2">
      <ns0:ICDTMetaData>
         <ns0:receiverOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.50</ns0:receiverOID>
         <ns0:receiverID>01232</ns0:receiverID>
         <ns0:senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</ns0:senderOID>
         <ns0:senderID>01231</ns0:senderID>
         <ns0:requestID>L151C2WESD0020315191038580</ns0:requestID>
         <ns0:contentType>15.1</ns0:contentType>
      </ns0:ICDTMetaData>
      <ns0:creationTime>2019-03-15T12:01:00.335-04:00</ns0:creationTime>
      <ns0:fileName>D#EFT.ON.ESMD2.L15_1.QC2W.ESD002.D031519.T1038580</ns0:fileName>
      <ns0:Status>
         <ns0:serviceSuccessful>true</ns0:serviceSuccessful>
      </ns0:Status>
   </ns0:ICDTNotification>
```

```
</ns0:ICDTCommunication>
```

12.2.9 ICDT Administrative Error Notification

The RCs can send the following Administrative Error responses for the ICDT request/Solicited response and Unsolicited responses:

1. The file is corrupt and/or cannot be read; and
2. A virus was found.

Table 42: D#EFT.ON.ESMD2.ADM.C2K2.ESD002.D031919.T1436180 shows the sample Administrative error response XML file.

Table 42: D#EFT.ON.ESMD2.ADM.C2K2.ESD002.D031919.T1436180

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ICDTCommunication xmlns="http://cms.hhs.gov/esmd/icdt">
   <ICDTNotificationFailure RequestType="ADMIN_ERROR">
      <creationTime>2019-03-19T14:36:18.854-04:00</creationTime>
      <fileName>D.ESD002.L15_2.V1W5.ESMD2.D030719.T1309450</fileName>
      <Status>
         <description>ERROR: Admin error notification</description>
         <serviceSuccessful>false</serviceSuccessful>
      </Status>
      <ValidationFailure>
         <FailureCode>ESMD_410</FailureCode>
         <FailureReason>File is corrupt and/or cannot be read</FailureReason>
      </ValidationFailure>
   </ICDTNotificationFailure>
</ICDTCommunication>
```

12.2.10 ADR RRL

The RCs send the ADR RRLs to the HIH via esMD in a PDF file as part of the ADR Review Results Package with delivery type “U”. The file name of the ADR RRL may contain only alphanumeric characters and underscores (i.e., “_”). More than one PDF file can be included in the ADR RRL package sent to esMD.

12.2.11 esMD Process Metadata (ADR RRL)

The metadata file accompanies the ADR RRLs as the outbound document package with the delivery type “U”. The metadata file contains information about the ADR RRL Package including the unique ID, sender and HIH Organizational Identifier (OID), submission metadata (sender routing ID, Content Type Code, Case ID, creation time, and National Provider Identifier (NPI)), and documentation information (Document Unique Identifier, Multipurpose Internet Mail Extension (MIME) Type, File Name, and Check Sum Value). Refer to Table 43: U_L13NFEESD0020319191438140_esMD_ProcessMetadata.xml.

Table 43: U_L13NFEESD0020319191438140_esMD_ProcessMetadata.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<esMDProcessMetadata xmlns="http://esmd.ois.cms.hhs.gov/rc/esMDProcessMetadata">
   <uniqueID>L13NFEESD0020319191438140</uniqueID>
   <numberOfDocuments>1</numberOfDocuments>
   <hihOID>urn:oid:123.456.657.126</hihOID>
   <senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</senderOID>
   <submissionMetadata>
      <creationTime>2019-03-19T14:38:14.803-04:00</creationTime>
      <routingName>ESD002</routingName>
      <deliveryType>U</deliveryType>
      <contentTypeCode>1.3</contentTypeCode>
      <letterID>Letter1234</letterID>
      <caseID>Case12345</caseID>
      <npi>1234567890</npi>
   </submissionMetadata>
   <Documentation DocumentUniqueIdentifier="sample_adr_letter2" MimeType="application/pdf"
FileName="sample_adr_letter2.pdf">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>fc80d59877b4ae21911591b53664b2da1324cf25</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>7945</FieldValue>
      </OptionalMetadata>
   </Documentation>
</esMDProcessMetadata>
```

12.2.12 Service Registration Pickup Notification

After picking up the Service Registration Request batch file, the RC generates the pickup notification and sends to esMD. Refer to Table 44: P_L5_L23_Pickup.xml for sample pickup notification.

Table 44: P_L5_L23_Pickup.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RCPickupNotification xmlns:ns2="http://esmd.ois.cms.hhs.gov/v1/rc/config">
   <ESMDTransactionId>L23</ESMDTransactionId>
   <RoutingId>ESD002</RoutingId>
   <PickupTime>2019-03-15T13:02:23.218-04:00</PickupTime>
   <SubmissionTime>2019-03-15T13:02:23.218-04:00</SubmissionTime>
   <fileName>D.ESD002.L5.EL23.ESMD2.D190315.T1202000</fileName>
</ns2:RCPickupNotification>
```

12.2.13 PA/PCR Decision Letters

The RCs send the PA/PCR Decision Letters to the HIH via esMD in a PDF file as part of the PA/PCR Decision Letters package with delivery type “U”. The file name of the decision letter may contain only alphanumeric characters and underscores (i.e., “_”). More than one PDF file can be included in the PA/PCR Decision Letters package sent to esMD.

12.2.14 esMD Process Metadata (PA/PCR Decision Letters)

The metadata file accompanies the PA/PCR Decision letters as the outbound document package with the delivery type “U”. The metadata file contains information about the PA/PCR Decision letters Package including the unique ID, sender and HIH OID, submission metadata (sender routing ID, Content Type Code, Case ID, creation time, and NPI), and documentation information (Document Unique Identifier, MIME Type, File Name, and Check Sum Value). Refer to Table 45: U_L14NFEESD0020319191438140_esMD_ProcessMetadata.xml

Table 45: U_L14NFEESD0020319191438140_esMD_ProcessMetadata.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<esMDProcessMetadata xmlns="http://esmd.ois.cms.hhs.gov/rc/esMDProcessMetadata">
   <uniqueID>L14NFEESD0020319191438140</uniqueID>
   <numberOfDocuments>1</numberOfDocuments>
   <hihOID>urn:oid:123.456.657.126</hihOID>
   <senderOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</senderOID>
   <submissionMetadata>
      <creationTime>2019-03-19T14:38:14.803-04:00</creationTime>
      <routingName>ESD002</routingName>
      <deliveryType>U</deliveryType>
      <contentTypeCode>1.4</contentTypeCode>
      <letterID>Letter1234</letterID>
      <caseID>Case12345</caseID>
      <npi>1234567890</npi>
   </submissionMetadata>
   <Documentation DocumentUniqueIdentifier="sample_adr_letter2" MimeType="application/pdf"
FileName="sample_adr_letter2.pdf">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>fc80d59877b4ae21911591b53664b2da1324cf25</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>7945</FieldValue>
      </OptionalMetadata>
   </Documentation>
</esMDProcessMetadata>
```

12.2.15 eMDR Process Metadata (Pre-Pay and Post-Pay eMDR letters)

The eMDR Process metadata file accompanies the ADR letters zip files for Pre-Pay and Post-Pay functionality as the outbound document package with the delivery type “U”.

The sample XML files are included in Table 46: Sample eMDRProcessMetadata XML. The same eMDR ProcessMetadata schema is used for Pre-Pay and Post-Pay functionality.

Table 46: Sample eMDRProcessMetadata XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<eMDRProcessMetadata xmlns="http://esmd.ois.cms.hhs.gov/rc/v1/emdr/processmetadata">
   <uniqueID>L15RUMESD0020502191356200</uniqueID>
   <numberOfDocuments>3</numberOfDocuments>
   <submissionMetadata>
      <creationTime>2019-05-02T13:56:20.561-04:00</creationTime>
      <routingName>ESD002</routingName>
      <deliveryType>U</deliveryType>
      <contentTypeCode>1.5</contentTypeCode>
   </submissionMetadata>
   <Documentation DocumentUniqueIdentifier="TESTLETTERID4478_20190405&amp;amp;_REVIEW3"
MimeType="application/pdf" FileName="TESTLETTERID4478_20190405&amp;amp;_REVIEW3.pdf">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>8c84d07d505ad1429725e5f09a79daf96465592d</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>105255200</FieldValue>
      </OptionalMetadata>
   </Documentation>
   <Documentation DocumentUniqueIdentifier="TESTLETTERID5678_20190405_REVIEW1"
MimeType="application/pdf" FileName="TESTLETTERID5678_20190405_REVIEW1.pdf">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>fc80d59877b4ae21911591b53664b2da1324cf25</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>7945</FieldValue>
      </OptionalMetadata>
   </Documentation>
   <Documentation DocumentUniqueIdentifier="TESTLETTERID9978_20190405_REVIEW2"
MimeType="application/pdf" FileName="TESTLETTERID9978_20190405_REVIEW2.pdf">
      <OptionalMetadata>
         <FieldName>CheckSum</FieldName>
         <FieldValue>fc80d59877b4ae21911591b53664b2da1324cf25</FieldValue>
      </OptionalMetadata>
      <OptionalMetadata>
         <FieldName>File Size</FieldName>
         <FieldValue>7945</FieldValue>
```

```
    </OptionalMetadata>
  </Documentation>
</eMDRProcessMetadata>
```

12.2.16 eMDR StructuredFile for Post pay ADR letters

Table 47: U_L16Y5XESD0020726192103460_eMDRStructuredFile.xml shows the sample eMDR structured ADR letter file for Post-Pay eMDR letters.

Table 47: U_L16Y5XESD0020726192103460_eMDRStructuredFile.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<EMDRPostPayRequest xmlns="http://esmd.ois.cms.hhs.gov/v3/rc/esmd/emdr/postpay">
  <eMDRType>POST-PAY</eMDRType>
  <uniqueLetterId>Letter2</uniqueLetterId>
  <letterDate>2019-03-13</letterDate>
  <respondTo>
    <organizationName>Keebler, Halvorson and Murphy</organizationName>
    <addressLine1>28569 Conn Manors</addressLine1>
    <city>Darionland</city>
    <state>RI</state>
    <zipCode>193084149</zipCode>
  </respondTo>
  <senderDetails>
    <organizationName>Marvin, Gleason and Hermiston</organizationName>
  </senderDetails>
  <providerDetails>
    <organizationOrlastName>Cartwright</organizationOrlastName>
    <addressLine1>4240 Isabel Ports</addressLine1>
    <city>South Janelle</city>
    <state>ME</state>
    <zipCode>410257281</zipCode>
    <npi>1932695756</npi>
  </providerDetails>
  <letterDetails>
    <respondBy>2019-09-04</respondBy>
    <jurisdiction>DW</jurisdiction>
    <programName>PARTB</programName>
  </letterDetails>
  <reviewLevelRecordList>
    <reviewLevel>
      <analysisRecordList>
        <analysisRecord>
          <analysisID>EKNSUKTAKTSHMSQNVBDA</analysisID>
          <claimLevelItemList>
            <claimSetLevel>
              <claimDetails>
                <claimID>BCQWQLZNJXKFFQUMVEAF</claimID>
                <beneficiaryID>BKSKIKK</beneficiaryID>
                <beneficiaryLastName>Bogan</beneficiaryLastName>
              </claimDetails>
            </claimSetLevel>
```

```
            </claimLevelItemList>
          </analysisRecord>
        </analysisRecordList>
      </reviewLevel>
    </reviewLevelRecordList>
</EMDRPostPayRequest>
```

12.2.17 API Error Messages for Pre-Pay and Post-Pay

Table 48: RC Client Error Codes and Error Messages lists the validation error messages while generating the Pre-Pay and Post-Pay zip packages in the RC Client API.

Table 48: RC Client Error Codes and Error Messages

ID	Scenario	Error Code	Error Message
1	Attachments other than PDF format for Pre-Pay packages	ADR_ESMD_LETTERS_FILE_INVALID_ATTACHMENT	ADR esMD Letters File must be in PDF format
2	File size is 0 MB or greater than 140 MB in size for Pre-Pay packages	ADR_ESMD_LETTERS_FILE _EXCEEDS_MAX_LIMIT	ADR esMD PDF Letters exceeds 140 MB
3	Missing ADR letter file	MISSING_ADR_ESMD_LETTERS_FILE _ERR_CODE	ADR esMD Letters File attachment is missing. One or more attachments is required
4	eMDR Process metadata parsing error	EMDR_PROCESS_METADATA_PARSING_ERR_CODE	Error Parsing the eMDR Process Metadata XML file.
5	File size is 0 MB or greater than 140 MB in size for Post-Pay packages	ADR_LETTER_PACKAGE_FILE_EQUALS_MIN_LIMIT_OR_EXCEEDS_MAX_LIMIT	ADR Letter Package file size is 0 MB or exceeds 140 MB
6	Attachments other than PDF format for Post-Pay packages	INVALID_FILE_EXTENSION_FOR_ADR_LETTER_IN_PDF	Invalid File Extension for ADR PDF Letter
7	Missing eMDR Structured file or ADR letter file	EMDR_STRUCTURED_FILE_OR_ADR_LETTER_IN_PDF_MISSING_OR_MULTIPLE_FOR_ADR_LETTER_PACKAGE	eMDR Structured XML File and/or ADR PDF Letter missing or more than one for ADR Letter Package
8	Structured XML file cannot be parsed	EMDR_STRUCTURED_FILE_PARSING_FAILURE	eMDR Structured XML File cannot be parsed

12.2.18 DCF Pickup Notification

RC Client after downloading the Document Code file, generates the successful pickup notification when all the edits are validated successfully. Table 49: P_L17_3C8_Pickup.xml shows the sample Pickup notification XML.

Table 49: P_L17_3C8_Pickup.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<ns2:RCPickupNotification xmlns:ns2="http://esmd.ois.cms.hhs.gov/v1/rc/config">
   <ESMDTransactionId>3C8</ESMDTransactionId>
   <RoutingId>ESD002</RoutingId>
   <PickupTime>2020-04-13T15:45:39.652-04:00</PickupTime>
   <SubmissionTime>2020-04-13T15:45:39.652-04:00</SubmissionTime>
   <fileName>T.ESD002.L17.E3C8.ESMD2.D041020.T160610</fileName>
</ns2:RCPickupNotification>
```

12.2.19 DCF Error Pickup Notification

If there are any validation edits failure, RC Client API will generate error pickup notification XML back to esMD. Table 50: DCF Error Pickup Notification shows the sample error pickup notification XML.

Table 50: DCF Error Pickup Notification

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<RCPickupNotification xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://esmd.ois.cms.hhs.gov/v1/rc/config">
        <ESMDTransactionId xmlns="">ZLM</ESMDTransactionId>
        <RoutingId xmlns="">ESD002</RoutingId>
        <PickupTime xmlns="">2020-02-12T14:54:04.599-05:00</PickupTime>
        <SubmissionTime xmlns="">2020-02-12T14:54:04.599-05:00</SubmissionTime>
        <ErrorInfo xmlns="">
                <ErrorCode>515</ErrorCode>
                <ErrorName>ERROR DOCUMENT CODES VALIDATE FILE</ErrorName>
                <ErrorDescription> Invalid line length for line 1; Expected: 1035, Actual: 1021
Invalid line length for line 2; Expected: 1035, Actual: 1028
Invalid line length for line 7; Expected: 1035, Actual: 1029
Invalid line length for line 8; Expected: 1035, Actual: 1029
Invalid line length for line 9; Expected: 1035, Actual: 1025
                </ErrorDescription>
        </ErrorInfo>
        <fileName xmlns="">T.ESD002.L17.EZLM.ESMD2.D200131.T0803010_Invalid</fileName>
</RCPickupNotification>
```

12.2.20 HOPD Pickup notification

RC Client API after downloading the HOPD PA program generates the pickup notification back to esMD. Table 51: HOPD Pickup Notification shows the sample HOPD pickup notification.

Table 51: HOPD Pickup Notification

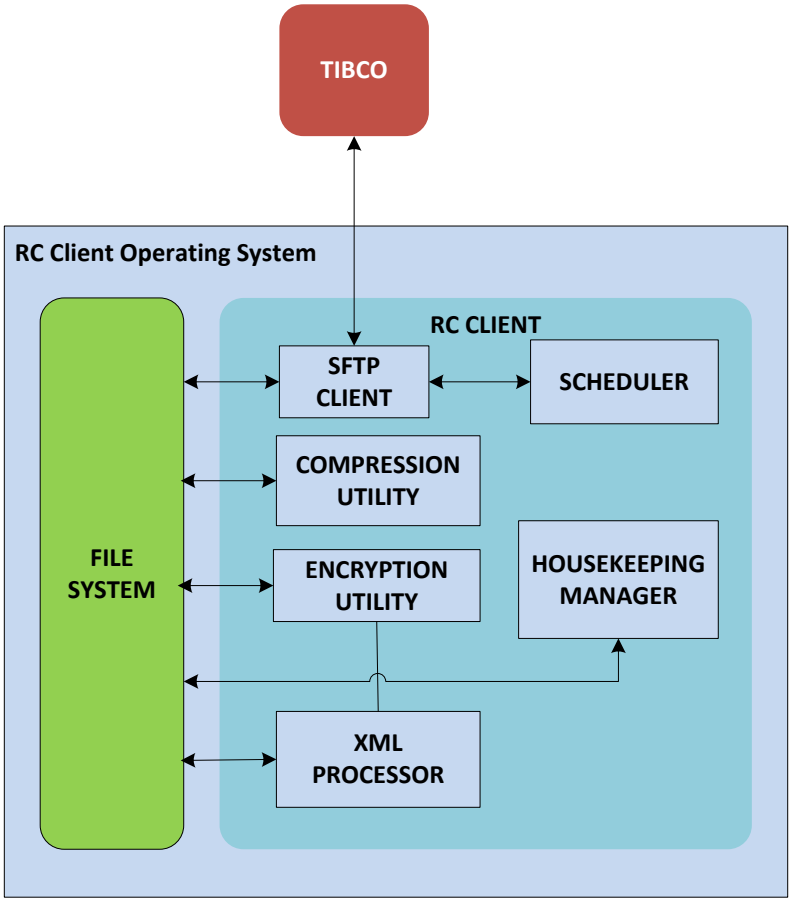
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RCPickupNotification xmlns:ns2="http://esmd.ois.cms.hhs.gov/v1/rc/config">
```

```
    <ESMDTransactionId>PWI000007041335</ESMDTransactionId>
    <RoutingId>ESD002</RoutingId>
    <PickupTime>2020-04-13T15:44:36.638-04:00</PickupTime>
    <SubmissionTime>2020-04-13T12:19:03.093-04:00</SubmissionTime>
    <fileName>T.ESD002.L8_5.EYVE.ESMD2.D041320.T121903103</fileName>
</ns2:RCPickupNotification>
```

13. RC Client Components

Figure 8: RC Client Components shows the internal components of RC Client application. The following sections describe each component in detail.

Figure 8: RC Client Components



13.1 SFTP Client

The SFTP Client is an internal component of the RC Client. It provides the following functionality:

1. Connects to the TIBCO MFT server using IDM ID;
2. Lists the available documents on the TIBCO MFT server;
3. Pulls the documents to the RC Client; and
4. Pushes the outbound documents from RC Client to the TIBCO MFT server.

13.2 Compression Utility

The Compression utility allows the RC Client to extract the payload, metadata file, and messages from the compressed file downloaded from the TIBCO MFT server. The RC Client uses the zip file format.

The same utility is used to create compressed file logs for extraction.

13.3 Encryption Utility

The Encryption utility encrypts the login credentials that will be stored in memory for the duration of the RC Client program execution. The Encryption utility is described in detail in Section 16.1 Security.

13.4 XML Processor

The XML Processor supports creating XML messages to send to esMD as well as loading the configuration files for the RC Client.

13.5 Scheduler

After the RC Client starts, the polling cycle begins. The poll is a redundant cycle; you can configure the interval (for example, 1 hour or 4 hours) through the RC Client property file. The Schedule component controls the RC Client threads and ensures the RC Client runs in regular intervals determined by the “checkFrequency” parameter in the XML Configuration File.

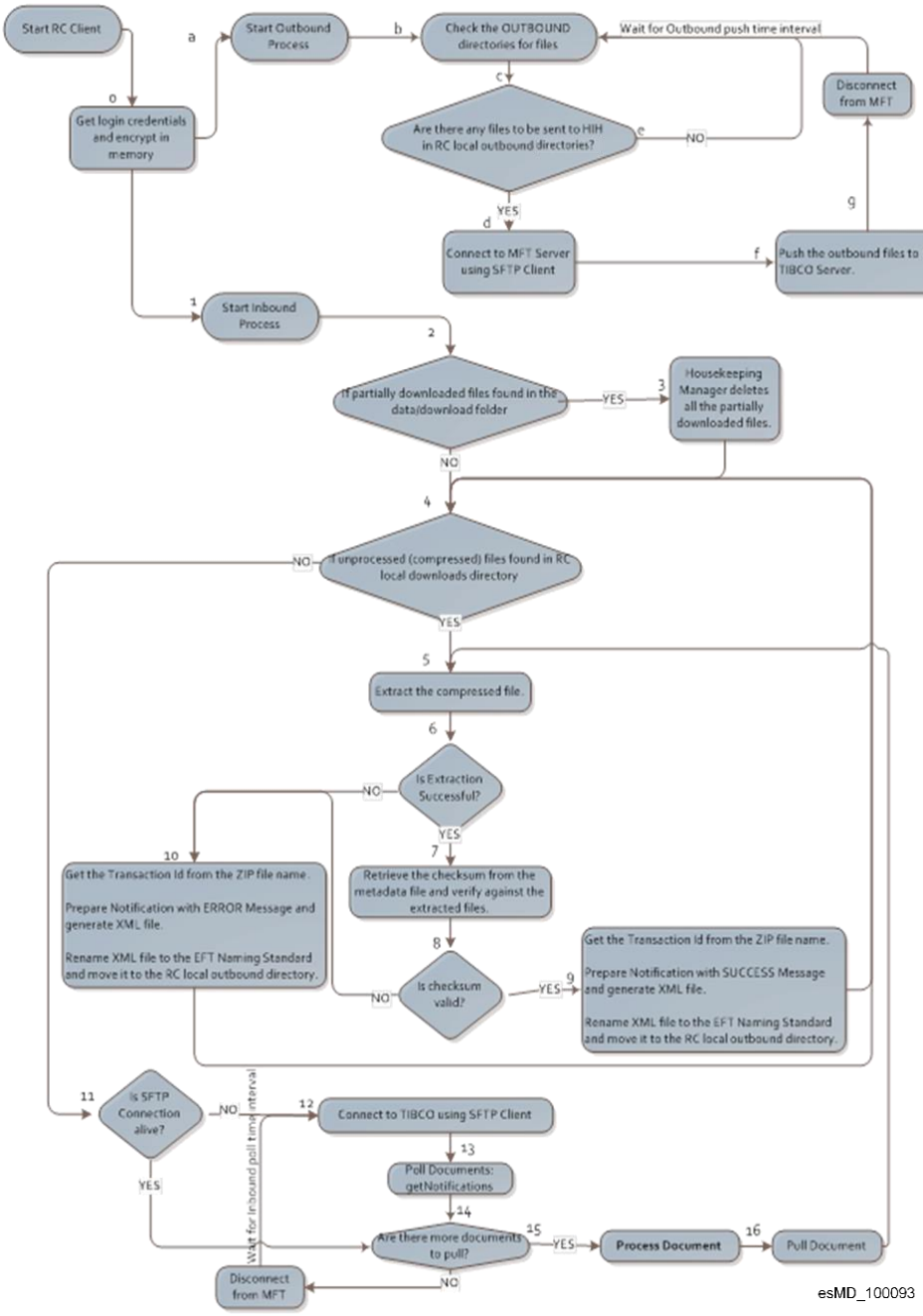
13.6 Housekeeping Manager

The Housekeeping Manager allows the RC Client to recover from any abnormal terminations with the exception of a Virus lockdown. In this situation, the RC Client does not enable recovery, and the RC must contact the esMD Service Desk.

14. RC Client Workflow

The workflow associated with Figure 8: RC Client Components is broken down in Figure 9: RC Client Workflow, followed by a detailed description of the workflow.

Figure 9: RC Client Workflow



14.1 ICDT Request/Response Business Process Flow

This section describes the process flow of the ICDT Request and ICDT Solicited/UnSolicited Response sent from one RC to another RC via the esMD application. Figure 10: ICDT Request/Response Business Process Flow Diagram shows the process flow of ICDT Request and Response, and Table 62: Manual Submission of PA and HHPCR Error (Rejected Decision) Response provides the detailed steps.

Figure 10: ICDT Request/Response Business Process Flow Diagram

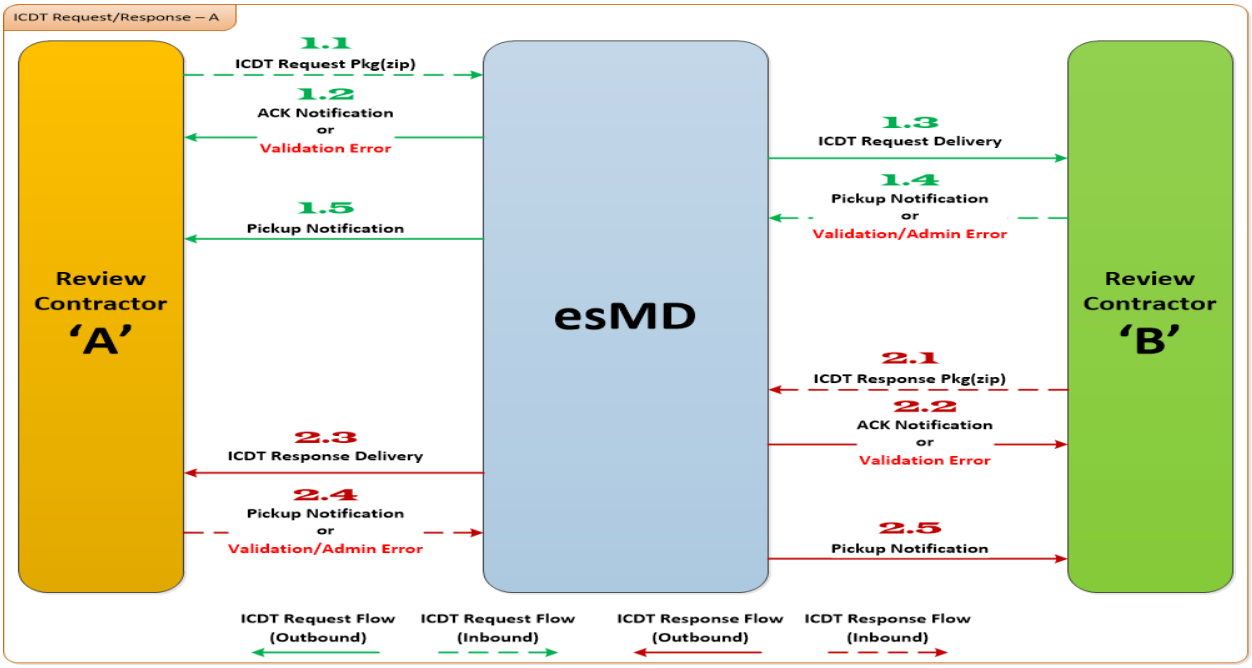


Table 52: ICDT Request/Response Business Process Flow Steps

Message Sequence	Description
1.1	Review Contractor 'A' creates the ICDT Request package, which consists of ICDT Request XML file, and sends the package to esMD via the TIBCO MFT Server to be delivered to another RC.
1.2	esMD sends the Acknowledgement notification to the Review Contractor 'A' if the validation of the ICDT Request package is successful or the esMD system sends the validation errors for any failures.
1.3	esMD delivers the ICDT Request package to the Review Contractor 'B' if the validation is successful.
1.4	Review Contractor 'B' downloads the ICDT Request Package and sends the Successful Pickup notification, Error pickup notification, or admin errors to esMD via the TIBCO MFT server.

Message Sequence	Description
1.5	esMD delivers the Pickup notification, error pickup notification, or admin error to the Review Contractor ‘A’.
2.1	Review Contractor ‘B’ sends the ICDT Response package to the esMD system.
2.2	esMD validates the Response package and sends the acknowledgement back to the Review Contractor ‘B’ if the validation is successful or validation errors in case of failures.
2.3	esMD system delivers the ICDT Response package to the Review Contractor ‘A’.
2.4	The RC sends the successful pickup notification, error pickup notification, or admin error to the esMD system via the MFT server.
2.5	esMD system validates and delivers the Pickup notification, error pickup notification, or admin errors to the Review Contractor ‘B’.

14.2 ADR RRL Business Process Flow

NOTE: **ADR Review Results letter and PA PCR Decision letter functionalities cannot be used or implemented in the way that’s currently implemented in esMD system. esMD will be making the updates to these functionalities in the future release. Please refrain from using these functionalities until the required updates are made in the esMD system.**

This section defines the process flow of the ADR RRL sent from the RC to the HIH via the esMD application. Refer to Figure 11: ADR RRL Business Process Flow and the Table 63: Manual Submission of Administrative Error Response provides the detailed steps.

Figure 11: ADR RRL Business Process Flow

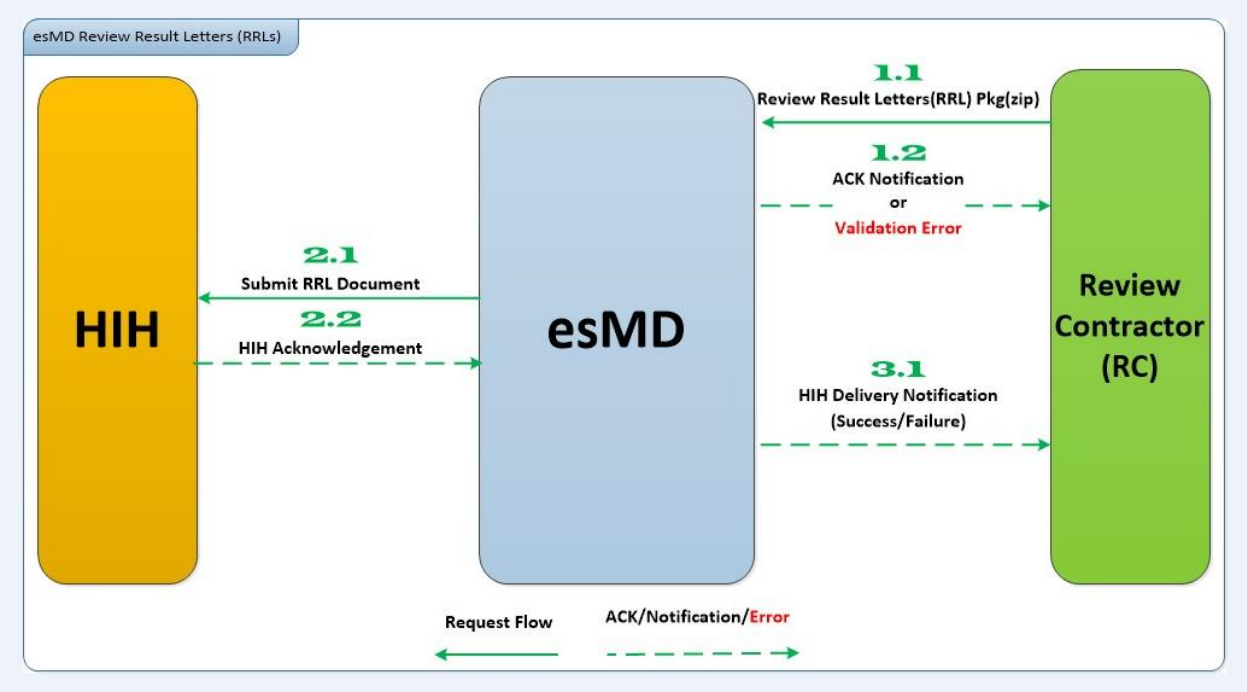


Table 53: ADR RRL Business Process Flow Steps

Message Sequence	Description
1.1	The RC Creates the ADR RRL packages (consisting of RRLs and the esMD Process Metadata XML file) and sends the package to esMD via the TIBCO MFT Server.
1.2	esMD validates the RRL Package received from the RC. The RC receives a success acknowledgement as an XML message from esMD after all validations are passed for the RRLs. The RC receives a validation error as an XML message from esMD for ADR RRLs sent by the RC.
2.1	RRLs are sent as clinical documents to the HIH by esMD after successfully processing the packages sent from the RC.
2.2	esMD receives the HIH Acknowledgement from HIH after HIH processes the RRL documents.
3.1	esMD creates the success or failure Notification based on the response from the HIH on the delivery of RRLs package. The RC Receives the HIH delivery notification from esMD.

14.3 PA/PCR Decision Letters Business Process Flow

NOTE: ADR Review Results letter and PA PCR Decision letter functionalities cannot be used or implemented in the way that's currently implemented in esMD system. esMD will be making the updates to these functionalities in the future

release. Please refrain from using these functionalities until the required updates are made in the esMD system.

This section provides the process flows of the PA/PCR decision letters sent from the RCs to the HIHs via the esMD application. Figure 12: esMD PA/PCR Decision Letter Flow Diagram shows the process flow for PA/PCR Decision Letters.

Figure 12: esMD PA/PCR Decision Letter Flow Diagram

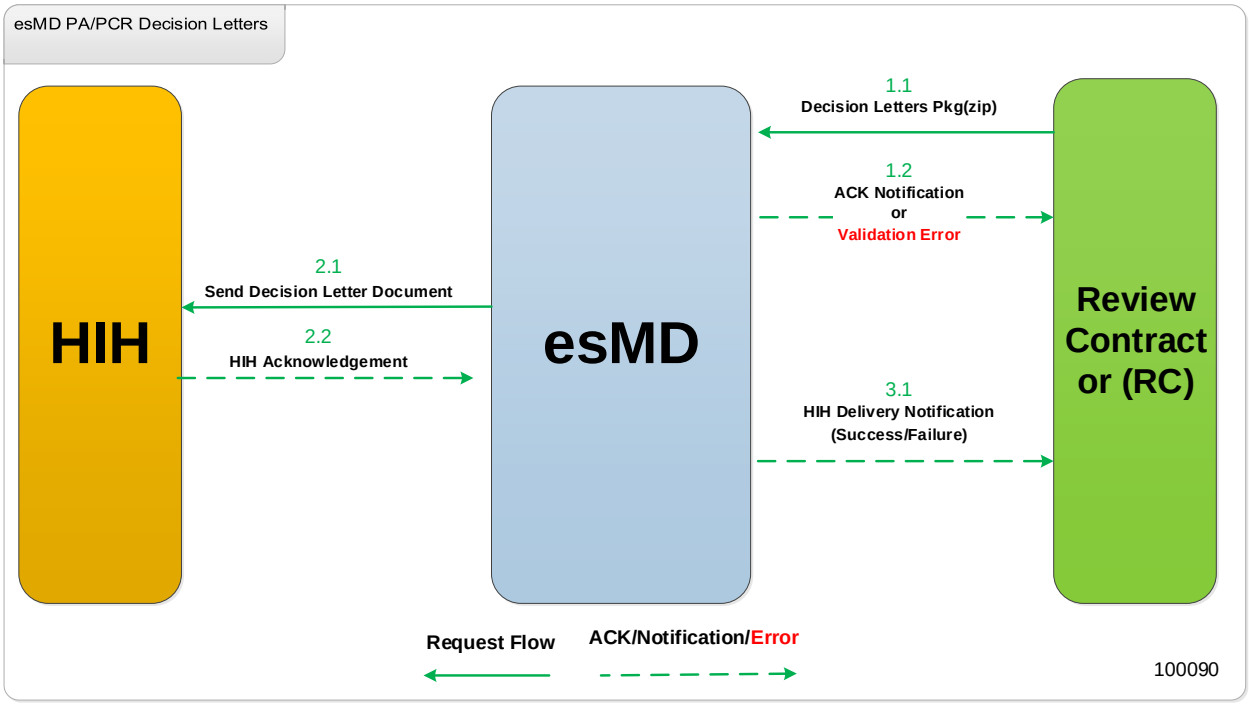


Table 54: PA/PCR Decision Letter Processing describes the typical PA/PCR decision letter interaction as shown in the Figure 12: esMD PA/PCR Decision Letter Flow Diagram.

Table 54: PA/PCR Decision Letter Processing

Message Sequence	Description
1.1	RCs send PA/PCR decision letters (PDF format) in a zip request package to esMD.
1.2	The esMD system processes the PA/PCR decision letter zip package received from the RC and generates the appropriate acceptance or rejection response acknowledgement to the RC.
2.1	The esMD system constructs the XDR request payload with the RC review decision letters embedded in unstructured HL7 clinical document standard and send it to the HIH.
2.2	The HIH acknowledges the acceptance status of the document/request received from esMD.

Message Sequence	Description
3.1	The esMD sends the PA/PCR documentation HIH delivery notifications to the RC.

14.4 Service Registration Processing Overview

Table 55: Service Registration Flow Steps describes the typical Service Registration flow interaction as shown in Figure 13: Service Registration Process Flow.

Figure 13: Service Registration Process Flow

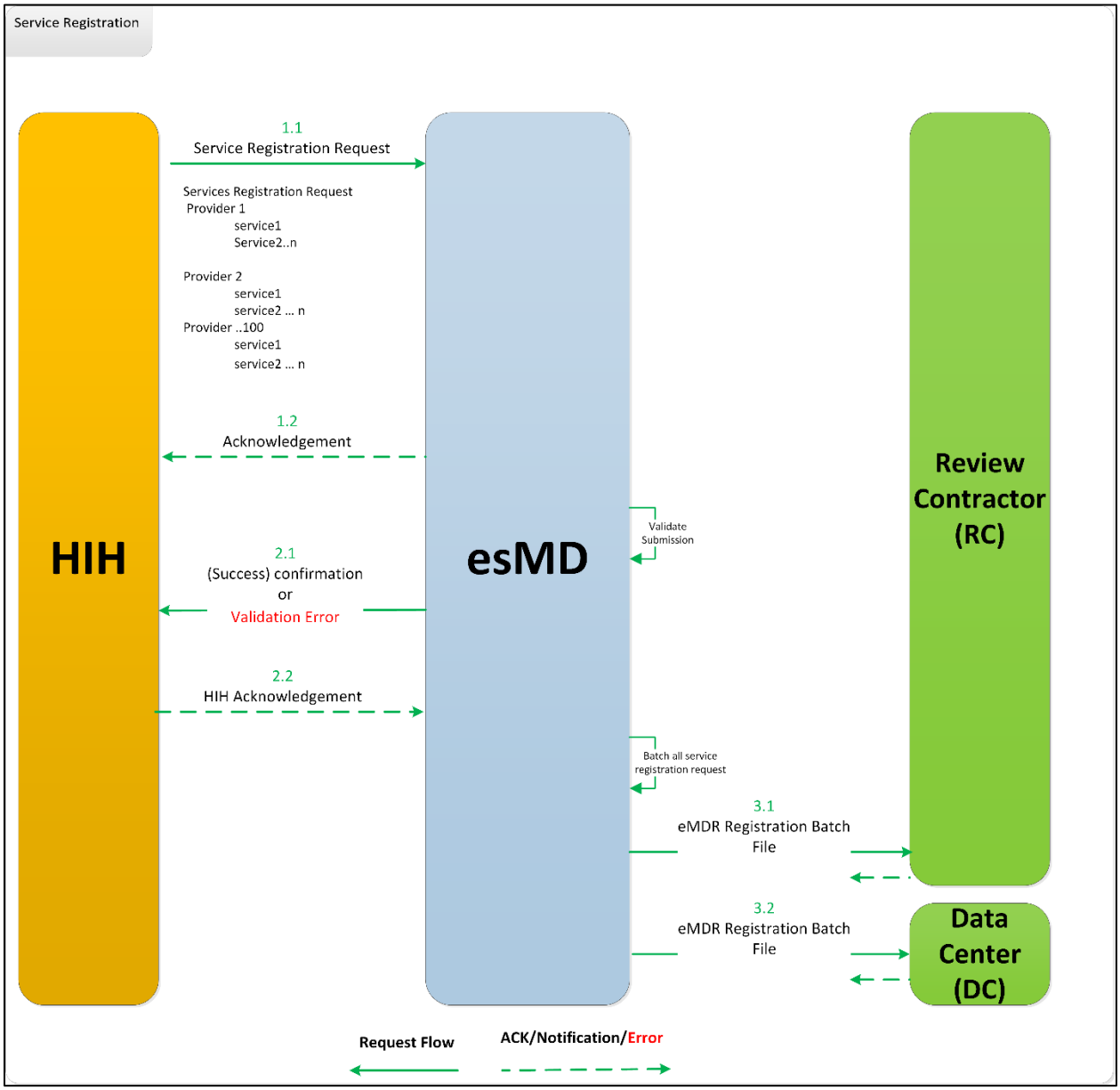


Table 55: Service Registration Flow Steps

Message Sequence	Description
1.1	The HIH submits the service registration request in XDR format to esMD with all necessary metadata information and the Service registration XML (consisting of information for one or more Provider(s) and Service(s)) wrapped as clinical information.
1.2	esMD sends the synchronous request acknowledgment to the HIH.
2.1	The esMD system processes the provider information received in the service registration request, and the success confirmation or error(s) are returned for any validation failures as the first notification. esMD sends one of the following notifications (asynchronous) to the HIH after completing processing of the service registration request: 1. esMD - Request Accepted; 2. esMD - Request Accepted with Errors; or 3. esMD - Meta Data Validation and Persistence.
2.2	The HIH acknowledges the acceptance/rejection status of the notification received from esMD.
3.1	esMD batches all the Service Registration requests and sends them to all of the MAC RCs.
3.2	esMD batches all the Service Registration requests and sends them to all of the Data Centers (DC).

14.5 Document Codes Processing Overview

Table 55: Service Registration Flow Steps describes the typical Document Codes flow interaction as shown in Figure 13: Service Registration Process Flow.

The RC Client API shall download/pull the new DCF Flat File from the Managed File Transfer (MFT) server and initially move the file to the Temp Folder. The RcClient shall continue processing with the header, body, and trailer validation. The RcClient shall move the DCF file to input folder send a Success pickup notification on successful validation of the file. The RcClient shall push the error pickup notification to esMD, if there is any validation failure and delete the downloaded DCF Flat file from the downloaded folder and processing ends.

The steps performed in the schema validation include:

- Length of the flat file lines should be within the limits as mentioned in data element documents in section 15.1 (DCF Flat File Format)
- Number of document code flat file lines present in flat file should be equal to number mentioned in same flat file trailer
- Name of the DCF file should contain proper content type code of 17.
- Header and Trailer should start with pre-defined character

15. Schema Definition and Sample Files

15.1 DCF File Format

Figure 14: Document code file (DCF) Flow

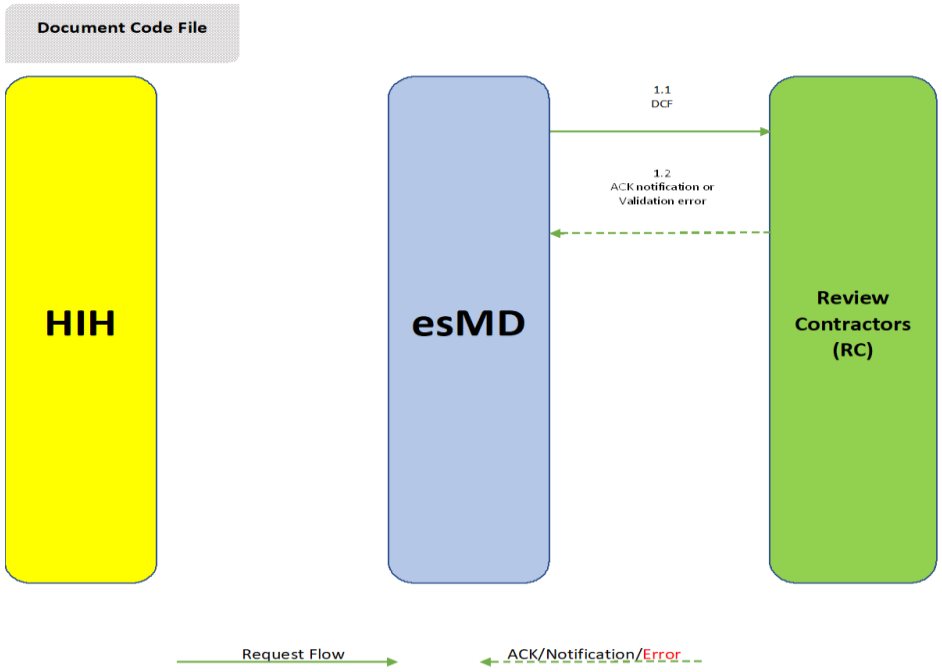


Table 56: Document Code File Process Flow Steps

Message Sequence	Description
1.1	The RC shall receive the DCF flat file from the esMD system via MFT server
1.2	The RC system processes the DCF flat file received from the esMD and generates the appropriate acceptance or rejection response acknowledgement to the esMD

15.2 eMDR (Pre-Pay/Post-Pay) Processing Overview

This section focuses on exchanging structured (Extensible Markup Language (XML)) and unstructured (Portable Document Format (PDF)) eMDR and ADR (Pre-Pay, Post-Pay) transactions in the form of electronic clinical documents and Nationwide Health Information Network (NwHIN)-Cross-Enterprise Document Reliable Interchange (XDR) profile standards, which may already exist in both the initiator and consumer entity systems or may need to be created for this exchange.

15.2.1 Logical Process Flow

15.2.1.1 eMDR Pre-Pay Logical Flow

The eMDR pre-pay logical flow depicts the series of events and sequence of interactions between esMD and Health Information Handlers (HIH) via the XDR interface. The order and timing of the exchange of messages with HIHs are driven by RC submissions. The Content Type Code for this program will be 1.5.

Figure 15: eMDR Pre-Pay Process Flow depicts the logical processing of the eMDR (Pre-Pay) process, and Table 57: eMDR Pre-Pay Logical Process Flow Steps details the eMDR process.

Figure 15: eMDR Pre-Pay Process Flow

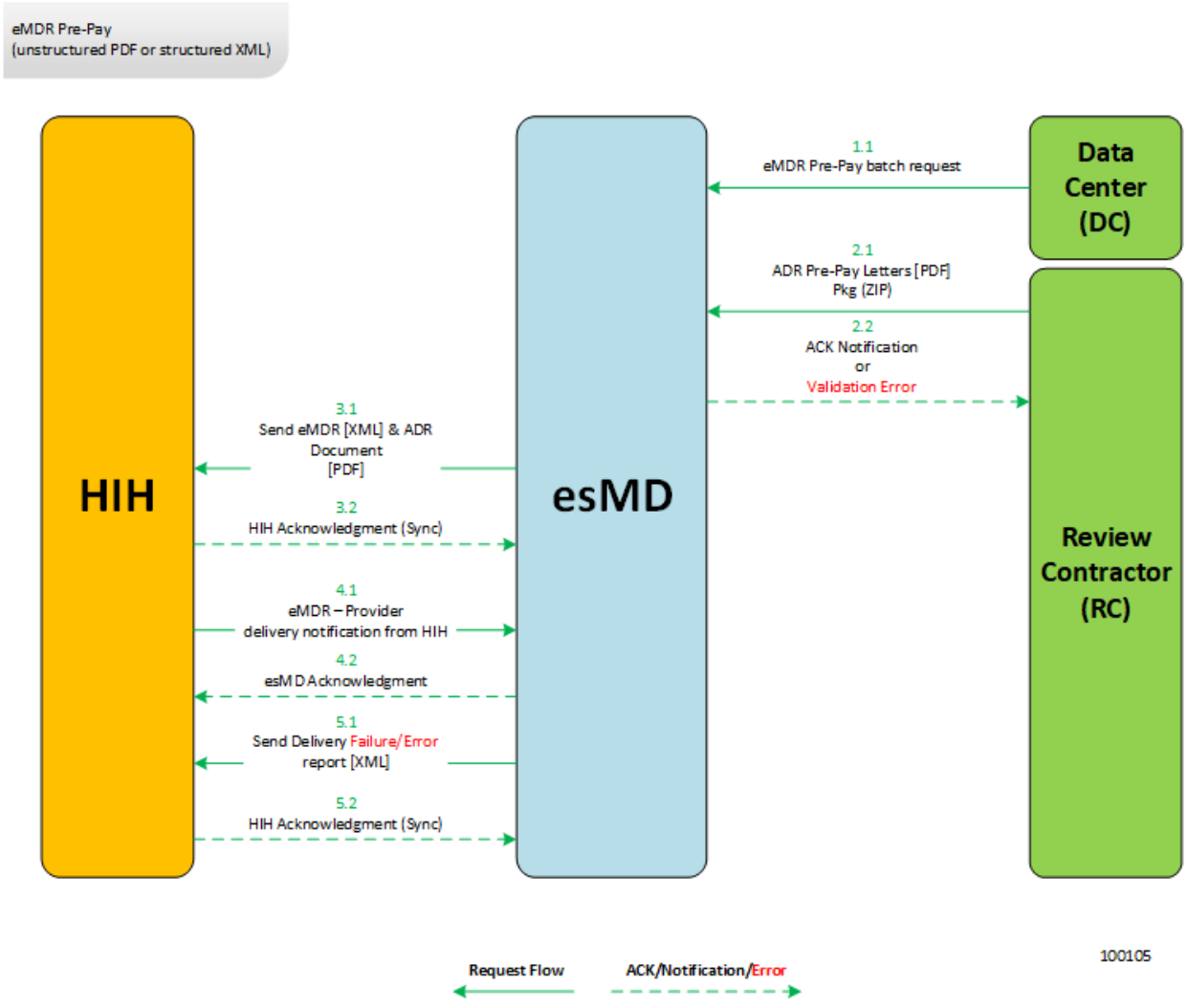


Table 57: eMDR Pre-Pay Logical Process Flow Steps

Message Sequence	Description
1.1	The esMD system receives the eMDR Pre-Pay batch request file from the Data Center (DC). The esMD system processes the eMDR (Pre-Pay) batch request file and holds the eMDR requests within the esMD system until the matching ADR letter (PDF) is received from the RC. The esMD system maintains the record of any processing errors or failures.
2.1	The RC sends ADR letters (PDF) matching the eMDR requests in the zip request package to esMD.
2.2	The esMD system processes the ADR letter zip packages received from the RC and generates the appropriate acceptance or rejection response acknowledgement to the RC.
3.1	The esMD system constructs the XDR request payload with the RC's ADR PDF letter and structured matching eMDR embedded in the unstructured HL7 clinical document standard and sends it to the HIH.
3.2	The HIH acknowledges the acceptance/failure with any of the following statuses for the document/request received from esMD: 1. RequestAccepted; 2. ResponseAccepted; 3. Success; or 4. Error.
4.1	The HIH sends the package delivery confirmation to esMD after the ADR PDF letter and eMDR structured XML are successfully transmitted to the Provider.
4.2	esMD acknowledges the delivery confirmation received from the HIH.
5.1	The esMD system sends the transaction details only when HIH delivery failed due to validation error or transmission error.
5.2	The HIH acknowledges the acceptance/failure with any of the following statuses for the document/request received from esMD: 1. RequestAccepted; 2. ResponseAccepted; 3. Success; or 4. Error.

15.2.2 eMDR Post-Pay Logical Flow

The eMDR post-pay logical flow depicts the series of events and sequence of interactions between esMD and the HIH via the XDR interface. The order and timing of the exchange of messages with HIHs are driven by RC submissions. The Content type code for this program will be 1.6.

Figure 16: eMDR Post-Pay Process Flow depicts the logical processing of eMDR (Post-Pay) process, and Table 58: eMDR Post-Pay Logical Process Flow Steps details the sequence of interactions between esMD and HIH.

Figure 16: eMDR Post-Pay Process Flow

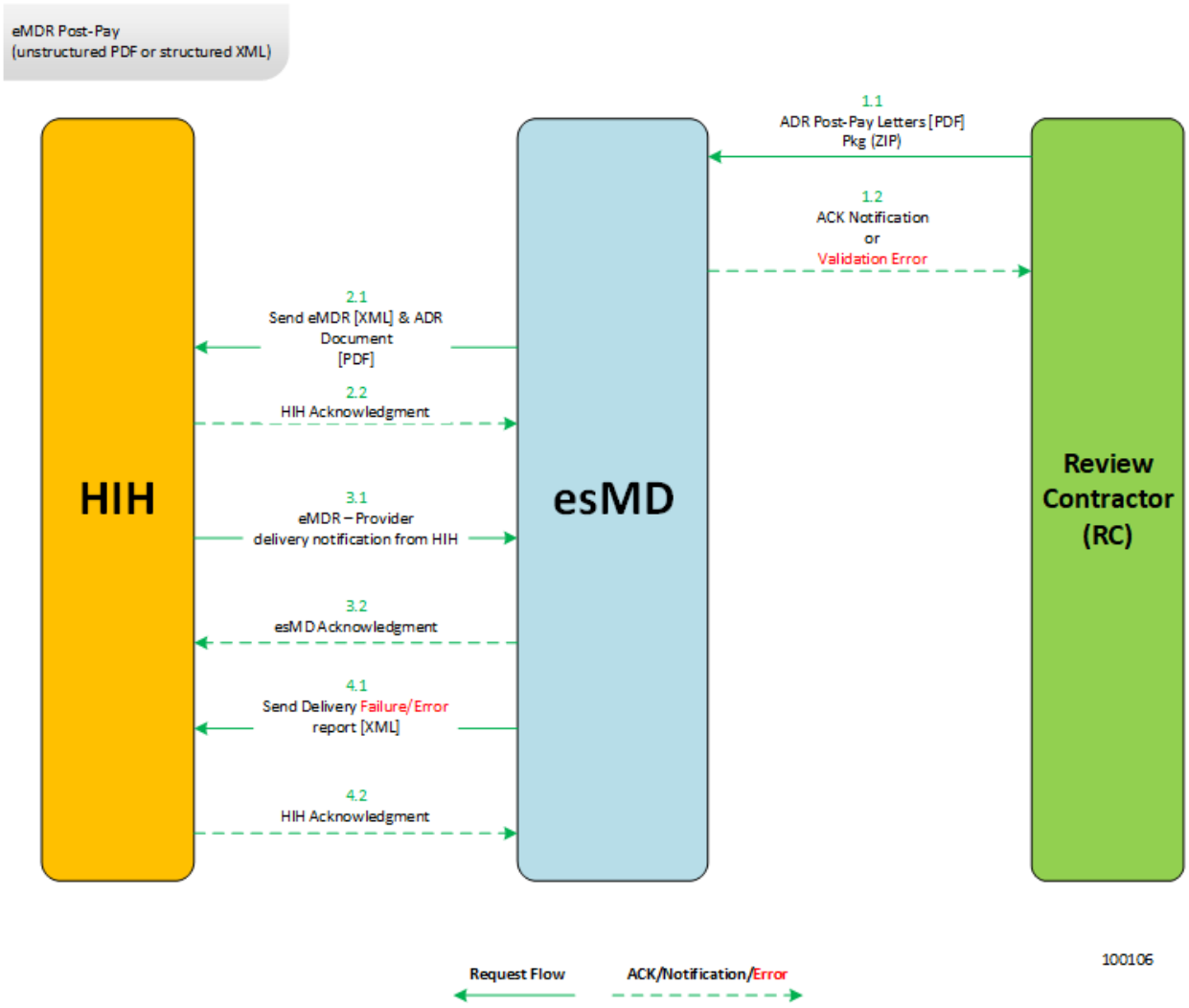


Table 58: eMDR Post-Pay Logical Process Flow Steps

Message Sequence	Description
1.1	The esMD system receives the ADR Letter PDF and structured eMDR Post-Pay XML in a zip file from the RC.
1.2	The esMD system processes the eMDR Post-Pay package received from the RC and generates the appropriate acceptance or rejection response acknowledgement to the RC.
2.1	The esMD system constructs the XDR request payload with the ADR PDF letter and structured Pre-Pay eMDR (XML) embedded in the unstructured HL7 clinical document standard and sends it to the HIH.

Message Sequence	Description
2.2	The HIH acknowledges the acceptance/failure with any of the following statuses for the document/request received from esMD: 1. RequestAccepted; 2. ResponseAccepted; 3. Success; or 4. Error.
3.1	The HIH sends the package delivery confirmation to esMD after the ADR PDF letter and eMDR structured XML are successfully transmitted to the Provider.
3.2	The esMD system acknowledges the delivery confirmation received from the HIH.
4.1	The esMD system sends the transaction details only when HIH delivery failed due to transmission error.
4.2	The HIH acknowledges the acceptance/failure with any of the following statuses for the document/request received from esMD: 1. RequestAccepted; 2. ResponseAccepted; 3. Success; or 4. Error.

15.3 Start RC Client

The RC Client starts on the RC machine or server. It loads the XML Configuration File.

15.3.1 Login and Encryption

The RC Client prompts the user for the following details:

- IDM User ID; and
- IDM Password.

After successful login, TIBCO login credentials are encrypted in memory and used when needed to log in to the TIBCO MFT server. The RC Client initiates two threads, one for the inbound process and one for the outbound process. These processes are described on in sections 15.4 Outbound Process and 15.5 Inbound Processes, respectively.

15.4 Outbound Process

15.4.1 Outbound Start

The RC Client loads configuration parameters for the outbound process from the XML configuration file. The configuration parameters are as follows:

1. Directories used by the RC Client to create the outbound files (outputDirectory);
2. The remote outbound directory to push the files to (remoteOutboundDir);
3. Push frequency (pushFrequency);
4. The outbound file name prefix for the TIBCO MFT server (outboundFilePrefix); and

5. SFTP server details for the chosen environment (ESMDSFTPServer).

15.4.2 Get Outbound Documents

The RC Client checks the output directory for any files to be sent to the HIH. If any such files exist, the process continues to Step D (Connect); otherwise, the outbound process thread sleeps for the time interval determined by the pushFrequency parameter in the XML Configuration file.

15.4.3 Connect

The RC Client connects to the TIBCO MFT server using IDM login credentials. The Encryption utility decrypts the credentials in memory and logs in to the TIBCO MFT server. If the user password is expired, the connection fails, prompting the user to provide the login information again.

15.4.4 Push

The RC Client pushes outbound files to the TIBCO MFT server. After that, the outbound process thread sleeps. The sleep time interval is determined by the outbound push frequency configuration parameter in the XML Configuration file.

15.5 Inbound Processes

15.5.1 Inbound Start

The RC Client loads Configuration parameters from the XML Configuration file. The configuration parameters are for the following inbound processes:

- Pull frequency; and
- SFTP server details for the chosen environment.

15.5.2 Housekeeping

The Housekeeping Manager is responsible for the cleanup and recovery from any abnormal terminations. If the extraction process was interrupted during extraction in the previous run, then there will be compressed files in the local “temp” directory.

15.5.3 Extraction

The Housekeeping Manager extracts compressed files found in the local “temp” directory for the RC Client before it pulls any new documents from the TIBCO MFT server. It will extract the oldest files first. If the extraction is successful, RC Client proceeds to “checksum verification”; otherwise, RC Client creates an error pickup notification.

15.5.4 Checksum Verification

After the extraction is complete, the RC Client uses the XML Processor to parse the metadata file from the zip package. This metadata file contains the checksums for all

payloads in the package. The RC Client verifies the checksum for each file in the package against the checksum in the metadata file. If the checksum is valid for all files, the RC Client will create a pickup notification; otherwise, the RC Client will create an error pickup notification.

15.6 Acknowledgements

15.6.1 Pickup Notification

If the RC Client successfully extracts and verifies compressed files, the RC Client sends a SUCCESS notification through esMD to inform the HIH that the document has been received and successfully processed.

To generate this SUCCESS notification, the RC Client should:

1. Get the Transaction ID from the metadata XML;
2. Prepare the notification with a SUCCESS message and generate an XML notification file; and
3. Rename the XML notification file to the EFT naming standard and move it to the outbound directory. Refer to Section 15.4 Outbound Process for more information.

15.6.2 Error Pickup Notification

If the RC Client encounters an error indicating a failure, while either extracting the compressed file or verifying the checksum for the contents of the package, the RC Client sends an error notification through the esMD system, asking the HIH to resubmit the package. To generate this error notification, the RC Client must:

1. Obtain the compressed file name;
2. Prepare the notification with an Error message;
3. Generate an XML notification file; and
4. Rename the XML notification file to the EFT naming standard and move it to the outbound directory. This file will be handled by the outbound process.

15.7 Connect

After the Housekeeping Manager completes preprocessing, the RC Client checks for an active connection to the TIBCO MFT server. If a connection is active, the RC Client uses this connection. If the connection is inactive, the RC Client uses the Encryption utility to decrypt the login credentials from memory and connects to the TIBCO MFT server.

15.8 Get Notifications

The RC Client uses the SFTP Client to get a list of the available inbound documents for the RC on the TIBCO MFT server.

15.9 Process Document

If any documents are available for the RC Client to pull from the TIBCO MFT server, the RC Client will go through the list to pull each document.

15.10 Pull Document

The RC Client uses the SFTP Client to pull each inbound document from the TIBCO MFT server. The RC Client then extracts the contents of the zip file and continues processing.

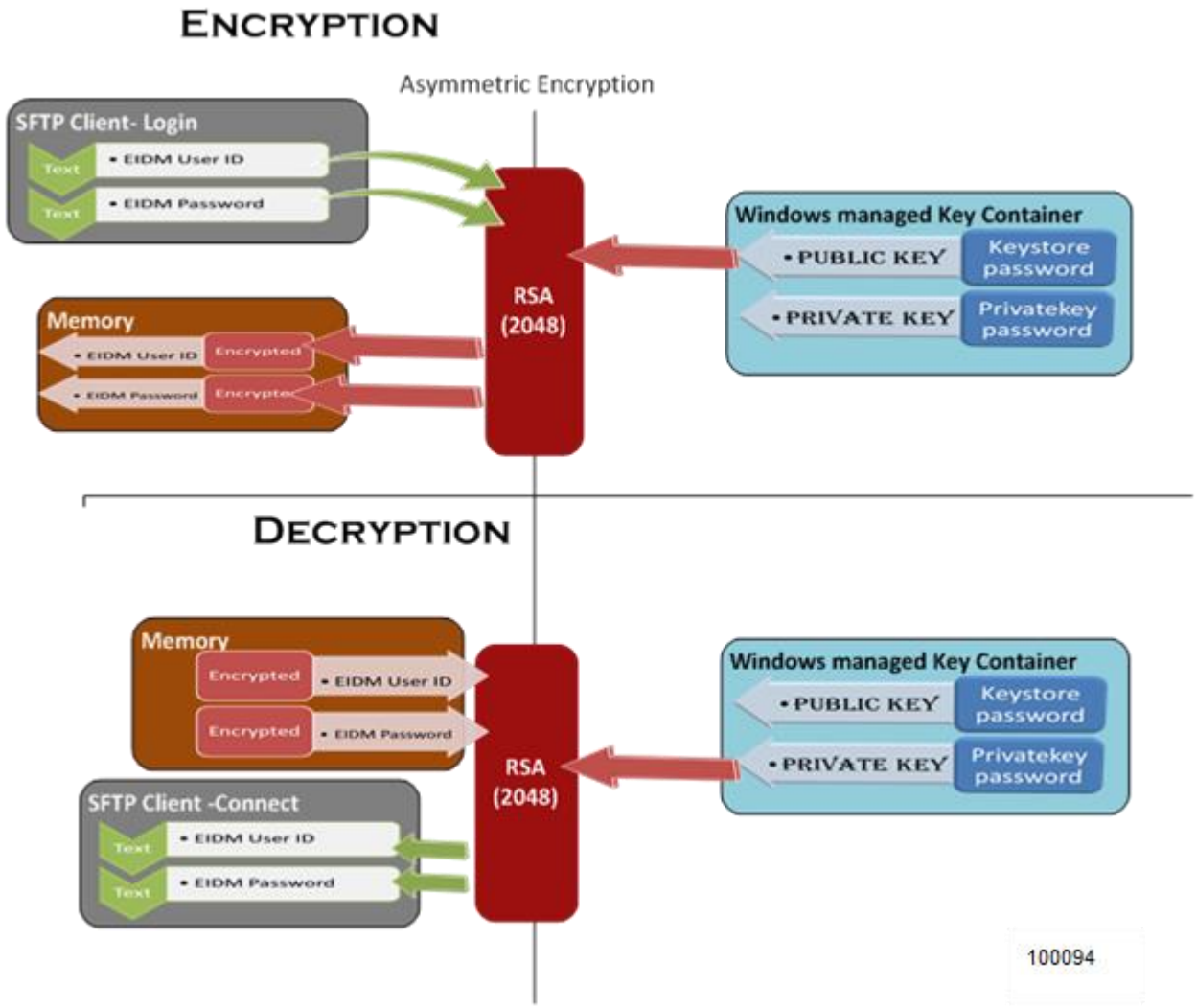
16. Java Client API

16.1 Security

When the RC Client starts, the user credentials are provided because they are stored in encrypted form in memory. Figure 17: Encryption and Decryption Process shows the processes used to safeguard the IDM user credentials from exposure.

The RC Java Client uses RSA asymmetric encryption algorithms to secure the login credentials.

Figure 17: Encryption and Decryption Process



16.2 Java API Documentation

This section discusses API methods that can be called for a custom solution to interface with the TIBCO MFT server. If you, as the RC, choose to use the RC Java client out-of-the-box, skip this section.

16.2.1 Login

Table 59: Login Details lists the methods and their descriptions used in the login process.

Table 59: Login Details

No.	Method	Description
1.	public LoginDetails loginAndEncrypt(SftpDetails _, ESMDConfig.KeyStoreInfo keyStoreInfo_, LoginBean loginBean_) throws Exception;	Logs into the server and stores the encrypted login information. Parameters: 1. sftpDetails_ – The SFTP server Details; 2. keyStoreInfo_ – The Keystore Details; and 3. loginBean_ – The Login Details (User ID and Password). Returns: The LoginDetails Object with the following properties populated: 1. encryptedUID – Encrypted User ID; 2. encryptedPWD – Encrypted Password; and 3. message – status(TRUE/FALSE) and description if any exceptions occurred.
2.	public LoginDetails decryptAndLogin(LoginDetails_, SftpDetails_, ESMDConfig.KeyStoreInfo keyStoreInfo_) throws Exception;	Decrypts the login credentials passed in the LoginDetails object and logs into the TIBCO MFT server. Parameters: 1. loginDetails_ – the LoginDetails object with the following properties populated: • encryptedUID – Encrypted User ID; and • encryptedPWD – Encrypted Password; 2. sftpDetails_ – The SFTP server Details; and 3. keyStoreInfo_ – The Keystore Details. Returns: The LoginDetails Object with the following properties populated: 1. encryptedUID – Encrypted User ID; 2. encryptedPWD – Encrypted Password; and 3. channelSftp – SFTP Channel connected.

16.2.2 Inbound

Table 60: Inbound Method Details lists the methods and their descriptions used in the inbound process.

Table 60: Inbound Method Details

No.	Method	Description
1.	public List<String> getNotifications(LoginDetails loginDetails_,String remoteDownloadDirectoryPath_, String filePattern_) throws SftpException ;	Uses the LoginDetails object to list the remote directory. Parameters: 1. loginDetails_ – the LoginDetails object with the following properties populated: • encryptedUID – Encrypted User ID; • encryptedPWD – Encrypted Password; and • channelSftp – SFTP Channel connected; 2. remoteDownloadDirectoryPath_– The remote directory path to download from as a String; and 3. filePattern_ – The File Name Pattern to look for as a String. Returns: The List<String> with the filenames to pull.
2.	public void pullDocument(String remoteDocumentName_, String localDocumentName_, LoginDetails loginDetails_) throws Exception;	1. Pulls the document (namely, the zip file) from the TIBCO MFT server with the name remoteDocumentName; and 2. Saves it as localDocumentName_ using the loginDetails _ to pull the file from TIBCO MFT server. Parameters: 1. remoteDocumentName_ – The remote file to pull as a String; 2. localDocumentName_ – The local file name to save as a String; and 3. loginDetails_ - the LoginDetails object with the following properties populated: • encryptedUID – Encrypted User ID; • encryptedPWD – Encrypted Password; and • channelSftp – SFTP Channel connected.
3.	public String extractDocument(File localDocumentName_, File localTargetDirectory_) throws Exception;	Extracts the zip file downloaded from the TIBCO MFT server. Parameters: 1. localDocumentName_ - The local zip file to extract; and 2. localTargetDirectory _ - The target directory to place the extracted contents. Returns: The extracted Directory name as a String.

No.	Method	Description
4.	public boolean processMedicalDocumentation(String remoteDocumentName_);	This method does the following: 1. Extracts the zip file into the “download” directory using the extractDocument() method; 2. If extraction fails, calls the acknowledge method with an error event; 3. After successful extraction, verifies the extracted payloads against the checksum in the metadata file using the checkPayloads() method; 4. If checksum fails, calls the acknowledge method with an error event; and 5. If checksum passes, calls the acknowledge() method with a success event. Parameter: 1. localDocumentPath_ - The local document name to process. Returns: The Boolean status of the processing for that document.
5.	public String acknowledge(RCPickupNotification rcPickupNotification_) throws Exception;	Generates the pickup notification for a downloaded document. If the ErrorInfo object is populated, it generates an error pickup notification. If the ErrorInfo object is null, it generates a pickup notification. Parameter: 1. rcPickupNotification_ - The RCPickupNotification object. Returns: The TIBCO MFT server ready compressed file name created in the output directory as a String.
6.	public boolean checkPayloads(File localExtractedDirectory_, RetrieveMedicalDocumentationResponse retrieveMedicalDocumentationResponse_);	Checks the payload against the metadata from the package. Parameters: 1. localExtractedDirectory_ – The directory in which the payloads, were extracted to as a File; and 2. retrieveMedicalDocumentationResponse_ – The metadata xml as object. Returns: The status of the checksum verification.

16.2.3 Outbound

Table 61: Retrieval of Outbound Documents Details provides the retrieval of Outbound documents.

Table 61: Retrieval of Outbound Documents Details

No.	Method	Description
1.	public List<String> getOutboundDocuments(String localOutputDirectoryPath_, String localOutboundDocumentNamePattern_) throws Exception;	This method is used to retrieve the list of outbound documents in the "output" directory to push. Parameters: 1. localOutputDirectoryPath_ – The local "output" directory to push files from as a String; and 2. localOutboundDocumentNamePattern_ – The file name pattern to push as a String. Returns: The List<String> with the names of the Outbound files in the "output" directory.
2.	public void pushDocument(String localOutboundDocumentPath_, String remoteOutboundDirectoryName_, LoginDetails loginDetails_) throws Exception;	This method is used to push a local compressed document from the "output" directory to the TIBCO MFT server. Parameters: 1. localOutboundDocumentPath_ – The name of the file to push as a String; 2. remoteOutboundDirectoryName_ – The remote directory name to push to as a String; and 3. loginDetails_ - The LoginDetails object with the following properties populated: • encryptedUID – Encrypted User ID; • encryptedPWD – Encrypted Password; and • channelSftp – SFTP Channel connected.

16.2.4 PA Error (Rejected Decision) Response

Table 62: Manual Submission of PA and HHPCR Error (Rejected Decision) Response details the methods to submit the PA and HHPCR Error (Rejected Decision) Response.

Table 62: Manual Submission of PA and HHPCR Error (Rejected Decision) Response

No.	Methods	Description
1.	public Message validationOfPAErrorResponse (PAErrorResponseBean paErrorResponseBean_);	This method takes PAErrorResponseBean object as input which has the review error (rejected decision) response information provided by user and validates all that information before generating the response XML. Parameter: 1. paErrorResponseBean_ – The PAErrorResponseBean object to use. Returns: The Message Object which has status of validations result and also the list of Validation Failure Bean object if there is any validation failure with the data provided by the user.

No.	Methods	Description
2.	public SubmitPADeterminationResponse createPAErrorResponseObject(PAErrorResponseBean paErrorResponseBean_) throws Exception;	This method takes PAErrorResponseBean object as input which has the review error (rejected decision) response information provided by user and creates the SubmitPADeterminationResponse object. Parameter: 1. paErrorResponseBean_ – The PAErrorResponseBean object to use. Returns: The SubmitPADeterminationResponse object populated with the data provided by the user.
3.	public String createCompressedTIBCOFileForPAErrorResponse(SubmitPADeterminationResponse submitPADeterminationResponse_) throws Exception;	This method is used create the XML file and compress it into a TIBCO MFT server file. Parameter: 1. submitPADeterminationResponse_ – The SubmitPADeterminationResponse object to use. Returns: The compressed outbound file name ready to be pushed by the outbound process.

16.2.5 Administrative Error Response to Inbound Submissions

Table 63: Manual Submission of Administrative Error Response details the methods to submit the Administrative Error Response to an Inbound submission.

Table 63: Manual Submission of Administrative Error Response

No.	Methods	Description
1.	public Message validationOfAdministrativeErrorResponse (AdministrativeErrorResponseBean administrativeErrorResponseBean_);	This method takes AdministrativeErrorResponseBean object as input which has the administrative error response information provided by user and validates all that information before generating the response XML. Parameter: 1. administrativeErrorResponseBean_ – The AdministrativeErrorResponseBean object to use. Returns: The Message Object which has status of validations result and the list of Validation Failure Bean object if there is any validation failure with the data provided by the user.
2.	private SubmitAdministrativeErrorResponse createAdministrativeErrorResponseObject(AdministrativeErrorResponseBean administrativeErrorResponseBean_) throws Exception;	This method takes AdministrativeErrorResponseBean object as input which has the administrative error response information provided by user and creates the SubmitAdministrativeErrorResponse object. Parameter: 1. administrativeErrorResponseBean_ – The AdministrativeErrorResponseBean object to use. Returns: The SubmitAdministrativeErrorResponse object populated with the data provided by the user.

No.	Methods	Description
3.	public String createCompressedTIBCOFileForAdministrativeErrorResponse(SubmitAdministrativeErrorResponse submitAdministrativeErrorResponse_) throws Exception;	This method is used create the XML file and compress it into a TIBCO MFT server file. Parameter: 1. submitAdministrativeErrorResponse_ – The SubmitAdministrativeErrorResponse object to use. Returns: The compressed outbound file name ready to be pushed by the outbound process.

16.2.6 Utilities - Encryption

Note: The Java Client release from April 28, 2014 does not include the encryption of login credentials. This section depicts the planned design and is subject to change. This guide will be updated as required when the security implementation is completed.

Table 64: Encryption provides the details on the EMSD.RcClient.Encryption.EncryptionUtil methods.

Table 64: Encryption

No.	Methods	Description
1.	public String encryptKSPassword(String keyStorePassword_) throws Exception;	This method encrypts the Keystore password so it can be stored in the configuration file. Parameter: 1. keyStorePassword_– The password to encrypt as a String. Returns: The Encrypted Keystore Password using “PBEWithMD5AndTripleDES.
2.	public String encryptPKPassword(String privateKeyPassword_) throws Exception;	This method encrypts the Private Key password so it can be stored in the configuration file. Parameter: 1. privateKeyPassword_ – The password to encrypt as a String. Returns: The Encrypted Private Key Password using “PBEWithMD5AndTripleDES”.

No.	Methods	Description
3.	public Map<String, String> encryptCredentials(Map<String, String> loginInfo_, ESMDConfig.KeyStoreInfo keyStoreInfo_) throws Exception;	This method encrypts the IDM login credentials using an RSA Public Key from the JKS Store. Parameters: 1. loginInfo_ - The Map<String, String> containing the Unique Identification Number (UID) and PWD as keys; and 2. keyStoreInfo_ - The ESMDConfig.KeyStoreInfo object with the following details populated: • keyStoreLocation – The JKS Store to use as a String; • encKeyInfo – The Encrypted Keystore password to load the JKS as a String; and • certAlias – The alias of the certificate to retrieve the public key as a String. Returns: The Map<String, String> of encrypted login credentials ENC_UID and ENC_PWD as keys.
4.	public Map<String, String> decryptCredentials(Map<String, String> encryptedLoginInfo_, ESMDConfig.KeyStoreInfo keyStoreInfo_) throws Exception;	This method decrypts the IDM login credentials using an RSA Private Key from the JKS Store. Parameters: 1. encryptedLoginInfo_ - The Map<String, String> of encrypted login credentials ENC_UID and ENC_PWD as keys; and 2. keyStoreInfo_ - The ESMDConfig.KeyStoreInfo object with the following details populated: • keyStoreLocation – The JKS Store to use as a String; • encKeyInfo - The Encrypted Keystore password to load the JKS as a String; • certAlias - The alias of the certificate to retrieve the public key as a String; and • encKeyInfoExt - The Encrypted private key password to load the private key from the JKS Store as a String. Returns: The Map<String, String> containing the UID and PWD as keys.

16.2.7 Test Connection

Refer to Table 65: Remote Troubleshooting for details on the ExecuteHandshake method.

Table 65: Remote Troubleshooting

No.	Methods	Description
1.	public bool executeHandshake()	This sample method invokes a call to the TIBCO MFT server to pass login information to assist in remote troubleshooting. Returns: TRUE if handshake succeeded.

17. API Methods

17.1 Unique ID Rules and Format

The Unique ID is generated based on the following format in the RC Client for the following Pilot Programs. Refer to Table 66: Example Unique ID Rules & Format.

1. ICDT Solicited Request – CTC 15.1;
2. ICDT Solicited Response – CTC 15.2;
3. ICDT Unsolicited Response – CTC 15.3;
4. Review Result Letters – CTC 1.3; and
5. PA/PCR Decision Letters – CTC 1.4.

Table 66: Example Unique ID Rules & Format

ID	Format	Example	Notes
1	L<CTC><3CharRandomID><routingID><date><time>	L13STSESD0020131191203070	1. CTC for RRL -- 13 (period in the CTC is removed in the Unique ID); 2. CTC for PA/PCR decision letters -- 14 (period in the CTC is removed in the Unique ID); 3. CTC for Solicited Request – 151 (period in the CTC is removed in the Unique ID); 4. CTC for Solicited Response – 152 (period in the CTC is removed in the Unique ID); 5. CTC for UnSolicited Response – 153 (period in the CTC is removed in the Unique ID); 6. routingId -- RCs mailbox ID; 7. date -- Date in MMDDYY format; and 8. time - Time in HHMMSS0

17.1.1 Unique ID Generation

Table 67: Unique ID Generation API Methods describes the API methods available to generate the Unique ID.

Table 67: Unique ID Generation API Methods

No.	Class Name	Method	Description
1	ICDTUtils	Public static String randomAlphaNumericValue() throws Exception;	This method is used to generate the 3-character alphanumeric value that is used as input for generating the Unique ID. Parameter: None Returns: The String Object which has the 5-character alphanumeric value.
2	ICDTUtils	public static String generateUniqueID(String randomAlphaNumericValue_, String date_, String timestamp_) throws Exception;	This method is used to generate a Unique ID for each ICDT Request and ICDT Response sent from the RC. Parameter: 1. **randomAlphaNumericValue_** - The value created using the randomAlphaNumericValue() method; 2. **Date_** - The current system date in MMddyy format; and 3. **Timestamp_** - The current system timestamp in HHmmss format. Returns: The String Object of Unique ID value for the ICDT Request and ICDT Response.

17.2 ICDT Request

Table 68: ICDT Request API Methods details the methods available to submit the ICDT Request by different Review Contractors.

Table 68: ICDT Request API Methods

No.	Class Name	Method	Description
1	ICDTRequestProcessorImpl	Message generateICDTPackage(ICDTMetadataBean_, boolean isUniqueIdCreate_)	This method is used to create the Request XML based on the bean object values. **icdtMetadataBean_** - ICDTMetadataBean object values to generate the Request XML file **isUniqueIdCreate_** - Boolean value (true or false) to denote if the Request ID is to be generated by API. True if the Request ID is to be generated by API and false if the RCs provide the Request ID to the API. **Returns Message Object -** The Message Object contains messages, status, list of errors and desc, randomNumber, Request ID, and filename.

No.	Class Name	Method	Description
2	ICDTRequestProcessorImpl	Message generateICDTPackage(String icdtSolicitedRequestXMLFileLocation_, boolean isUniqueIdCreate_, Collection<File> icdtAttachmentFiles_)	This method is used to generate the ICDT Request package based on the absolute path of the Request XML file. **icdtSolicitedRequestXMLFileLocation_** - The absolute file path of the request XML. **isUniqueIdCreate_** - Boolean value (true or false) to denote if the Request ID is to be generated by API. True if the Request ID is to be generated by API and false if the RCs provide the Request ID to the API. **icdtAttachmentFiles_** - List of attachment files to be included in the Solicited response package. **Returns Message Object -** The Message Object contains message, status, list of errors and desc, randomNumber, Request ID, and filename
3	ICDTRequestProcessorImpl	Message generateICDTPackage(File icdtSolicitedRequestXMLFileObj_, boolean isUniqueIdCreate_, Collection<File> icdtAttachmentFiles_)	This method is used to create the Request XML based on the file object **icdtSolicitedRequestXMLFileObj__** - The request XML file object passed by RCs **isUniqueIdCreate_** - Boolean value (true or false) to denote if the Request ID is to be generated by API. True if the Request ID is to be generated by API and false if the RCs provide the Request ID to the API **icdtAttachmentFiles_** - List of attachment files to be included in the request package **Returns Message Object -** The Message Object contains message status list of errors and desc, randomNumber, Request ID, and filename
4	ICDTRequestProcessorImpl	ICDTRequest readRequestXMLFile(String xmlFileNamewithAbsolutePath_)	This method is used to read the ICDT Request XML file received from the esMD system **xmlFileNamewithAbsolutePath_** - The absolute path of the Request XML downloaded in the RC Client

17.3 ICDT Solicited Response

Table 69: ICDT Solicited Response API Methods details the methods available for sending the ICDT Response by RCs.

Table 69: ICDT Solicited Response API Methods

No.	Class Name	Method	Description
1	ICDTSolicitedResponseProcessorImpl	Message generateICDTPackage(ICDTMetadataBean_, boolean isUniqueIdCreate_)	This method is used to generate the Solicited Response package based on the metadata bean object. **icdtMetadataBean**_ - ICDTMetadataBean object values for generating the Solicited Response **isUniqueIdCreate**_ - Boolean value (true or false) to denote if the Response ID is to be generated by API. True if the Response ID is to be generated by API and false if the RCs provide the Request ID to the API **Returns Message Object -** The Message Object contains message, status, list of errors and desc, randomNumber, Response ID, and the filename
2	ICDTSolicitedResponseProcessorImpl	Message generateICDTPackage(String icdtSolicitedResponseXMLFileLocation_, boolean isUniqueIdCreate, Collection<File> icdtAttachmentFiles_)	This method is used to create the Response XML based on the absolute path of the file. **icdtSolicitedResponseXMLFileLocation**_ - The absolute file path of the response XML **isUniqueIdCreate**_ - Boolean value (true or false) to denote if the Response ID is to be generated by API. True if the Response ID is to be generated by API and false if the RCs provide the Request ID to the API **icdtAttachmentFiles**_ - List of attachment files to be included in the Solicited response package **Returns Message Object -** The Message Object contains message, status, list of errors and desc, randomNumber, Request ID, and filename
3	ICDTSolicitedResponseProcessorImpl	Message generateICDTPackage(File icdtSolicitedRequestXMLFileObj_, boolean isUniqueIdCreate_, Collection<File> icdtAttachmentFiles_)	This method is used to create the Response XML based on the file object **icdtSolicitedRequestXMLFileObj**__ - The response XML file object passed by RCs **isUniqueIdCreate**_ - Boolean value (true or false) to denote if the Response ID is to be generated by API. True if the Response ID is to be generated by API and false if the RCs provide the Request ID to the API **icdtAttachmentFiles**_ - List of attachment files to be included in the Solicited response package **Returns Message Object -** The Message Object contains message, status, list of errors and desc, randomNumber, Request ID and filename

No.	Class Name	Method	Description
4	ICDTSolicitedResponseProcessorImpl	ICDTResponse readSolicitedResponseXMLFile (String xmlFileNamewithAbsolutePath_)	This method is used to read the ICDT Solicited Response XML file received from the esMD system **xmlFileNamewithAbsolutePath_** - The absolute path of the Solicited Response XML downloaded in the RC Client

17.4 ICDT Unsolicited Response

Table 70: ICDT UnSolicited Response API Methods details the methods available for sending ICDT Response by RCs.

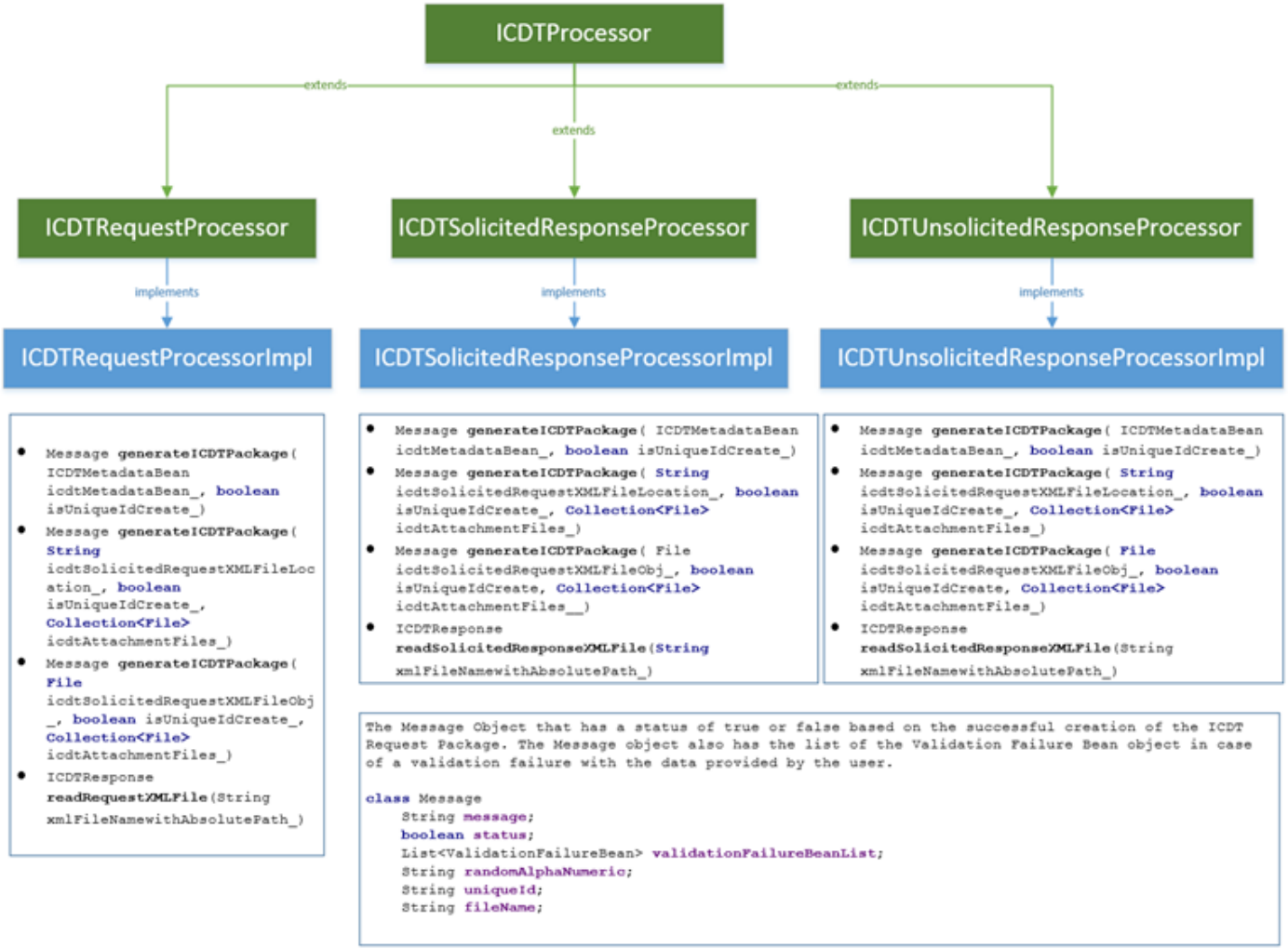
Table 70: ICDT UnSolicited Response API Methods

No.	Class Name	Method	Description
1	ICDTUnsolicitedResponseProcessorImpl	Message generateICDTPackage(ICDTMetadataBean_, boolean isUniqueIdCreate_)	This method is used to generate the Unsolicited Response package based on the metadata bean object. **icdtMetadataBean_** - ICDTMetadataBean object values for generating the Unsolicited Response **isUniqueIdCreate_** - Boolean value (true or false) to denote if the Response ID is to be generated by API. True if the Response ID is to be generated by API and false if the RCs provide the Request ID to the API **Returns Message Object -** The Message Object contains message, status, list of errors and desc, randomNumber, Response ID, and the filename
2	ICDTUnsolicitedResponseProcessorImpl	Message generateICDTPackage(String icdtUnsolicitedResponseXMLFileLocation_, boolean isUniqueIdCreate, Collection<File> icdtAttachmentFiles_)	This method is used to create the Request XML based on the absolute path of the file. **icdtSolicitedResponseXMLFileLocation_** - The absolute file path of the Response XML **isUniqueIdCreate_** - Boolean value (true or false) to denote if the Response ID is to be generated by API. True if the Response ID is to be generated by API and false if the RCs provide the Request ID to the API **icdtAttachmentFiles_** - List of attachment files to be included in the UnSolicited response package **Returns Message Object -** The Message Object contains message, status, list of errors and desc, randomNumber, Request ID, and filename **Note:** RCs must pass dummy values for the file size tags.

No.	Class Name	Method	Description
3	ICDTSolicitedResponseProcessorImpl	Message generateICDTPackage(File icdtUnsolicitedRequestXMLFileObj_, boolean isUniqueIdCreate_, Collection<File> icdtAttachmentFiles_)	This method is used to create the Request XML based on the file object **icdtUnsolicitedRequestXMLFileObj__** - The Response XML file object passed by RCs **isUniqueIdCreate_** - Boolean value (true or false) to denote if the Response ID is to be generated by API. True if the Response ID is to be generated by API and false if the RCs provide the Request ID to the API **icdtAttachmentFiles_** - List of attachment files to be included in the UnSolicited response package **Returns Message Object -** The Message Object contains message, status, list of errors and desc, randomNumber, Request ID, and filename **Note:** RCs must pass dummy values for the file size tags.
4	ICDTSolicitedResponseProcessorImpl	ICDTResponse readUnsolicitedResponseXMLFile (String xmlFileNamewithAbsolutePath_)	This method is used to read the ICDT Solicited Response XML file received from the esMD system **xmlFileNamewithAbsolutePath_** - The absolute path of the Unsolicited Response XML downloaded in the RC Client

Figure 18: High-level ICDT API Architecture identifies the classes and method signatures for ICDT Request/Solicited Response and Unsolicited Response.

Figure 18: High-level ICDT API Architecture



17.5 Administrative Error Response

There are two additional Administrative error responses for supporting ICDT Request/Solicited Response and Unsolicited response:

1. The file is corrupt and/or cannot be read; and
2. A virus was found.

In order to generate Administrative error response for ICDT functionality, the API methods are provided with different method signatures as shown in Table 71: Administrative Error Response API Methods.

Table 71: Administrative Error Response API Methods

No.	Class/Interface Name	Methods	Description
1	ICDTAdminErrorNotificationProcessImpl	public Message generateICDTNotification(NotificationBean adminErrorBean_) throws Exception	This method is used to generate the administrative error response based on the bean object **adminErrorBean_** - AdminErrorBean object that holds the administrative error response details
2	ICDTAdminErrorNotificationProcessImpl	public ICDTCommunication readICDTNotification(String fileNameWithAbsolutePath_)throws Exception	This method is used to read the administrative error response based on the absolute path of the administrative error response file. **fileNameWithAbsolutePath_** - The String object that has the absolute path of the administrative error response file.

17.6 ADR RRL

Table 72: ADR Review Results Letter API Methods lists the API methods for generating the ADR Review Results package.

Table 72: ADR Review Results Letter API Methods

No.	Class/Interface Name	Methods	Description
1	ESMDManualSubmitADRReviewResultLetterImpl	Message generateADRReviewResultsPackage(Object eSMDProcessMetadataBean, Collection<File> adrLetterFiles)	This method generated the RRL package based on the bean object for the esMD Process Metadata file and the attachment files for the ADR RRL files. Parameters: 1. **eSMDProcessMetadataBean –** Process Metadata bean object for generating the ADR RRL package; 2. **adrLetterFiles –** Collection of the ADR RRL files to be included in the package; and 3. **Returns Message Object -** The Message Object contains message, status, list of errors and description, randomNumber, Response ID, and the filename.
2	ESMDManualSubmitADRReviewResultLetterImpl	String generateUniqueID(String randomAlphaNumericValue_, String date_, String timestamp_)	This method is used to generate a Unique ID for each ADR RRL sent from the RC. Parameters: 1. **randomAlphaNumericValue_** - The value created using the randomAlphaNumericValue() method; 2. **Date_** - The current system date in MMddyy format; and 3. **Timestamp_** - The current system timestamp in HHmmss format. Returns: The String Object of Unique ID value for the ADR RRL.

17.7 PA/PCR Decision Letters

Table 73: PA/PCR API Methods lists the API methods for generating the PA/PCR Decision letters package.

Table 73: PA/PCR API Methods

No.	Class	Method	Description
1	ESMDManualSubmitDecisionLettersImpl	Message generateDecisionLettersPackage (Object eSMDProcessMetadataBean, Collection<File> decisionLetterFiles)	This method generates the PA/PCR decision letters package based on the bean object for the esMD Process Metadata file and the attachment files for the decision files: 1. **eSMDProcessMetadataBean** – Process Metadata bean object for generating the decision letters package; 2. **decisionLetterFiles** – Collection of the decision letter files to be included in the package; and 3. **Returns Message Object** - The Message Object contains the message, status, list of errors and description, random number, response ID, and the filename.
2	ESMDManualSubmitDecisionLettersImpl	String generateUniqueID(String randomAlphaNumericValue_, String date_, String timestamp_)	This method generates a Unique ID for each PA/PCR decision letter sent from the RC: 1. **randomAlphaNumericValue_** - The value created using the randomAlphaNumericValue() method; 2. **Date_** - The current system date in MMddyy format; 3. **Timestamp_** - The current system timestamp in HHmmss format; and 4. **Returns**: The String Object of Unique ID value for the decision letter files.

17.8 Pre-Pay and Post-Pay eMDR Letters

Table 74: Pre-Pay and Post-Pay API Methods lists the API methods for generating the Pre-Pay and Post-Pay eMDR letter packages.

Table 74: Pre-Pay and Post-Pay API Methods

No.	Class/Interface Name	Methods	Description
1.	ESMDManualSubmitADReMDRPrePayImpl	public Message generateEMDRPrepayPackage(String eMDRPrepayLetterDirectory_)	Used to generate the ADR Pre-Pay letters zip package that contains the PDF file, structured XML file, and eMDR process metadata XML file. Parameters: **eMDRPrepayLetterDirectory_** – The absolute folder path of the ADR letters in PDF format. **Returns**: The Message Object that has a status of True or False based on the successful creation of the ADR letters zip package. The Message Object also has the list of validation failure objects in case of a validation failure, with the data provided by the user.

No.	Class/Interface Name	Methods	Description
2	ESMDManualSubmitADReMDRPostPayImpl	public Message generateEMDRPostpayPackage (Object eMDRPostPayStructuredBean_, Collection<File> eMDRPDFFiles_)	Used to generate the ADR Post-Pay letters zip package that contains the PDF file, structured XML bean, and PDF file. Parameters: 1. **eMDRPostPayStructuredBean_** – The bean object of the structured XML file. 2. **eMDRPDFFiles_** - The PDF file for the ADR letter file. **Returns**: The Message object that has a status of True or False based on the successful creation of the ADR letters zip package. The Message object also has the list of validation failure objects in case of a validation failure, with the data provided by the user.
3	ESMDManualSubmitADReMDRPostPayImpl	public Message generateEMDRPostpayPackage (File eMDRStructuredXMLFile_, Collection<File> eMDRPDFFiles_)	Used to generate the ADR Post-Pay letters zip package that contains the PDF file, structured XML file, and eMDR process metadata XML file. Parameters: 1. **eMDRStructuredXMLFile_** – The absolute file path of the eMDR Structured file in XML format. 2. **eMDRPDFFiles_** - The PDF file for the ADR letter file. **Returns**: The Message bean object that has a status of True or False based on the successful creation of the ADR letters zip package. The Message bean object also has the list of validation failure objects in case of a validation failure, with the data provided by the user.

No.	Class/Interface Name	Methods	Description
4	ESMDManualSubmitADReMDRPostPayImpl	public Message generateEMDRPostpayPackage (string eMDRStructuredXMLFilePath_, Collection<String> eMDRPDFFilesPath_)	Used to generate the ADR Post-Pay letters zip package that contains the PDF file, structured XML file, and eMDR process metadata XML file. Parameters: 1. **eMDRStructuredXMLFilePath_** – The absolute file path of the ADR letters in PDF format. 2. **eMDRPDFFiles_** - The PDF file for the ADR letter file. **Returns**: The Message bean object that has a status of True or False based on the successful creation of the ADR letters zip package. The Message bean object also has the list of validation failure objects in case of a validation failure, with the data provided by the user.

17.9 Logs

Table 75: RC Client Logs lists the logs the RC Client provides. The RC is advised to monitor the logs for errors and exceptions.

Table 75: RC Client Logs

Log	Description
config.log	Logging for the encryptConfig.bat utility.
handshake.log	Logging for the test connection process.
rc.log	Logging for the sample application.
Inbound.log	Logging for the Inbound Process.
outbound.log	Logging for the Outbound Process.
response.log	Logging for the Response File (PA Review and Administrative Error) Creation Process.

Log	Description
request.log	Logging for the Request File (eMDR Request) Creation Process.

17.10 Utilities

Table 76: RC Client Utilities lists the utilities the RC Client provides.

Table 76: RC Client Utilities

Log	Description
encryptConfig.bat	Encrypts the provided passwords and updates the configuration XML file.
rcclient.bat	The RC Client User Interface Application. This application will have: 1) Login; 2) Review Decision Response to PA Request; 3) Error Response to PA Request; 4) Administrative Error Response to Inbound Submissions; and 5) Advanced/Debugging functionalities.

18. Error Codes

18.1 Errors: esMD to RC

Table 77: Error Codes Sent from esMD to RC lists all the error codes sent from the esMD to the RC.

Table 77: Error Codes Sent from esMD to RC

Error Code	Error Description
111	File not formatted correctly
118	ESMD validation error: Error encountered while saving ReviewContractorPickUpStatus data
144	Failure in sending the Administrative error response to HIH
145	Failure in sending the Administrative PA response to HIH
146	Failure in sending the Pickup notification to HIH
222	Number of Records Mismatch
303	esMD validation error: Empty file received in the response.
305	esMD validation error: Review Contractor PickUp Timestamp is not a valid Timestamp. Correct and resubmit.
306	esMD validation error: esMD Delivery Timestamp is not a valid Timestamp. Correct and resubmit.
333	Duplicate File Sent
501	Missing Contractor/Workload number
502	Missing esMD Transaction ID
503	Missing Mode of Receipt
504	Missing Service Trace Number
516	ESMD validation error: Error encountered while storing PA Review Results Response
517	ESMD validation error: Error encountered while fetching PA Review Results Response Notification objects
518	ESMD validation error: Error encountered while updating PA Review Results Response Notification objects to DB
534	Unzip error
535	Checksum error
536	Metadata error
537	Registration Request error
539	esMD internal system error (Unzip failure). Resubmit.
541	esMD validation error: Transaction ID is invalid. Correct and resubmit.
542	ESMD validation error: Outbound Content Type Code does not match Inbound Content Type Code for this transaction ID

Error Code	Error Description
543	ESMD validation error: RC is not authorized to use this Content Type Code
544	esMD validation error: Reason Code is required when Decision Indicator is N or R. Correct and resubmit.
545	esMD validation error: Total number of Reason Codes cannot exceed 25. Reduce the number of Reason Codes and Resubmit.
546	ESMD validation error: Warning: Total number of Denial Codes exceeds 25
547	ESMD validation error: Denial Code Description must be 2 - 256 positions
555	ESMD validation error: Content Type Code does not exist
556	esMD validation error: Decision Indicator must be A, N, M or R. Correct and resubmit.
557	esMD validation error: Review Contractor Unique Tracking Number must be 1 - 50 alphanumeric characters with no special characters. Correct and resubmit.
558	esMD validation error: Reason Code does not exist in the esMD database. Correct and resubmit.
559	ESMD validation error: Denial Code Description is required, if Decision Indicator is N or R and Denial Code is "Other."
560	esMD validation error: Submission is infected with virus. Correct and resubmit.
562	esMD validation error: Unique Tracking Number is required when Decision Indicator is A, N, or M. Correct and resubmit.
563	Error encountered while validating the PA Review Results Response
564	Error occured while storing the RC Response Virsu Scan Result
565	esMD Internal System Error: Unable to process your response. Correct and resubmit.
566	esMD validation error: A required element is either missing, has an invalid element format, or has an invalid length. Correct and resubmit.
567	esMD validation error: A Decision Indicator of 'M' is invalid for PMD PA or DMEPOS response. Provide a valid Decision Indicator and resubmit.
569	esMD validation error: Number of Approved Units, Approved Service Date, and Date Range are not allowed for this response. Correct and resubmit.
572	esMD validation error: Approved Service End Date is less than or equal to Approved Service Start Date. Correct and resubmit.
576	esMD validation error: Number of Approved Units, Approved Service Date, Approved Service Date Range, Industry Code(s) and Reason Code(s) are not allowed for this response. Correct and resubmit.
577	esMD validation error: Unable to parse response XML file. Correct XML and Resubmit.
600	esMD validation error: Duplicate Reason Codes found. Correct and resubmit.
601	Invalid Contractor/Workload Number
601	esMD validation error: Procedure Code in response not equal Procedure Code in request. Correct and resubmit.
602	esMD validation error: Approved Service Date must be greater than or equal to current system date.

Error Code	Error Description
603	esMD validation error: Decision Indicator = R; response is missing at least one combination of Error Category Code: Error Code. Add the combination(s) of Error Category Code: Error Code and Resubmit.
604	esMD validation error: More than 9 Error Codes were reported for a single Error Category Code. Reduce the number of errors for each Error Category Code to 9 and Resubmit.
605	esMD validation error: Decision Indicator = R; Category Code is invalid for the combination of Error Category Code: Error Code. Correct the Error Category Code and resubmit with correct combination(s) of Error Category Code: Error Code.
606	esMD validation error: Decision Indicator = R; invalid Error Code for the combination of Error Category Code: Error Code. Correct the Error Code and Resubmit with correct combination(s) of Error Category Code: Error Code
607	esMD validation error: Invalid Industry Code. Correct and resubmit.
608	esMD validation error: Invalid Reason Code. Correct and resubmit.
609	esMD Virus Scanning service is unavailable. Retry later
610	esMD validation error: Empty File Received in the Response. Correct and resubmit.
611	esMD validation error: Multiple Files Received in the Response. Resubmit with only one file
612	esMD validation error: Approved Service Date or Approved Service Date Range and Approved Unit are not allowed for this response. Correct and resubmit.
613	esMD validation error: Administrative error code is invalid. Correct and resubmit
614	esMD validation error: Approved Service End Date is less than the Current Date. Correct and resubmit
615	esMD validation error: Invalid error in the pickup notification. Correct and resubmit.
616	esMD validation error: Intended Recipient OID is deactivated and cannot accept response. Correct and resubmit.
617	esMD validation error: Mailbox ID in the response does not match with the Mailbox ID that the request was sent.
618	esMD validation error: Intended Recipient OID is deactivated and cannot accept response.
619	esMD validation error: Mailbox ID in the response does not match with the Mailbox ID that the request was sent.
620	esMD validation error: Invalid Review Response Creation Time format
621	esMD validation error: Invalid review Response Submission Time Format
622	esMD validation error: The Decision Indicator is not valid for this response. For a PA Response, it must be A, M or N. For an Error Response, it must be R. Correct and resubmit.
623	esMD validation error: Both Approved Service Date and Approved Service Date range cannot exist in same response. Correct and Resubmit.
624	esMD validation error: Approved Service Start Date cannot be greater than the Approved Service End Date. Correct and resubmit
625	esMD validation error: Reason Code is not allowed for Decision Indicator A. Correct and resubmit.
631	esMD validation error: A Review or Error Response is not allowed for this transaction.

Error Code	Error Description
632	esMD validation error: Total number of Industry Codes cannot exceed 5. Reduce the number of Industry Codes and resubmit.
633	esMD validation error: Either HIH is not active or agreement has expired to receive the response.
634	esMD validation error: Invalid Number of Approved Unit value, The Number of Approved Unit value should be greater than zero, a non-negative whole number.
637	esMD validation error: Outbound response received for the submission that failed for the Inbound
638	OUTBND_EMPTY_PICKUP_FILE_ERROR
640	esMD validation error: Intended recipient OID and Procedure Code is not a valid combination. Correct and resubmit
701	Missing esMD Transaction ID
702	Missing Procedure Code
703	Missing Decision Indicator
704	Missing Subscriber HIC
705	Missing Workload Number
706	Missing Service Trace Number
800	ESMD validation error: Error occurred while storing the Review Contractor Status PickUp
801	Invalid esMD Transaction ID
801	ESMD validation error: Error occurred while validating the Review Contractor PickUp Status Data
802	Invalid Procedure Code
803	Invalid Decision Indicator
804	Invalid Subscriber HIC Number
805	Invalid Workload Number
806	Invalid Service Trace Number
901	Invalid AAA codes
902	Invalid PA Program Reason Code
903	Invalid Review Decision Reason Code
904	esMD Validation Error: The ICDT Request Zip File received from RC is Zero Byte in size. Correct and resubmit.
905	esMD validation error: The Checksum received does not match the Checksum in the zip file. Correct and resubmit.
907	esMD validation error: The combination of Review Contractor OID and the Content type code received in the request from RC is incorrect. Correct and resubmit.
908	esMD validation error: The combination of HIH OID and the Content type code received in the eMDR request from RC is incorrect. Correct and resubmit.
909	esMD validation error: The Content type code received for the eMDR Request is incorrect. Correct and resubmit.
910	esMD Validation Error: The ICDT Request Zip File received from RC exceeded the maximum allowable size. Correct and resubmit.

Error Code	Error Description
911	esMD Validation Error: The Unique ID received in the eMDR Request Metadata XML File already exists in the database. Correct and resubmit.
912	esMD validation error: The name of the document does not match the document ID in the Metadata zip file. Correct and resubmit.
913	esMD validation error: The Size of Document received does not match with the Size of the Document in the zip file. Correct and resubmit.
914	esMD validation error: The Number of documents in the eMDR Request xml does not match with the number of documents in the zip file. Correct and resubmit.
915	esMD validation error: Unable to encode the response
916	esMD Validation Error: The eMDR request Zip file extraction failed. Correct and resubmit.
917	esMD validation error: Unable to parse request XML file. Correct XML and Resubmit.
918	esMD validation error: The Sender OID received from the Review Contrator for the eMDR Request is Invalid. Correct and resubmit.
919	esMD validation error: RC type provided in the metadata is Invalid for the eMDR request. Correct and resubmit.
920	esMD validation error: The name of the document does not match the document ID in the Metadata zip file. Correct and resubmit.
921	esMD validation error: The HIH OID received from the Review Contrator for the eMDR request is Invalid. Correct and resubmit.
922	esMD Validation Error: A Duplicate RC Unique ID received in the ADR Review Result Response XML File already exists. Correct and resubmit.
923	esMD Validation Error: The File received from RC exceeded the maximum allowable size for ADR Review Result Response. Correct and resubmit.
924	esMD Validation Error: The ADR Review Result Response Zip file extraction failed. Correct and resubmit.
925	esMD validation error: Unable to parse response XML file. Correct XML and Resubmit.
926	esMD validation error: The name of the document does not match the document ID in the Metadata zip file. Correct and resubmit.
927	esMD validation error: The Size of Document received does not match with the Size of the Document in the ADR Review Result Response zip file. Correct and resubmit.
928	esMD validation error: The Checksum received does not match the Checksum in the ADR Review Result Response zip file. Correct and resubmit.
929	esMD validation error: The HIH OID received from the Review Contractor for the ADR Review Result Response is Invalid. Correct and resubmit.
930	esMD validation error: The Sender OID received from the Review Contractor for the ADR Review Result Response is Invalid. Correct and resubmit.

Error Code	Error Description
931	esMD validation error: The Number of documents in the ADR Review Result Response does not match with the number of documents in the zip file. Correct and resubmit.
932	esMD validation error: The Content type code received for the ADR Review Result Response is incorrect. Correct and resubmit.
933	esMD validation error: The combination of Review Contractor OID and the Content type code received in the ADR Review Result Response from RC is incorrect. Correct and resubmit.
934	esMD validation error: The combination of HIH OID and the Content type code received in the ADR Review Result Response from RC is incorrect. Correct and resubmit.
935	esMD validation error: The MIME TYPE IS MISSING IN THE EMDR REQUEST Process Metadata. Correct and resubmit.
936	esMD Validation Error: The Document Unique ID received from RC for the eMDR Request exceeds the maximum length. Correct and resubmit.
937	esMD Validation Error: The Document Unique ID received from RC for the ADR Review Result Response exceeds the maximum length. Correct and resubmit.
938	esMD validation error: The MIME type is missing in the ADR Review Result Response Process Metadata. Correct and resubmit.
939	esMD validation error: Unable to parse {0} XML file. Correct XML and Resubmit
940	esMD Validation Error: The {0} received in the {1} XML File already exists in the database. Correct and resubmit
941	esMD validation error: The Receiver OID received from the Review Contractor for the {0} is non-participating. Correct and resubmit
942	esMD validation error: The Sender OID received from the Review Contractor for the {0} is non-participating. Correct and resubmit
943	esMD validation error: The Content type code received for the {0} is incorrect. Correct and resubmit
944	esMD validation error: The combination of Sender OID and the Content type code received in the {0} from RC is incorrect. Correct and resubmit
945	esMD validation error: The combination of Receiver OID and the Content type code received in the {0} from RC is incorrect. Correct and resubmit.
1032	esMD Validation Error: The Claim ID received in the {0} is Invalid. Correct and Resubmit
1034	esMD Validation Error: Missing Claim ID in the {0}. Correct and Resubmit
1035	esMD Validation Error: The Case ID received in the {0} is Invalid. Correct and Resubmit.
1036	esMD Validation Error: The NPI received in the {0} is Invalid. Correct and Resubmit
1037	esMD Validation Error: Missing NPI in the {0}. Correct and Resubmit
1038	esMD Validation Error: The HICN received in the {0} is Invalid. Correct and Resubmit
1039	esMD Validation Error: Missing HICN in the {0}. Correct and Resubmit

Error Code	Error Description
1040	esMD Validation Error: The OCN received in the {0} is Invalid. Correct and Resubmit
1041	esMD Validation Error: Missing OCN in the {0}. Correct and Resubmit
1049	esMD Validation Error: Sender OID and Receiver OID received in the {0} match. Correct and Resubmit
1050	esMD Validation Error: Internal System issue
1051	esMD validation error: The Checksum received does not match the Checksum in the Zip file. Correct and resubmit
1052	esMD validation error: The MIME type is missing in the {0} Metadata. Correct and resubmit.
1053	esMD validation error: The Size of Document received does not match with the Size of the Document in the {0} Zip file. Correct and resubmit
1054	esMD Validation error: The number of documents received does not match the Number of Documents as stated in the {0} zip file. Correct and resubmit
1055	esMD Validation error: The Mime type {0} is invalid. Correct and resubmit
1056	esMD Validation Error: Missing NPI in the ICDT Request. Correct and Resubmit.
1057	esMD Validation error: The name of the document does not match the name of the document received in the {0} in the zip file
1058	esMD Validation error: The Request ID provided in the {0} is either missing or not exist in the esMD database
1059	esMD Validation Error: Invalid Admin Error Code received from the Review Contractor
1063	esMD Validation Error: The ICDT Request Zip File received from RC exceeded the maximum allowable size. Correct and resubmit
1064	esMD Validation Error: The ICDT Request Zip File received from RC is Zero Byte in size. Correct and resubmit
1065	esMD Validation Error: The {0} Zip file extraction failed. Correct and resubmit
1066	esMD Validation Error: The documentation type received in the {0} XML is invalid. Correct and Resubmit.
1068	esMD validation error: The MIME type is missing in the esMD Process Metadata. Correct and resubmit.
1069	esMD Validation Error: The Document Unique ID received from RC for the ADR Review Result Letter exceeds the maximum length. Correct and resubmit.
1070	esMD validation error: The Number of documents in the esMD Process Metadata xml does not match with the number of documents in the zip file. Correct and resubmit.

Note: The dynamic value {0} will be replaced by ICDT Request or ICDT Solicited Response or ICDT Unsolicited Response.

18.2 Errors: RC to esMD

There are two types of Error Codes sent by the RC to the esMD. They are:

1. Administrative Errors; and

2. Pickup Errors.

18.2.1 Administrative Errors

Table 78: Administrative Error Codes lists the error codes used to report unexpected errors related to the payload received in a downloaded file from esMD. For more details, please refer to section 12.2.4 Administrative Error Response to Inbound Submissions.

Table 78: Administrative Error Codes

Administrative Error	Error Code	Description
Corrupt files/cannot read files	ESMD_410	ESMD_410- Administrative Error (corrupt files/cannot read files).
Submission Sent to Incorrect RC	ESMD_411	ESMD_411- Administrative Error (Submission Sent to Incorrect RC).
Virus Found	ESMD_412	ESMD_412- Administrative Error (Virus Found).
Other	ESMD_413	ESMD_413- Administrative Error (Other).
Incomplete File	ESMD_414	ESMD_414- Administrative Error (Incomplete File).
Unsolicited Response	ESMD_415	ESMD_415- Administrative Error (Unsolicited Response).
Documentation cannot be matched to a case/claim	ESMD_416	ESMD_416- Administrative Error (Documentation cannot be matched to a case/claim).
Duplicate	ESMD_417	ESMD_417- Administrative Error (Duplicate).
Document Code Flat File Validation Error	ESMD_515	Validation error on the document code file
The date(s) of service on the cover sheet received is missing or invalid.	GEX10	The date(s) of service on the cover sheet received is missing or invalid.
The NPI on the cover sheet received is missing or invalid.	GEX11	The NPI on the cover sheet received is missing or invalid.

Administrative Error	Error Code	Description
The state where services were provided is missing or invalid on the cover sheet received.	GEX12	The state where services were provided is missing or invalid on the cover sheet received.
The Medicare ID on the cover sheet received is missing or invalid.	GEX13	The Medicare ID on the cover sheet received is missing or invalid.
The billed amount on the cover sheet received is missing or invalid.	GEX14	The billed amount on the cover sheet received is missing or invalid.
The contact phone number on the cover sheet received is missing or invalid.	GEX15	The contact phone number on the cover sheet received is missing or invalid.
The beneficiary name on the cover sheet received is missing or invalid	GEX16	The beneficiary name on the cover sheet received is missing or invalid.
The claim number on the cover sheet received is missing or invalid.	GEX17	The claim number on the cover sheet received is missing or invalid.
The ACN on the coversheet received is missing or invalid.	GEX18	The ACN on the coversheet received is missing or invalid.

18.2.2 Pickup Errors

Table 79: Pickup Error Codes lists the types of error codes and their descriptions. These codes are used to populate the ErrorInfo object inside the error pickup notification .Please refer to section 12.2.2 Error Pickup Notification for more details.

Table 79: Pickup Error Codes

Error Type	Error Code	Description
UNZIP ERROR	534	ESMD_534 – RC Client processing error (Unzip failure). Please resubmit.
CHECKSUM ERROR	535	ESMD_535 – RC Client processing error (Checksum issue). Please resubmit.
METADATA ERROR	536	ESMD_536 – RC Client processing error (Metadata issue). Please resubmit.
ERROR DOCUMENT CODES VALIDATE FILE	515	Invalid line length for line 1; Expected: 1035, Actual: 1021 Invalid line length for line 2; Expected: 1035, Actual: 1028 Invalid line length for line 7; Expected: 1035, Actual: 1029 Invalid line length for line 8; Expected: 1035, Actual: 1029 Invalid line length for line 9; Expected: 1035, Actual: 1025 **Note**: This is dynamic error message based on the edit validation.

19. PA Requests and Responses Automation with Shared Systems

19.1 Introduction

PA requests and responses are exchanged between the Providers and RCs via mail and fax as well as through the esMD system. esMD allows the exchange of PA information in electronic format as Accredited Standards Committee (ASC) X12N 278 transactions (requests/responses) along with the current acceptable format as XDR transactions. The corresponding medical documentation to the PA request is in XDR (PDF) format only.

19.1.1 Overview of the Automation Process

Currently, populating the PA screens in the Shared Systems is a manual process that is laborious and time consuming. The RCs receive the requests, manually enter the information, and respond with a written response or a response entered into RC Client. With the automation of PA requests/responses, esMD will intake the PA requests, automatically send the requests into the Shared System PA Screens and process the finalized PA requests sent from Shared Systems. This implementation will remove the manual data entry of X12N 278 PA request information into the PA screens by the RCs.

Refer to sections 19.2.1 Logical Workflow and 19.2.2 Application Workflow for detailed information on the automation processing of PA requests and responses with Shared System/Workloads.

19.1.2 Shared Systems

The automation of PA requests/responses will be implemented at different timelines by each of the Shared Systems (Multi-Carrier System (MCS), Viable Information Processing System (VIPS) Medicare System (VMS), and Fiscal Intermediary Shared System (FISS)).

In October 2016, release AR2016.10.0 implemented the changes in the esMD System to cover the initial rollout changes at MCS and Part B RCs.

19.1.2.1 PA Review Response

The X12N 278 Part B and XDR PA Review Response can be submitted using the Shared System PA Screens.

Assumptions

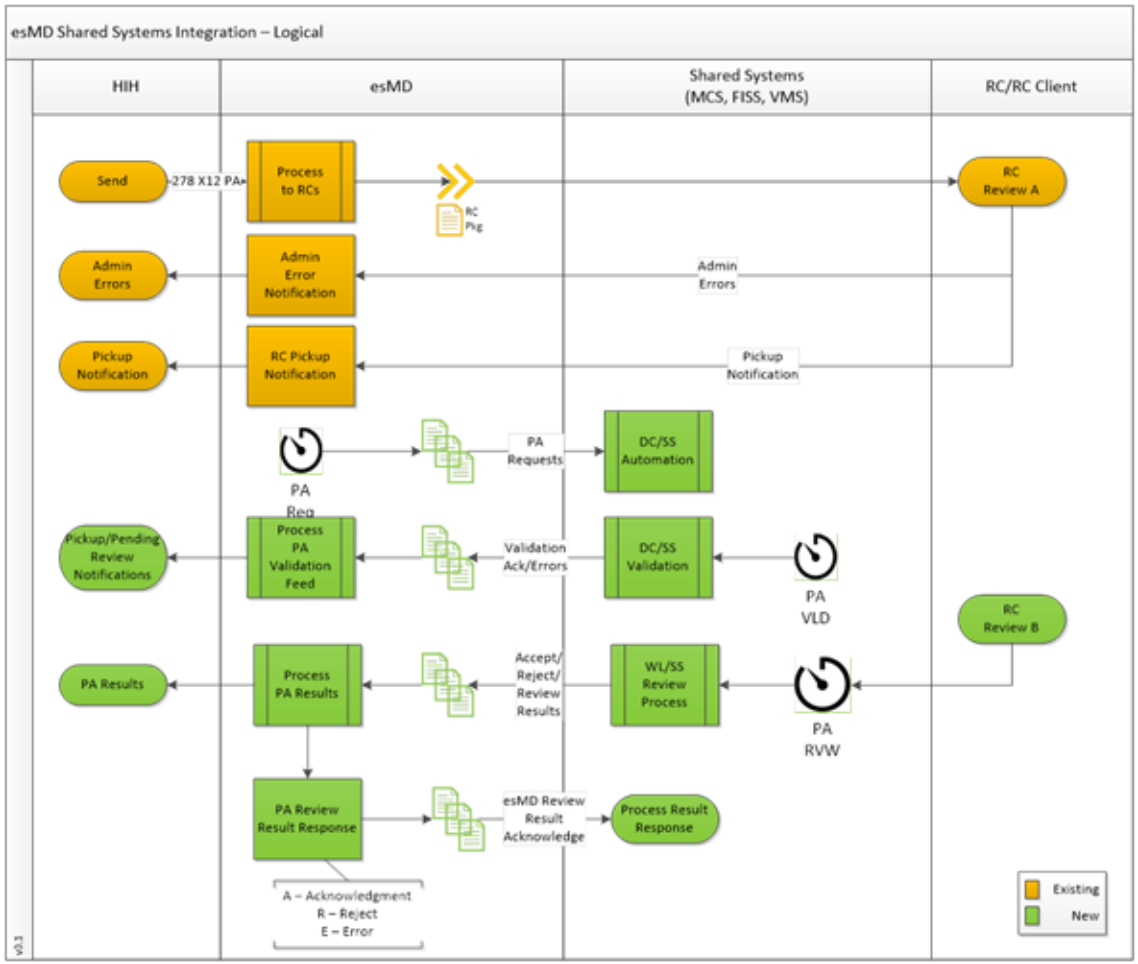
1. The esMD system will not perform any virus scanning of the batch file responses received from the shared system (data center or workload); and
2. No User Interface feature will be available for eMDR and ICDT; only the API will be provided to support the ICDT, RRL, and PA/PCR functionality.

19.2 Automation of PA Requests/Responses – Application Workflow

19.2.1 Logical Workflow

Figure 19: esMD Shared System Integration - Logical provides an overview of the logical flow of PA Requests/Responses between esMD and Shared System (data center or workload).

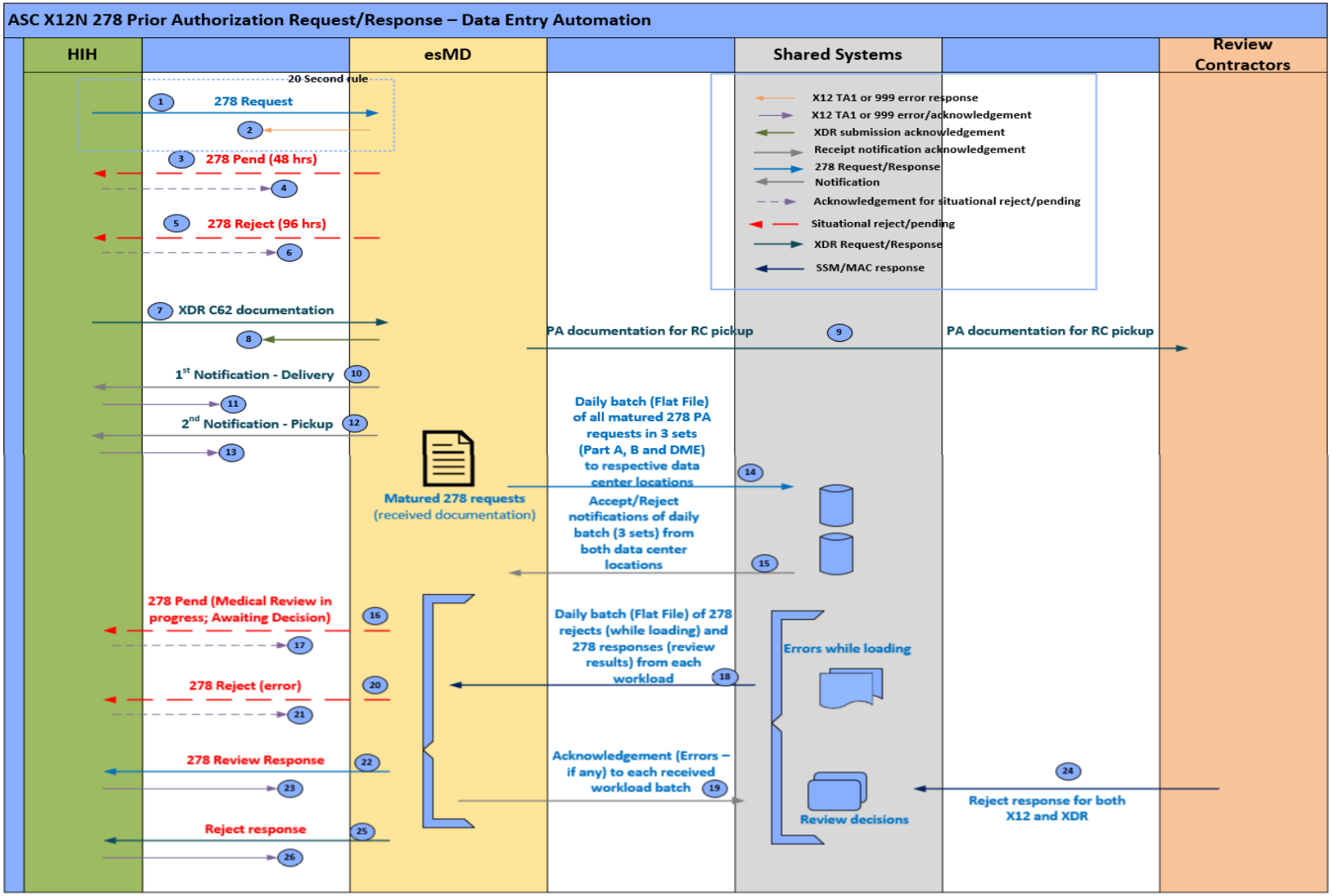
Figure 19: esMD Shared System Integration - Logical



19.2.2 Application Workflow

Figure 20: Information Flow – X12N 278 PA Request/Response Integration with Shared Systems provides an overview of the workflow of automation of X12N 278 PA Requests/Responses between esMD and Shared System/Workload.

Figure 20: Information Flow – X12N 278 PA Request/Response Integration with Shared Systems



20. Inbound/Outbound File Names and Data Directories

Table 80: Inbound/Outbound File Names and Data Directories lists all the files received by the RC and the corresponding data directories these files will reside in along with a brief description.

Note: GUID refers to esMD Transaction ID.

Table 80: Inbound/Outbound File Names and Data Directories

Data Directory	Program Type	Folder Names	XML File Name	Notes
Input	ADR Request (CTC-1)	E_L1_GUID	E_L1_GUID_metadata.xml	N/A
Error	ADR Validation error response	F_L1_GUID	F_L1_GUID_Validation_Error.xml	N/A
notification	ADR HIH delivery notification	N_L1_GUID_MMDDYY_HHMM	N_L1_GUID_Delivery_Acknowledgement.xml	N/A
N/A	ADR Pickup Notification	P_L1_GUID	P_L1_GUID_Pickup.xml	N/A
input	PWK Unsolicited documents (CTC-7)	E_L7_GUID	E_L7_GUID_metadata.xml	N/A
error	PWK Unsolicited Validation error response	F_L7_GUID	F_L7_GUID_Validation_Error.xml	N/A
notification	PWK Unsolicited HIH delivery notification	N_L7_GUID_MMDDYY_HHMM	N_L7_GUID_Delivery_Acknowledgement.xml	N/A
N/A	PWK Unsolicited Pickup Notification	P_L7_GUID	P_L7_GUID_Pickup.xml	N/A
input	First level Appeals (CTC-9)	E_L9_GUID	E_L9_GUID_metadata.xml	N/A
error	First Level Appeals Validation error response	F_L9_GUID	F_L9_GUID_Validation_Error.xml	N/A
notification	First Level Appeals HIH Delivery Notification	N_L9_GUID_MMDDYY_HHMM	N_L9_GUID_Delivery_Acknowledgement.xml	N/A
N/A	Pickup Notification	P_L9_GUID	P_L9_GUID_Pickup.xml	N/A
input	Second Level Appeals (CTC 9.1)	E_L9_1_GUID	E_L9_1_GUID_metadata.xml	N/A

Data Directory	Program Type	Folder Names	XML File Name	Notes
error	Second Level Appeals Validation error response	F_L9_1_GUID	F_L9_1_GUID_Validation_Error.xml	N/A
notification	Second Level Appeals HIH delivery notification	N_L1_GUID_MMDDYY_HHMM	N_L9_1_GUID_Delivery_Acknowledgement.xml	N/A
N/A	Second Level Appeals Pickup Notification	P_L9_1_GUID	P_L9_1_GUID_Pickup.xml	N/A
input	ADMC (CTC 10)	E_L10_GUID	E_L10_GUID_metadata.xml	N/A
error	ADMC Validation error response	F_L10_GUID	F_L10_GUID_Validation_Error.xml	N/A
notification	ADMC HIH delivery notification	N_L10_GUID_MMDDYY_HHMM	N_L10_GUID_Delivery_Acknowledgement.xml	N/A
N/A	ADMC Pickup Notification	P_L10_GUID	P_L10_GUID_Pickup.xml	N/A
input	RAC Discussion Request (CTC 11)	E_L11_GUID	E_L11_GUID_metadata.xml	N/A
error	RAC Discussion Validation error response	F_L11_GUID	F_L11_GUID_Validation_Error.xml	N/A
notification	RAC Discussion HIH delivery notification	N_L11_GUID_MMDDYY_HHMM	N_L11_GUID_Delivery_Acknowledgement.xml	N/A
N/A	RAC Discussion Pickup Notification	P_L11_GUID	P_L11_GUID_Pickup.xml	N/A
inpit	Phone Discussion Request (CTC 11.1)	E_L11_1_GUID	E_L11_1_GUID_metadata.xml	N/A
error	Phone Discussion Validation error response	F_L11_1_GUID	F_L11_1_GUID_Validation_Error.xml	N/A
notification	Phone Discussion HIH delivery notification	N_L11_1_GUID_MMDDYY_HHMM	N_L11_1_GUID_Delivery_Acknowledgement.xml	N/A
N/A	Phone Discussion Pickup Notification	P_L11_1_GUID	P_L11_1_GUID_Pickup.xml	N/A
input	Ambulance (CTC 8.1)	E_L8_1_GUID	E_L8_1_GUID_metadata.xml	N/A
error	Ambulance Validation error response	F_L8_1_GUID	F_L8_1_GUID_Validation_Error.xml	N/A
notification	Ambulance HIH delivery notification	N_L8_1_GUID_MMDDYY_HHMM	N_L8_1_GUID_Delivery_Acknowledgement.xml	N/A
N/A	Ambulance Pickup Notification	P_L8_1_GUID	P_L8_1_GUID_Pickup.xml	N/A
input	HHPCR (CTC 8.3)	E_L8_3_GUID	E_L8_3_GUID_metadata.xml	N/A

Data Directory	Program Type	Folder Names	XML File Name	Notes
error	HHPCR Validation error response	F_L8_3_GUID	F_L8_3_GUID_Validation_Error.xml	N/A
notification	HHPCR HIH delivery notification	N_L8_3_GUID_MMDDYY_HHMM	N_L8_3_GUID_Delivery_Acknowledgement.xml	N/A
N/A	HHPCR Pickup Notification	P_L8_3_GUID	P_L8_3_GUID_Pickup.xml	N/A
input	DMEPOS (CTC 8.4)	E_L8_4_GUID	E_L8_4_GUID_metadata.xml	N/A
error	DMEPOS Validation error response	F_L8_4_GUID	F_L8_4_GUID_Validation_Error.xml	N/A
notification	DMEPOS HIH delivery notification	N_L8_4_GUID_MMDDYY_HHMM	N_L8_4_GUID_Delivery_Acknowledgement.xml	N/A
N/A	DMEPOS Pickup Notification	P_L8_4_GUID	P_L8_4_GUID_Pickup.xml	N/A
input	X12 XDR (CTC 12)	E_L12_GUID	E_L12_GUID_metadata.xml	N/A
error	Validation error response	F_L12_GUID	F_L12_GUID_Validation_Error.xml	N/A
notification	HIH delivery notification	N_L12_GUID_MMDDYY_HHMM	N_L12_GUID_Delivery_Acknowledgement.xml	N/A
N/A	Pickup Notification	P_L12_GUID	P_L12_GUID_Pickup.xml	N/A
input	Additional Documentation X12 XDR (CTC 13)	E_L13_GUID	E_L13_GUID_metadata.xml	N/A
error	Validation error response	F_L13_GUID	F_L13_GUID_Validation_Error.xml	N/A
notification	HIH delivery notification	N_L3_GUID_MMDDYY_HHMM	N_L13_GUID_Delivery_Acknowledgement.xml	N/A
N/A	Pickup Notification	P_L13_GUID	P_L13_GUID_Pickup.xml	N/A
N/A	RRL (CTC 1.3)	U_UniqueId	U_UniqueID_esMD_ProcessMetadata.xml	N/A
Acknowledgment	RRL esMD Acknowledgment	A_L1_3_GUID	A_L1_3_GUID_Receipt_Acknowledgement.xml	N/A
error	RRL Validation error response	F_L1_3_GUID	F_L1_3_GUID_Validation_Error.xml	N/A
notification	RRL HIH delivery notification	N_L1_3_GUID_MMDDYY_HHMM	N_L1_3_GUID_Delivery_Acknowledgement.xml	N/A
N/A	Decision Letters (CTC 1.4)	U_UniqueId	U_UniqueID_esMD_ProcessMetadata.xml	N/A
Acknowledgment	Decision Letters esMD Acknowledgment	A_L1_4_GUID	A_L1_4_GUID_Receipt_Acknowledgement.xml	N/A
error	Decision Letters Validation error response	F_L1_4_GUID	F_L1_4_GUID_Validation_Error.xml	N/A

Data Directory	Program Type	Folder Names	XML File Name	Notes
notification	Decision Letters HIH delivery notification	N_L1_4_GUID_MMDDYY_HHMM	N_L1_4_GUID_Delivery_Acknowledgement.xml	N/A
N/A	Admin Error Response	D_ADM_GUID	D_ADM_GUID_AdminResponse.xml	The Folder Name and XML File Name are the same for all the following Content Type Codes: CTC-1, CTC-7, CTC-8.1, CTC-8.2, CTC-8.3, CTC-8.4, CTC-9, CTC-9.1, CTC-10, CTC-11, CTC-11.1, CTC-12, and CTC-13
N/A	Reject Review Response	R_PA_GUID	R_PA_GUID_Response.xml	The Folder Name and XML File Name are the same for the following Content Type Codes: CTC-8.1, CTC-8.2, CTC-8.3, CTC-8.4, and CTC-13
error	Virus Scan Timeout	Y_GUID	Y_<<RandomNo>>_Virus_Scan_Gateway_Failure.xml	N/A
error	Virus Found or Infected File	X_GUID	X_<<RandomNo>>_Virus_Scan_Error.xml	N/A
ICDT/input	ICDT - Solicited Request	Q_UniqueID	T#EFT.ON.<<ReceiverRoutingId>>.L15_1.Q<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	Q_<<RCUniqueId>>_ICDTSolicitedRequest.xml
ICDT/input	ICDT - Solicited Response	R_UniqueID	T#EFT.ON.<<ReceiverRoutingId>>.L15_2.R<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	R_<<RCUniqueId>>_ICDTSolicitedResponse.xml
ICDT/input	ICDT - Unsolicited Response	R_UniqueID	T#EFT.ON.<<ReceiverRoutingId>>.L15_3.R<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	R_<<RCUniqueId>>_ICDTUnSolicitedResponse.xml
Icdt\ntfn_ack	ICDT – Pickup Notifications/Acknowledgment	B_3CharRandomId	N/A	T#EFT.ON.<<ReceiverRoutingId>>.ICDT.B<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS

Data Directory	Program Type	Folder Names	XML File Name	Notes
Icdt\error	ICDT - Admin Error Response	C_3CharRandomId	N/A	T#EFT.ON.<<ReceiverRoutingId>>.ADM.C<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS
Icdt\error	ICDT Solicited Request - Validation errors from esMD	V_PackageUniqueID	T#EFT.ON.<<ReceiverRoutingId>>.L15_1.V<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	V_Packageunique id_Validation_Error.xml
Icdt\error	ICDT Solicited Response - Validation errors from esMD	V_PackageUniqueID	T#EFT.ON.<<ReceiverRoutingId>>.L15_2.V<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	V_Packageunique id_Validation_Error.xml
Icdt\error	ICDT Unsolicited Response - Validation errors from esMD	V_PackageUniqueID	T#EFT.ON.<<ReceiverRoutingId>>.L15_3.V<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	V_Packageunique id_Validation_Error.xml
eMDRRegistration	eMDR Registration Batch File from esMD	E_3CharRandomId	T.<<ReceiverRoutingId>>.L5.E<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	N/A
N/A	Pickup Notification for eMDR Registered Provider Batch File	P_L5_3CharRandomID	T.<<ReceiverRoutingId>>.L5.P<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	P_L5_3CharRandomID_Pickup.xml
N/A	PrePay eMDR letters (CTC 1.5)	U_UniqueId	U_UniqueID_eMDR_ProcessMetadata.xml	N/A
Acknowledgment	PrePay eMDR letters Acknowledgment	A_L1_5_GUID	A_L1_5_GUID_Receipt_Acknowledgement.xml	N/A
error	PrePay eMDR letters Validation error response	F_L1_5_GUID	F_L1_5_GUID_Validation_Error.xml	N/A
N/A	PostPay eMDR letters (CTC 1.6)	U_UniqueId	U_UniqueID_eMDR_ProcessMetadata.xml	N/A
Acknowledgment	PostPay eMDR letters Acknowledgment	A_L1_6_GUID	A_L1_6_GUID_Receipt_Acknowledgement.xml	N/A
error	PostPay eMDR letters Validation error response	F_L1_6_GUID	F_L1_6_GUID_Validation_Error.xml	N/A
Input	Document Code Batch File from esMD	E_3CharRandomId	T.<<ReceiverRoutingId>>.L17.E<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	N/A
N/A	Pickup Notification for document Code Batch File	P_L17_3CharRandomID	T.<<ReceiverRoutingId>>.L17.P<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.THHmmssS	P_L17_3CharRandomID_Pickup.xml
input	HOPD (CTC 8.5)	E_L8_5_GUID	E_L8_5_GUID_metadata.xml	N/A

Data Directory	Program Type	Folder Names	XML File Name	Notes
error	HOPD Validation error response	F_L8_5_GUID	F_L8_5_GUID_Validation_Error.xml	N/A
notification	HOPD HIH delivery notification	N_L8_5_GUID_MMDDYY_HHMM	N_L8_5_GUID_Delivery_Acknowledgement.xml	N/A
N/A	HOPD Pickup Notification	P_L8_5_GUID	P_L8_5_GUID_Pickup.xml	N/A

21. Contacts

Table 81: Support Points of Contact list for esMD.

Table 81: Support Points of Contact

Contact	Phone	Email	Hours of Operation
CMS esMD Service Desk	(443) 832-1856	esMD_Support@cms.hhs.gov	Regular Business Hours: 8 a.m. to 8 p.m. Eastern Time (ET).

Appendix A: Description of Fields on RC Client Tabs

Table 82: Descriptions of Fields on Error Response to PA Request Tab lists the descriptions of the fields on the Review Decision Response to PA Request tab.

Table 82: Descriptions of Fields on Error Response to PA Request Tab

Name of Field	Description
Transaction ID	The esMD TransactionId format is as follows: • The length of TransactionId will be 15 alphanumeric characters; and • TransactionId will consist of alphabetic (a-z, A-Z) and numeric (0-9) characters.
Reject Error Category	One or multiple Reject Error Category is selected for each Response; each Reject Error Category has number of Reject Error Codes associated with it. Required Element.
Reject Error Code	Under Each Reject Error Category, either one or multiple Reject Error Codes are selected. Required Element. Minimum 1 and maximum 9 reject error codes can be selected for each category.
Decision Indicator	Decision provided for the Error Response should be the following: • "R"- Decision Indicator. "R" is provided in the response when the Decision is "Rejected" for the request received.
Reason Code	5-character reason code is provided. Minimum of 1 and up to maximum of 25 reason codes can be provided. Required Element.
Request Level Unique Tracking Number (UTN)	UTN is provided for each response. Optional Element. Format of the unique tracking number is 14 Alpha Numeric Characters.

Table 83: Descriptions of Fields on Administrative Error Response to Inbound Submissions Tab lists the descriptions of the fields on the Administrative Error Response to Inbound Submissions tab.

Table 83: Descriptions of Fields on Administrative Error Response to Inbound Submissions Tab

Name of Field	Description
Transaction ID	Transaction Identifier of the request this response is being sent for. Required Element. It is a 15 Alphanumeric Value.
Error Situation	Error code/situation; can be one of the following: • Corrupt Files/Cannot read file, • virus found, • Submission sent to incorrect RC, • Other, • Incomplete File, • Unsolicited Response, • Documentation cannot be matched to a case/claim, • Duplicate, •

Table 84: Descriptions of Fields on Advanced/Debugging Tab lists the descriptions of the fields on the Advanced/Debugging tab.

Table 84: Descriptions of Fields on Advanced/Debugging Tab

Name of Field	Description
User ID	IDM User ID Required Element for testing the connectivity to TIBCO MFT Server.
Password	IDM password Required Element for testing the connectivity to TIBCO MFT Server.

Appendix B: Reject Error Codes

For an up-to-date list of Reject Error Codes, please refer to the esMD Downloads section, using the link below:

http://www.cms.gov/Research-Statistics-Data-and-Systems/Computer-Data-and-Systems/ESMD/Information_for_Review-Contractors.html

Note: An up-to-date list of Reject Error Codes will be added to this web site by CMS.

Appendix C: Industry Codes

For an up-to-date list of Industry Codes, please refer to the esMD Downloads section, using the link below:

http://www.cms.gov/Research-Statistics-Data-and-Systems/Computer-Data-and-Systems/ESMD/Information_for_Review-Contractors.html

Note: An up-to-date list of Industry Codes will be added to this web site by CMS.

Appendix D: Content Type Codes

Table 85: Content Type Code Descriptions provides the descriptions of the Content Type Codes.

Table 85: Content Type Code Descriptions

Content Type Code	Description	Comment
1	Response to ADR	N/A
1.3	ADR RRL	ADR RRL from RC to esMD.
1.4	PA/PCR Decision Letters	PA/PCR Decision Letters from RC to esMD.
1.5	PrePay eMDR letters	PrePay eMDR letters from RC to esMD
1.6	Postpay eMDR letters	Post-Pay eMDR letters from RC to esMD
5	Service Registration	Service Registration
7	PWK Unsolicited documents	PWK Unsolicited documents
8.1	Non-Emergent Ambulance Transport	N/A
8.3	HHPCR	N/A
8.4	DMEPOS	N/A
8.5	HOPD	N/A
9	First Level Appeal Requests	N/A
9.1	Second Level Appeal Requests	N/A
10	ADMC	N/A
11	RA Requests	N/A
11.1	DME Phone Discussion Requests	N/A
13	Supporting Documentation for the X12N 278 Request	N/A
15.1	ICDT Request	Supports requests for documentation from an RC to another RC.
15.2	ICDT Solicited Response	Supports responses from an RC for previously requested documentation to another RC.
15.3	ICDT Unsolicited Response	Supports an RC sending misdirected documentation to another RC.
17	Document Codes	Document Codes

Table 86: Content Type Codes and Business Types shows the list of Business Types associated with them.

Table 86: Content Type Codes and Business Types

Content Type Code	Business Type
1	Response message for additional documentation request
1.3	ADR Review Result Letter
1.4	PA/PCR Decision Letters
1.5	Pre-Pay eMDR
1.6	Post-Pay eMDR
5	Service Registration
7	PWK Unsolicited documents
8.1	Non-Emergent Ambulance Transport.
8.3	HHPCR
8.4	DMEPOS PA
8.5	HOPD
9	First Level Appeal
9.1	Second Level Appeal
10	ADMC
11	RA Requests
11.1	DME Phone Discussion Requests
13	XDR X12
15.1	ICDT Request
15.2	ICDT Solicited Response
15.3	ICDT Unsolicited Response
17	Document Code File

Appendix E: Acronyms

Table 87: Acronyms

Acronym	Literal Translation
A/B	Part A/Part B
ACN	Attachment Control Number
ADMC	Advance Determination of Medicare Coverage
ADR	Additional Documentation Request
API	Application Programming Interface
ASC	Accredited Standards Committee
C-CDA	Consolidated Clinical Document Architecture
CMS	Centers for Medicare & Medicaid Services
CTC	Content Type Code
DME	Durable Medical Equipment
DMEPOS	Durable Medical Equipment, Prosthetics/Orthotics and Supplies
EDI	Electronic Data Interchange
EFT	Enterprise File Transfer
IDM	Identity Management
eMDR	Electronic Medical Documentation Request
esMD	Electronic Submittal of Medical Documentation
FFS	Fee-For-Service
GUI	Graphical User Interface
GUID	esMD Transaction ID
HHPCR	Home Health Services Pre-Claim Review
HIC	Health Insurance Claim
HICN	Health Insurance Claim Number
HIH	Health Information Handler
HL7	Health Level Seven International
HOPD	Hospital Outpatient Department
ICDT	Inter Contractor Document Transfer
ID	Identifier
JAR	Java Archive
JCE	Java Cryptography Encryption
JDK	Java Development Kit
JKS	Java KeyStore
JRE	Java Runtime Environment
Kbps	Kilobits Per Second
LOB	Line of Business

Acronym	Literal Translation
MAC	Medicare Administrative Contractor
MB	Megabytes
MCS	Multi-Carrier System
MFT	Managed File Transfer
MIME	Multipurpose Internet Mail Extension
NPI	National Provider Identifier
OID	Object Identifier or Organizational Identifier
PA	Prior Authorization
PCR	Pre-Claim Review
PDF	Portable Document Format
PROD	Production
PWK	Paperwork
QIC	Qualified Independent Contractor
RA	Recovery Auditor
RAC	Recovery Audit Contractor
RC	Review Contractor
RRL	Review Results Letter
RSA	Rivest, Shamir & Adleman
SFTP	SSH File Transfer Protocol
SSH	Secure Shell
TID	Transaction Identifier
UAT	User Acceptance Testing
UI	User Interface
UID	Unique Identifier
URL	Universal Resource Locator
UTN	Universal Tracking Number
VDC	Virtual Data Center
XDR	Cross-Enterprise Document Reliable Interchange
XML	Extensible Markup Language
ZPIC	Zone Program Integrity Contractor

Appendix F: Glossary

Table 88: Glossary

Glossary	Description
Additional Documentation Request (ADR)	Official letters sent to Providers from CMS RCs requesting additional documentation that is needed to process claims.
Advanced Determination of Medical Coverage (ADMC)	A voluntary program that allows Suppliers and Beneficiaries to request prior approval of eligible items (e.g., wheelchairs) before delivery of the items to the beneficiary.
CONNECT	CONNECT implements a flexible, open-source gateway solution that enables healthcare entities - Federal agencies or private-sector health organizations or networks - to connect their existing health information systems to the eHealth Exchange. CONNECT is fully functional out-of-the-box, while at the same time configurable and flexible to allow organizations to customize it to meet their needs and those of their existing health information systems.
Electronic Submission of Medical Documentation (esMD)	A new mechanism for submitting medical documentation via a secure internet gateway connecting Providers to the Centers for Medicare & Medicaid Services (CMS). In its second phase, esMD will allow Medicare RCs to electronically submit claim related Additional Document Request (ADR) letters, and other use case requests, to Providers when their claims are selected for review.
Health Information Handler (HIH)	A Health Information Handler (HIH) is defined as an organization that oversees and governs the exchange of health-related claim reviewer information from Provider to CMS esMD Gateway according to nationally recognized standards.
Inter Contractor Document Transfer (ICDT)	A new functionality that allows RCs to exchange files/documents from one RC to another RC through the esMD system.
Interface	A well-defined boundary where direct contact between two different environments, systems, etc., occurs, and where information is exchanged.
Power Mobility Device (PMD) Prior Authorization (PA)	The CMS implemented a Prior Authorization process for scooters and power wheelchairs for people with Fee-For-Service Medicare who reside in seven states with high populations of fraud- and error-prone Providers (CA, FL, IL, MI, NY, NC, and TX). This demonstration will help ensure that a beneficiary's medical condition warrants their medical equipment under existing coverage guidelines. Moreover, the program will assist in preserving a Medicare beneficiary's ability to receive quality products from accredited suppliers.
Security	The physical, technological, and administrative safeguards used to protect individually identifiable health information.
Simple Object Access Protocol (SOAP)	Simple Object Access Protocol is a message exchange format for web services.
Transaction	Event or process (such as an input message) initiated or invoked by a user or system, regarded as a single unit of work and requiring a record to be generated for processing in a database.

Appendix G: FAQ's

- RC Client is inactive, but the screen says "Login Successful. RC Client is Active"
    - The blue successful message that is shown while logging in "Login Successful. RC Client is Active" only means that the login was successful and RC client is now active at the time. If there is any temporary internet disconnection, RC Client will stop pulling and pushing the document. In that case, please check the log files and make sure the timestamp is up to date. If not, please restart the RC Client.
- RC Client is not working properly, when using multiple instances
    - It's not advised to use multiple copies of RC Client simultaneously. Only use one copy at time.

      Note: Running multiple instances of the Java RC Client for the same jurisdiction could result in errors while pulling the files.
- RC Client is unable to download the files, every file is erroring out.
    - RC Client needs folder permission to download the files. It needs folder read/write permission to download and copy the files. Please check with your IT team if there is any such issue.
- User can log into the CMS portal (https://home.idm.cms.gov/)but not into RC Client
    - Please make sure your KeyStore is created and updated with new or reset password.

Appendix H: Approvals

The undersigned acknowledge that they have reviewed the Review Contractor (RC) Client Java User Guide and Installation Handbook, Version 8.1, and agree with the information presented within this document. Changes to this Guide will be coordinated with, and approved by, the undersigned, or their designated representatives.

Signature:		Date:	01/21/2021
Print Name:	Ayana Chavis		
Title:	Contracting Officer's Representative		
Role:	CMS Approving Authority		