# Document Change History

## Table 1: Record of Changes

<table>
<thead>
<tr>
<th>Version Number</th>
<th>Date</th>
<th>Author/Owner</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.0</td>
<td>08/22/2022</td>
<td>Venkata Gurram</td>
<td>1. Updated Sections 1.2, 1.4, 1.4.1, 1.4.2, 1.4.3, 2, 3, 4, 5, 6, 7.2, 8.1.2, 8.2, 9, 12.1.4, 12.1.6, 12.2.1, 12.2.2, 12.2.8, 12.2.9, 12.2.10, 12.2.14, 13, 13.1, 13.2, 14, 15, 16, 17, 17.1.1, 17.1.2, 18, 19, 19.3, 20.3.1, 20.4.1, 20.4.3, 20.4.4, 20.5, 20.5.1, 20.6.1, 20.6.2, 20.7, 20.8, 20.9, 20.10, 21.3, 21.3.1, 21.3.2, 21.3.3, 21.3.6, 25. 2. Updated Tables 4, 4.1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 37, 37, 38, 39, 41, 44, 47, 48, 49, 50, 51, 63, 69. 3. Page 15 - Updated hyperlink for IDM’s public facing documentation.</td>
</tr>
<tr>
<td>10.0</td>
<td>01/27/2022</td>
<td>Khader Mohammad/Lisa Massengill</td>
<td>• Updated changes for the April 2022 release in Section 1.4, 4, 11.1, 17.1.  • Updated Figure 3 and Tables 31 and 71.  • Updated Section 4, steps 2, 3, 4, &amp; 5.</td>
</tr>
<tr>
<td>9.0</td>
<td>10/19/2021</td>
<td>Rohini Kolan</td>
<td>4. Included January 2022 release functionality in section 1.4 5. Removed all references to Review Result Letters (CTC 1.3), PA/PCR Decision Letters (CTC 1.4) 6. Updated Sections 4 and 12.1.9 7. Updated Tables 13, 74, and 81</td>
</tr>
<tr>
<td>8.2</td>
<td>07/09/2021</td>
<td>Rohini Kolan</td>
<td>8. Included October release functionality in section 1.4 9. Updated sections 1.5.1,3, 4, 12.1.2, 12.1.23, 15.2.2</td>
</tr>
<tr>
<td>8.1</td>
<td>01/21/2021</td>
<td>Steven Prest</td>
<td>10. Replaced broken image in Figure 15</td>
</tr>
<tr>
<td>8.0</td>
<td>01/15/2021</td>
<td>Karthik Srinivasan</td>
<td>11. Updated document to reflect changes from EIDM to IDM. 12. Updated section 1.4.3 13. Updated Table 76 – Request Level UTN: Changed tracking number from “1-50 Alphanumeric Characters” to “14 Alphanumeric Characters”. 14. Replaced all “EIDM” links with “IDM” links</td>
</tr>
<tr>
<td>7.1</td>
<td>10/13/2020</td>
<td>Vijayalakshmi Muthukrishnan</td>
<td>15. Updated Table 72</td>
</tr>
<tr>
<td>Version Number</td>
<td>Date</td>
<td>Author/Owner</td>
<td>Description of Change</td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>----------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 7.0            | 07/27/2020 | Boris Nakshun              | 16. Included November release functionality in Section 1.4  
17. Updated Table 6, 71, 72, 74, 77, 78, 79 and 80.  
18. Updated Section 12.1.2.1, 12.2, 13.6, 19.1 and 19.1.2.1  
19. Updated Figure 20  
20. Added Appendix G - FAQs  
21. Removed all reference to Coversheet and FFR throughout the document.  
22. Removed all errors related to Coversheet and FFR in section 18.2.1 |
| 6.1            | 04/20/2020 | Karthik Srinivasan         | Added  
- Sections 12.1.26, 12.2.18, 12.2.19, 12.2.20, 14.5, 15  
- Figure 14  
Updated  
- Section 1.4 – to include AR2020.07.0 release details  
- Section 1.4.1 – to include Document Code File Request and pickup notification for RC client functionality  
- Section 12.2.12 – Replaced “Zip” with “flat” in beginning sentence.  
- Section  
- Table 4 – Included ICDT Directory  
- Table 5 – Inbound for Document Codes Flat file  
- Table 74 – Included Error Document Codes Validate File  
- Table 75  
- Table 81 – Included CTC 8.5 & 17  
- Table 82 – Included CTC 17 |
| 6.0            | 09/30/2019 | Vijayalakshmi Muthukrishnan | 1. Updated for AR2020.01.0:  
a. Sections 1.4, 10.1, 12.1, and 12.2.  
b. Tables 5, 68, 71, and 77.  
2. Added:  
a. Sections 12.1.24, 12.1.25, 12.2.15 through 12.2.17, 14.5, and 17.8.  
b. Figures 16 and 17.  
c. Tables 28, 29, 42 through 46, 49, 50, and 67. |
| 5.2            | 05/14/2019 | Vijayalakshmi Muthukrishnan | Updated:  
1. Sections 1.4 and 6 Step 2.  
2. Figures in Section 3 Step 2. Section 4 Steps 2, 3, and 5. Section 5 Steps 1 through 3. Section 6 Steps 1 and 2.  
3. Tables 5, 13, 14, 28 (title), 29 (title), and 61. |
<table>
<thead>
<tr>
<th>Version Number</th>
<th>Date</th>
<th>Author/Owner</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>04/09/2019</td>
<td>Vijayalakshmi Muthukrishnan</td>
<td>Resolved review comments. Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1. Sections 1.4, 2, and 12.1.10.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Table 66.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. Figure 7.</td>
</tr>
<tr>
<td>5.0</td>
<td>03/29/2019</td>
<td>Vijayalakshmi Muthukrishnan</td>
<td>1. Updated for AR2019.07 release:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>a. Sections 1.4, 1.4.1, 3 through 5, 8.2, 9, 12.1.4, 12.1.5, 12.1.7, 12.2.10, 12.2.11, 18.2.2, and Appendix A.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b. Tables 5, 7, 9 through 39, 58, 60, 63, 64, and 66 through 70.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>c. Figures 6 through 9.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b. Tables 6, 41, 50, and 61.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>c. Figures 3, through 5, and 16.</td>
</tr>
<tr>
<td>4.1</td>
<td>07/16/2018</td>
<td>Vijayalakshmi Muthukrishnan</td>
<td>Updated to resolve review comments in sections 1.4.1 and 1.5.1.</td>
</tr>
<tr>
<td>4.0</td>
<td>06/27/2018</td>
<td>Vijayalakshmi Muthukrishnan</td>
<td>Updated for AR2018.10.0:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1. Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>o Section 1.4, 11.1, and 11.2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>o Tables 6, 79, and 89 through 92.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>o Sections 11.1.29, 11.1.30, 11.2.22, 11.2.23, 13.5, and 15.2.19.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>o Figure 13.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>o Tables 35, 36, 57, 59, and 78.</td>
</tr>
<tr>
<td>3.1</td>
<td>04/10/2018</td>
<td>Vijayalakshmi Muthukrishnan</td>
<td>1. Resolved review comments in Table 72.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Updated Table 73 to include new error codes from esMD to RC.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>a. Sections 11.1.27 and 11.1.28.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b. Tables 33 and 34.</td>
</tr>
<tr>
<td>3.0</td>
<td>03/13/2018</td>
<td>Vijayalakshmi Muthukrishnan</td>
<td>Original Update for AR2018.07.0:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1. Updated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b. Tables 5, 6, 73, 74, 80, and 83 through 85.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>a. Sections 11.1.26, 11.2.20, 11.2.21, 13.4, and 15.2.18.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b. Figure 12.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>c. Tables 32, 51 through 53, and 71.</td>
</tr>
<tr>
<td>2.1</td>
<td>01/31/2018</td>
<td>Pallavi Sakhare</td>
<td>Updated in response to CMS comments:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1. Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>a. Sections 1.4 paragraph 13 and 14.1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b. Table 76.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Added section 11.2.18.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. Moved:</td>
</tr>
<tr>
<td>Version Number</td>
<td>Date</td>
<td>Author/Owner</td>
<td>Description of Change</td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>-------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>a. Former sections 14.1 and 14.2 to sections 15.2.10 and 15.2.17, respectively.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>b. Former tables 50 and 52 to tables 59 and 66, respectively.</td>
</tr>
<tr>
<td>2.0</td>
<td>01/12/2018</td>
<td>Pallavi Sakhare</td>
<td>Updated updates for AR2018.04.0:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1. Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>o Tables 5, 6, 67, and 77 through 80.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>o Figures 3 through 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>o Sections 11.1.22 through 11.1.25, 11.2.18, 13.3, 15, 15.1, and 15.2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>o Tables 28 through 31, 49, and 52.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>o Figure 11</td>
</tr>
<tr>
<td>1.0</td>
<td>10/04/2017</td>
<td>Vijayalakshmi Muthukrishnan</td>
<td>No substantive changes for Release AR2018.01.0 added.</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **Introduction** .................................................................................................................. 1
   1.1 **Overview of the esMD System** ........................................................................ 1
       1.1.1 The esMD Claim Review Contractors .................................. 1
   1.2 **System Overview** ......................................................................................... 2
   1.3 **System Requirements** .................................................................................. 3
   1.4 **RC Client Overview** ...................................................................................... 3
       1.4.1 RC Client Upload/Download Functionality ....................... 7
       1.4.2 RC Client Application Overview ......................................... 8
       1.4.3 RC Client Operation Overview ........................................ 10
   1.5 **ICDT Overview** ........................................................................................ 11
       1.5.1 RC Client ICDT Folder Structure ........................................ 11

2. **Overview of How This Document is Structured** .................................................... 13

3. **How to Start the RC Client and Log In** ................................................................. 15

4. **How to Enter an Error Code on the Error Response to PA Request Tab** ........ 17

5. **How to Submit an Inbound Submission Error on the Administrative Error Response to Inbound Submissions Tab** ................................................................. 22

6. **How to Verify Connection to esMD Cloud, Using the Advanced/Debugging Tab** 24

7. **System Requirements** ........................................................................................ 25
   7.1 **Processor** ............................................................................................... 25
   7.2 **Disk Space** ............................................................................................ 25
   7.3 **Memory** ............................................................................................... 25
   7.4 **Permissions** .......................................................................................... 25
   7.5 **Network** .............................................................................................. 25
   7.6 **Microsoft .NET Framework** ..................................................................... 25
   7.7 **Libraries** .............................................................................................. 25

8. **How to Install and Configure a Microsoft .NET Version of RC Client** ............... 27
8.1 **Out-of-the-Box** ........................................................................................................ 27  
8.1.1 Keystore Set Up ........................................................................................................ 27  
8.1.2 Configuring the RC Client ..................................................................................... 28  
8.1.3 Running the RC Client .......................................................................................... 31  

8.2 **Custom RC Client** .................................................................................................. 31  

9. **esMD Cloud File Transfers** ................................................................................ 32  

10. **XML Schema Definitions** .................................................................................. 51  

11. **Review Responses Through RC Client** .............................................................. 52  
    11.1 Review Decision Response to PA Request ........................................................ 52  
    11.2 Multiple Review Responses .............................................................................. 53  

12. **XML Messages** .................................................................................................... 54  
    12.1 **Inbound** .......................................................................................................... 54  
        12.1.1 Payload Files ............................................................................................... 55  
        12.1.2 Metadata File ............................................................................................. 55  
        12.1.3 Pickup HIH Status Response ..................................................................... 56  
        12.1.4 Pickup Validation Error Response ............................................................. 57  
        12.1.5 Administrative Error HIH Status Response ................................................ 57  
        12.1.6 Administrative Error HIH Status Response Validation Error .................... 58  
        12.1.7 esMD Virus Scanning Service Down Error Response .............................. 59  
        12.1.8 Virus Scan Error Response ....................................................................... 59  
        12.1.9 PA Reject response Delivery Acknowledgment for X12N 278 or XDR ......... 60  
        12.1.10 PA Review Response Validation Error Response ................................. 61  
        12.1.11 ICDT Request XML ................................................................................... 62  
        12.1.12 CDT Solicited Response XML .................................................................. 63  
        12.1.13 ICDT Unsolicited Response XML .............................................................. 64  
        12.1.14 ICDT Pickup Notification/Acknowledgement Response (as a Batch Process) .................................................................................................................. 65  
        12.1.15 ICDT Validation Error/Pickup Error Notification ....................................... 67  

**AR2023.01.0 Review Contractor (RC) Client Microsoft .NET User Guide and Installation Handbook**

**Version 11.0** vii **Electronic Submission of Medical Documentation (esMD)**
12.1.16 ICDT Administrative Error Response .......................................................... 67
12.1.17 esMD Validation Error Response for Pre-Pay eMDR Letters ........ 68
12.1.18 esMD validation Error Response for Post-Pay eMDR Letters ........ 69
12.1.19 Document Code File ............................................................................... 70

12.2 Outbound ........................................................................................................... 70

12.2.1 Pickup Notification ....................................................................................... 71
12.2.2 Error Pickup Notification ........................................................................... 71
12.2.3 Error Response to PA Request ................................................................. 72
12.2.4 Administrative Error Response to Inbound Submissions ....................... 73
12.2.5 ICDT Request ............................................................................................. 74
12.2.6 ICDT Solicited Response ............................................................................ 74
12.2.7 ICDT Unsolicited Response ....................................................................... 75
12.2.8 ICDT Pickup/Pickup Error Notification ..................................................... 77
12.2.9 ICDT Administrative Error Notification ................................................... 78
12.2.10 Service Registration Pickup Notification ............................................. 78
12.2.11 eMDR Process Metadata (Pre-Pay and Post-Pay eMDR Letters) ... 79
12.2.12 eMDR StructuredFile for Post-Pay ADR Letters .................................. 80
12.2.13 API Error Messages for Prepay and Post-pay ....................................... 81
12.2.14 DCF Pickup Notification ....................................................................... 82
12.2.15 DCF Error Pickup Notification ................................................................ 82
12.2.16 HOPD Pickup notification ..................................................................... 83

13. RC Client Components .......................................................................................... 84

13.1 esMD API’s ...................................................................................................... 85
13.2 Compression Utility ......................................................................................... 85
13.3 Encryption Utility ............................................................................................. 86
13.4 XML Processor ................................................................................................. 86
13.5 Scheduler .......................................................................................................... 86
13.6 Housekeeping Manager ................................................................................ 86
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14. RC Client Workflow</td>
<td>87</td>
</tr>
<tr>
<td>15. Auth API Overview</td>
<td>89</td>
</tr>
<tr>
<td>16. Upload API Overview</td>
<td>91</td>
</tr>
<tr>
<td>17. Download API Overview</td>
<td>95</td>
</tr>
<tr>
<td>17.1.1 Get List of Files</td>
<td>96</td>
</tr>
<tr>
<td>17.1.2 Generate Presigned URL for zip file</td>
<td>98</td>
</tr>
<tr>
<td>18. Notification API</td>
<td>101</td>
</tr>
<tr>
<td>19. Status API</td>
<td>104</td>
</tr>
<tr>
<td>19.1 ICDT Request/Response Business Process Flow</td>
<td>106</td>
</tr>
<tr>
<td>19.2 Service Registration Processing Overview</td>
<td>108</td>
</tr>
<tr>
<td>19.3 Document Codes Processing Overview</td>
<td>110</td>
</tr>
<tr>
<td>20. Schema Definition and Sample Files</td>
<td>111</td>
</tr>
<tr>
<td>20.1 DCF File Format</td>
<td>111</td>
</tr>
<tr>
<td>20.2 eMDR (Pre-Pay/Post-Pay) Processing Overview</td>
<td>111</td>
</tr>
<tr>
<td>20.2.1 Logical Process Flow</td>
<td>112</td>
</tr>
<tr>
<td>20.2.2 eMDR Post-Pay Logical Flow</td>
<td>113</td>
</tr>
<tr>
<td>20.3 Start RC Client</td>
<td>115</td>
</tr>
<tr>
<td>20.3.1 Login and Encryption</td>
<td>115</td>
</tr>
<tr>
<td>20.4 Upload Process</td>
<td>116</td>
</tr>
<tr>
<td>20.4.1 Outbound Start</td>
<td>116</td>
</tr>
<tr>
<td>20.4.2 Get Outbound Documents</td>
<td>116</td>
</tr>
<tr>
<td>20.4.3 Connect</td>
<td>116</td>
</tr>
<tr>
<td>20.4.4 Upload</td>
<td>116</td>
</tr>
<tr>
<td>20.5 Download Process</td>
<td>116</td>
</tr>
<tr>
<td>20.5.1 Inbound Start</td>
<td>116</td>
</tr>
<tr>
<td>20.5.2 Extraction Failure</td>
<td>117</td>
</tr>
<tr>
<td>20.5.3 Extraction</td>
<td>117</td>
</tr>
<tr>
<td>20.5.4 Checksum Verification</td>
<td>117</td>
</tr>
</tbody>
</table>
20.6 Acknowledgements

20.6.1 Pickup Notification

20.6.2 Error Pickup Notification

20.7 Connect

20.8 Get Notifications

20.9 Process Document

20.10 Download Document

21. .NET Client API

21.1 Push Frequency for Outbound Responses

21.2 Security

21.3 .NET API Documentation

21.3.1 Auth

21.3.2 Download

21.3.3 Outbound

21.3.4 PA Error (Rejected Decision) Response

21.3.5 Administrative Error Response to Inbound Submissions

21.3.6 Status

21.3.7 Utilities – Encryption

21.3.8 Advanced / Debugging API

21.3.9 Validation API

22. API Methods

22.1 Unique ID Rules and Format

22.1.1 Unique ID Generation

22.2 ICDT Request

22.3 ICDT Solicited Response

22.4 ICDT Unsolicited Response

22.5 ICDT Administrative Error Response

22.6 Pre-Pay and Post-Pay eMDR Letters
22.7 Logs ................................................................................................................ 137

23. Error Codes ........................................................................................................ 138
   23.1 Errors: esMD to RC ................................................................................ 138
   23.2 Errors: RC to esMD ............................................................................. 146
      23.2.1 Administrative Errors: ............................................................. 147
      23.2.2 Pickup Errors: .......................................................................... 148

24. PA Requests and Responses Automation with Shared Systems .................. 149
   24.1 Introduction ............................................................................................... 149
      24.1.1 Overview of the Automation Process ........................................ 149
      24.1.2 Shared Systems: ....................................................................... 149
   24.2 Assumptions ............................................................................................... 149
   24.3 Automation of PA Requests/Responses – Application Workflow .......... 150
      24.3.1 Logical Workflow: ...................................................................... 150
      24.3.2 Application Workflow: ............................................................ 150

25. esMD Cloud File Transfers ............................................................................... 152

26. Inbound/Outbound File Names and Data Directories ...................................... 153

27. Contacts............................................................................................................. 159

Appendix A: Description of Fields on RC Client Tabs ..................................... 160
Appendix B: Reject Error Codes ......................................................................... 162
Appendix C: Industry Codes .................................................................................. 163
Appendix D: Content Type Codes ....................................................................... 164
Appendix E: Acronyms ........................................................................................ 166
Appendix F: Glossary ............................................................................................. 168
Appendix G: Frequently Asked Questions (FAQs) ........................................... 169
Appendix H: Approvals .......................................................................................... 170
List of Figures

Figure 1: RC Client Inbound and Outbound Process ................................................................. 9
Figure 2: RC Client ICDT Folder Structure .............................................................................. 12
Figure 3: Review Response To PA Request ............................................................................ 52
Figure 4: RC Client Components .......................................................................................... 84
Figure 5: RC Client APIs implemented in the esMD Cloud .................................................... 85
Figure 6: RC Client Workflow ............................................................................................... 88
Figure 7: esMD Auth API Sequence Diagram ........................................................................ 89
Figure 8: esMD Upload API Sequence Diagram ..................................................................... 91
Figure 9: esMD Download API Sequence Diagram ................................................................. 96
Figure 10: esMD GetListOfFiles Sequence Diagram ............................................................... 97
Figure 11: esMD Generate PresignedURL Sequence Diagram ............................................. 99
Figure 12: Figure 9: esMD Notification API Sequence Diagram .......................................... 101
Figure 13: esMD Status API Sequence Diagram ................................................................... 104
Figure 14: Request/Response Business Process Flow Diagram ............................................ 107
Figure 15: Service Registration Process Flow ....................................................................... 109
Figure 16: Document code file (DCF) Flow ......................................................................... 111
Figure 17: eMDR Pre-Pay Process Flow ................................................................................. 112
Figure 18: eMDR Post-Pay Process Flow ............................................................................... 114
Figure 19: Encryption and Decryption Process ..................................................................... 119
Figure 20: High-level ICDT API Architecture ...................................................................... 134
Figure 21: esMD Shared System/Workload Integration – Logical ........................................ 150
Figure 22: Information Flow – X12N 278 PA Request/Response Integration with Shared Systems .... 151
# List of Tables

Table 1: Record of Changes......................................................................................................................................... ii
Table 2: Medicare Contractors, Responsibilities and Contact Information................................................................. 2
Table 3: Libraries ............................................................................................................................................................ 25
Table 4: Sample RC Client Configuration File ........................................................................................................... 28
Table 5: Sample RC Client API Properties File ........................................................................................................ 30
Table 6: Inbound and Outbound File Formats ........................................................................................................... 33
Table 7: E_L13_BGR000007095735_metadata.xml................................................................................................... 55
Table 8: N_L8_1_KBW000000006908_Delivery_Acknowledgement.xml............................................................. 57
Table 9: F_L13_PDW000000007903_Pickup_V alidation_Error.xml ........................................................................ 57
Table 10: N_L1_IUC000000006217_Delivery_Acknowledgement.xml ................................................................. 58
Table 11: F_ADMIN_123456788912345_Validation_Error.xml ............................................................................. 58
Table 12: Y_L151Q5DNES9996D030819T1405310<<RandomNo>>_Virus_Scan_Gateway_F ailure.xml..................... 59
Table 13: X_L151Q5DNES9996D030819T1405310_Virus_Scan_Error.xml ......................................................... 60
Table 14: N_L<ContentTypeCode>_ <Transaction ID>_Delivery_Acknowledgement.xml ........................................ 61
Table 15: F_PA_321312313112312_Review_Response_V alidation_Error.xml ......................................................... 61
Table 16: Q_L1518DMESD0020315191038490_ICDTSolicitedRequest.xml ....................................................... 62
Table 17: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml ..................................................... 63
Table 18: R_L153PQQES99960308191418450_ICDTUnSolicitedResponse.xml ................................................... 64
Table 19: ESD002.T.ICDT.BWBT0002051601EC.ESMD2.D031519.T1201000 ...................................................... 66
Table 20: V_L153RLELES002D031519T1040260_V alidation_Error.xml ........................................................ 67
Table 21: ESD002.T.ADM.CWBT0002051601EC.ESMD2.D031919.T1334570 .................................................... 68
Table 22: esMD Validation Error Response for Pre-Pay eMDR Letters ............................................................. 68
Table 23: esMD Validation Error Response for Post Pay eMDR Letters ......................................................... 69
Table 24: Document Code File .................................................................................................................... 70
Table 25: Table 4: P_L1_DLL00000009524_Pickup.xml ............................................................................ 71
Table 26: P_L1_WBT0002051601EC_Pickup_Error_Notification.xml ......................................................... 72
Table 27: R_PA_LQA00000006905_Response.xml ..................................................................................... 72
Table 28: D_ADM_EPP00000008983_AdminResponse.xml ........................................................................ 73
Table 29: Q_L1518DMESD0020315191038490_ICDTSolicitedRequest.xml .................................................. 74
Table 30: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml ............................................... 75
Table 31: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml ............................................... 76
Table 32: T.ESD002.ICDT.BJ99.ESMD2.D031519.T1201000 .................................................................... 77
Table 33: ESMD2.T.ADM.CWBT0002051601EC.ESD002.D031919.T1436180 ........................................... 78
Table 34: P_L5_WBT0002051601EC_Pickup.xml ...................................................................................... 79
Table 35: Sample eMDRProcessMetadata XML ....................................................................................... 79
Table 36: U_L16Y5XESD0020726192103460_eMDRStructuredFile.xml .................................................... 80
Table 37: RC Client Error Codes and Error Messages .................................................................................. 81
Table 38: P_WBT0002051601EC_3C8_Pickup.xml .................................................................................... 82
Table 39: DCF Error Pickup Notification .................................................................................................. 83
Table 40: HOPD Pickup Notification ........................................................................................................ 83
Table 41: ICDT Request/Response Business Process Flow Steps ............................................................... 107
Table 42: Service Registration Flow Steps ................................................................................................ 109
Table 43: Document Code File Process Flow Steps .................................................................................. 111
Table 44: eMDR Pre-Pay Logical Process Flow Steps .............................................................................. 113
Table 45: eMDR Post-Pay Logical Process Flow Steps ............................................................................. 114
Table 46: The esMD.RC Client.Login.LoginProcess Methods .................................................................. 120
Table 47: The esMDRC Client.Inbound.Inbound Methods ....................................................................... 121
Table 48: The esMD.RC Client.Outbound.Outbound Methods ................................................................. 123
Table 49: Manual Submission of PA Error (Rejected Decision) Response .............................................. 124
Table 50: Administrative Error Response to Inbound Submissions .......................................................... 124
Table 51: Status messages from esMD .................................................................................................... 125
Table 52: EMSD.RC Client.Encryption.EncryptionUtil Methods ............................................................... 126
Table 53: Remote Troubleshooting ........................................................................................................ 126
Table 54: Validation Methods ................................................................................................................ 127
Table 55: Example Unique ID Rules and Format ..................................................................................... 128
Table 56: Unique ID Generation API Methods ......................................................................................... 128
Table 57: ICDT Request API Methods ..................................................................................................... 129
Table 58: ICDT Solicited Response API Methods ..................................................................................... 130
Table 59: ICDT Unsolicited Response API Methods ................................................................................ 132
Table 60: Administrative Error Response API Methods ......................................................................... 135
Table 61: Pre-Pay and Post-Pay API Methods ........................................................................................ 136
Table 62: Error Codes Sent from the esMD to RC .................................................................................. 138
Table 63: Administrative Error Codes .................................................................................................. 147
Table 64: Pickup Error Codes ................................................................................................................ 148
Table 65: Inbound/Outbound File Names and Data Directories ............................................................... 153
Table 66: Support Points of Contact ...................................................................................................... 159
Table 67: Descriptions of Fields on Error Response to PA Request Tab ................................................ 160
Table 68: Descriptions of Fields on Administrative Error Response to Inbound Submissions Tab ...... 160
Table 69: Descriptions of Fields on Advanced/Debugging Tab ............................................................... 161
Table 70: Content Type Code Descriptions ........................................................................................... 164
Table 71: Content Type Codes and Business Types ............................................................................... 164
Table 72: Acronyms ............................................................................................................................. 166
Table 73: Glossary ............................................................................................................................... 168
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 of 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 use gateway services to send responses for requests for additional documentation electronically to a 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

<table>
<thead>
<tr>
<th>Type of Contractor</th>
<th>Responsibilities</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medicare Administrative Contractors (MAC)</td>
<td>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.</td>
<td><a href="http://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/Review-Contractor-Directory-Interactive-Map">http://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/Review-Contractor-Directory-Interactive-Map</a></td>
</tr>
<tr>
<td>Supplemental Medical Review Contractor (SMRC)</td>
<td>Conduct nationwide medical review, as directed by CMS. This includes identifying underpayments and overpayments.</td>
<td><a href="http://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/Medical-Review/SMRC.html">http://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/Medical-Review/SMRC.html</a></td>
</tr>
<tr>
<td>Contractor (CERT DC), CERT Review Contractor (CERT RC), and CERT Statistical Contractor (CERT SC)</td>
<td>Collect documentation and perform reviews on a statistically valid random sample of Medicare FFS claims to produce an annual improper payment rate.</td>
<td><a href="https://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/CERT/index.html?redirect=/cert">https://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicare-FFS-Compliance-Programs/CERT/index.html?redirect=/cert</a></td>
</tr>
<tr>
<td>Qualified Independent Contractor (QIC)</td>
<td>A party to the redetermination may request a reconsideration if dissatisfied with the redetermination decision. A QIC conducts the reconsideration.</td>
<td><a href="https://www.cms.gov/medicare/appeals-and-grievances/orgmedffsappeals/reconsiderationbyaquallifiedindependentcontractor.html">https://www.cms.gov/medicare/appeals-and-grievances/orgmedffsappeals/reconsiderationbyaquallifiedindependentcontractor.html</a></td>
</tr>
</tbody>
</table>

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/X12N 275 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 using Upload, Download, Notification, and Status API's in the esMD Cloud environment.

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

### 1.3 System Requirements

See Section 7, System Requirements for the system requirements for installing a Microsoft .NET 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 .NET Framework version needed to run the RC Client properly.

Refer to the Identity Management (IDM) Instructions in the link below on how to obtain an IDM login:


Refer to Section 1.4.3 RC Client Operation Overview for Enterprise File Transfer (EFT) Password requirements as per IDM policy for logging in to Internal Server.

### 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, the CMS expanded the CMS esMD Gateway to allow Durable Medical Equipment (DME) suppliers and providers to send electronic Prior Authorization (PA) Requests to Medicare RCs.

In June 2013, the 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 decisions on PA requests.
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 Durable Medical Equipment, Prosthetics, Orthotics and Supplies (DMEPOS) PA requests and Pre-Claim Review Demonstration for Home Health Services Pre-Claim Review (HHPCR) requests as X12N transactions, and to send Review Result Responses for these programs. The “RC Client” also receives 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 (ParentUniqueld and SplitNumber value in optional metadata element tags) in the RC Metadata Extensible Markup Language (XML) for matching/grouping the split payloads submitted by 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 Seven International (HL7) standard, Consolidated Clinical Document Architecture (C-CDA) from the HIHs. The structured medical documentation is sent to the “RC Client” application as an XML format in the RC Package in addition to the existing Portable Document Format (PDF) format of Payload files. The MFT folders for the Review Contractors are moved from the Baltimore Data Center (BDC) to the Virtual Data Center (VDC).

In April 2017, the “RC Client” Application Programming Interface (API) 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 the functionality to receive the DMEPOS PA program requests 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 continued to support existing functionality and Lines of Business (LOB) while adding the following new capabilities. The RC Client application began sending Additional Documentation Request (ADR) Review Results Letters (RRL) electronically through esMD system using Content Type Code 1.3. Updates were made to the existing ADR Review Responses functionality. HIHs began sending Structured Documentation in 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 capability:

- Unsolicited Paperwork (PWK) Claim Documentation: The RC Client application receives Unsolicited PWK claim documentation in XDR format using Content Type Code 7.

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

1. 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.
2. 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 API are updated to support the new format of the esMD Transaction ID. Refer to section 21 .NET 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 must 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.

Note: The latest version of the .NET RC Client released for the July 2019 release is v6.1. All the RCs using the .NET version of the RC Client must use v6.1.

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 Content Type Code 1.5.
2. The RC Client application will begin sending Post-Pay eMDR ADR letters electronically through the esMD system using Content Type Code 1.6.
esMD release AR2020.07.0, in July 2020, supported existing functionality and LOBs while adding the following new capabilities:

4. The RC Client application began receiving Document Codes flat files from the esMD system using Content Type Code 17.

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

esMD release AR2020.11.0, in November 2020, supported existing functionality and LOBs while adding the following capabilities:

23. The RC client application began receiving X12 transaction packages without FFRs and the coversheet from the esMD system using Content Type Code 13.

esMD release AR2021.04.0 in April 2021, did not have any changes to the RC Client API. The document was updated for the EIDM to IDM migration.

esMD release AR2021.10.0, which took place in October 2021, continued to support existing functionality and LOBs, while adding the following new capabilities:

1. The RC Client application began receiving the RC package for Hospital OutPatient Department (HOPD) in X12N 278 format.

2. The RC Client application received specific error messages for required elements when there are validation errors in the Electronic Medical Documentation Request (eMDR) Post-Pay structured XML.

3. The message on the RC Client login screen was updated based on the connectivity to the Managed File Transfer (MFT) server.

4. The RC Client application began receiving the RC Metadata XML file with specific PA program content type along with the workload number.

esMD release AR2022.01.0, scheduled for January 2022, continued to support existing functionality and LOBs, while adding the following new capabilities:

1. The following X12N 278 transaction error category, ‘Medical Info’ error codes, can be sent to the HIH through the RC Client application for supporting documentation either in the XDR (CTC=13) or the X12N 275 format:
   - ‘15 – Number of Units is missing or invalid’ and/or
   - ‘57 – Proposed Date/Date Range’.

2. For the X12N 278 reject response sent through the RC Client, an HIH delivery notification will be sent to the RC Client, once the response is received by the HIH.

For the April 2022 release (AR2022.04.0), the following new capabilities were added:
1. When the RC rejects an XDR PA request for an invalid Procedure Code with error code “AG: Procedure Code(s) is Invalid”, the RC will also include one or more Procedure Code(s).
2. The RC Client UI screens and API were updated to allow the RCs to submit the error response without a Reason Code.

esMD release AR2023.01.0, in January 2023, continued to support existing functionality and LOBs, while adding the following new enhancements as part of the esMD cloud migration:

1. Replaced the existing file transfer mechanism (Tibco MFT) with esMD Upload, Download, Notification, and Status API’s to transfer the files in the esMD cloud environment.
2. The RC Clients login with an IDM username and password and the user credentials are authenticated by the esMD Authentication API.
3. The Auth API is implemented to authenticate the user with IDM credentials in the esMD cloud environment.
4. The Notification API is implemented to send notifications (Admin Error, Pickup, PA Reject Response) to the HIH and receive the responses back in real time.
5. The Status API is implemented to run as a scheduled service to pull the delivery confirmation errors or any validation failures from esMD at specific intervals of time.

1.4.1 RC Client Upload/Download Functionality
The RC Client provides the following functionality:

- **Download:**
  - Inbound documents (submitted by HIHs) from the esMD Cloud Environment.
  - HIH acknowledgements indicating receipt of pick-up notifications, PA review result responses, Administrative Error Response HIH delivery notification.
  - 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.
- eMDR Service Registration Request.
- Document Codes Request.

- Upload:
  - 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 responses 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.
    - Folder locations for both Inbound and Outbound files.
  - ICDT to esMD:
    - ICDT Request.
    - ICDT Solicited Response.
    - ICDT Unsolicited Response.
  - ICDT Pickup Notifications/Error Notifications to esMD.
  - ICDT Administrative Error Response to esMD.
  - eMDR Service Registration Request Pickup Notification.
  - Document Codes Request Pickup Notification.

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 esMD cloud environment using the esMD Auth API to upload and download the files. The RC Java Client uses IDM System login credentials, and the user is authenticated using the esMD Auth API. 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 upload and download files automatically without continual user intervention, based on the time intervals set by the RC. The user is authenticated every
time the inbound and outbound process starts and also when the files are actually uploaded and downloaded.

**Figure 1: RC Client Inbound and Outbound Process**

In the Inbound process, the RC Client connects to esMD cloud environment using the Auth API and executes the Download API process to get a list of available files using the token returned by the Auth API. The download process then iterates every file from the list and invokes the esMD Download API again with the file name and token to get the presigned URL. The file object is then downloaded from the esMD cloud using the presigned URL and the token. The documents are pulled into the RC’s inbound user directory for the authenticated user and waits for the next cycle, as determined by the Inbound Pull Time Interval setting.

In the Outbound process the RC Client connects to the esMD cloud environment using the Auth API and executes the Upload API process to upload the files to the esMD cloud by using the presigned URL and token returned by the esMD Auth API. The RC Client 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.

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

The RC does not need to login to the esMD cloud environment to create PA Reject Error Responses and Administrative Error Responses. The login is necessary only to download or upload files from or to esMD cloud. The Notification API is used to process the PA Reject and Administrative Errors which internally uses the encrypted user credentials stored in memory and invokes the Auth API to get the user authenticated.
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)*
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 esMD cloud environment. 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:


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

PASSWORD POLICY

1. Passwords must be at least 8 characters in length.  
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.

---

1 HARP user passwords must currently contain a minimum of 12 characters.
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 has been received from esMD, all the processes of the RC Client are suspended and the RC Client is terminated. In addition, the RC Client is locked and cannot upload/download files even if the RC Client is restarted. The RC is advised to contact the esMD Service Desk (refer to Section 27 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 two different type of ICDT Request and ICDT Solicited/Unsolicited Response as part of the initial pilot program:

1. **ICDT Request/Solicited Response**: RC-A sends an ICDT Request to RC-B requesting certain documentation of a claim or a case and RC-B responds (ICDT Solicited Response) to RC-A with the requested attachments.

2. **ICDT Unsolicited Response**: RC-A sends (ICDT Unsolicited Response) documentation bundle to another RC (RC-B, e.g., misdirected documentation).

#### 1.5.1 RC Client ICDT Folder Structure

The separate folder structure is used for placing the ICDT Request/Solicited Response, Unsolicited Response, Notifications, errors and Acknowledgments files. The folder ‘icdt’ is created under the ‘data’ folder as shown in Figure 2: RC Client ICDT Folder Structure. The ‘icdt’ contains three folders. The ICDT Request and ICDT Response files are moved to ‘input’ folder. All the notifications and acknowledgments are placed in ‘ntfn_ack’ folder. Any validations errors and admin errors received from the esMD system are moved to ‘error’ folder.
As part of the October 2021 release, the file pattern issue was resolved to allow all downloaded ICDT files to be navigated to their respective ICDT directories, instead of placing the files in “download” or “temp” folders.
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:
   a. How to start and log into the RC Client.
   b. How to enter a Review Response decision.
   c. How to enter an error code for a PA request.
   d. How to submit Inbound Submissions errors.
   e. Advanced debugging, which shows how to test to see if your RC Client application can connect to the esMD Cloud environment 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 Microsoft .NET 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:

   a. Overview of the installation process.
   b. Systems Requirements for a Microsoft .NET installation.
   c. Installing an Out-of-Box .NET version of the RC Client application.
   d. File transfers from esMD Cloud environment.
   e. XML/JSON Messages, including Outbound, Inbound, and Error messages.
   f. Inbound Processes and Files.
   g. Outbound Processes and Files.
   h. Configuring the RC Client application.
   i. RC Client Components.
   j. RC Client Workflow.
k. RC Client application Utilities, API’s, Components, Schedulers, and Encryption.

l. Pilot Programs API Methods.

m. Configuring the RC Client application for notifications.

n. Processing and pulling in documents.

o. 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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1.</td>
<td>Starting the RC Client and Logging In</td>
</tr>
<tr>
<td></td>
<td>Start the RC Client by selecting the RCClientUI-V9.0.0.exe in the RC Installation folder or directory.</td>
</tr>
<tr>
<td>Step 2.</td>
<td>Starting the RC Client and Logging In</td>
</tr>
<tr>
<td></td>
<td>The Login screen is displayed.</td>
</tr>
<tr>
<td></td>
<td>Enter your IDM User ID and password and select Login and Run RC Client. All uppercase capital letters should be used for the IDM user name.</td>
</tr>
</tbody>
</table>

 Velvet 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 here: https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/EnterpriseIdentityManagement/Guides-and-Documentation). A document that provides additional IDM account information specific to esMD users will be provided in the near future.
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1.</td>
<td>Entering an Error Code</td>
<td>Select the Error Response to PA Request tab.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>❖ After a successful log in, another log in is not required to navigate to and use the Error Response to PA Request tab.</td>
</tr>
</tbody>
</table>

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 3. Entering an Error Code

Select a Reject Error Code and the Add button 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 4. Entering an Error Code

Enter the Reason Code(s). Select the Add button at the end of the row of the Reason Code(s) field to add additional rows of Reason
Enter the Request Level UTN and select Save to submit the Error Code for submission.

For information on how to access an up-to-date list of Reason Codes, see Appendix A: Description of Fields on RC Client Tabs.

---

**Technical Note:** After selecting Save, the Notification API creates the PA Reject Response JSON message and sends it to the esMD system after authenticating the user using the Auth API. The PA Reject Response is delivered to the HIH and the delivery confirmation in JSON format is returned to the RC Client. The notification API process converts the JSON message to an XML file and saves it in the notification folder.

---

**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 error 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 is cleared, and another response can be entered.
As part of the October 2021 release, the Error Response to PA Request screen on the RC Client user interface (UI) was updated to add a new Reject Error Code, which allows RCs to reject PA requests for Provider Exemption. The new Reject Error Code was added under the Requester Category of the Reject Error Category. In addition, esMD started accepting the GEX19 Reason Code from the RCs as part of the Reject responses.

As part of the January 2022 release, for the reject error category ‘Medical Info’, the following reject error codes will be sent to the HIH through the RC Client in the X12N 278 response:

- ‘15 – Number of Units is missing or invalid’.
- ‘57 – Proposed Date/Date Range’.
How to Enter an Error Code on the Error Response to PA Request Tab

Error Response to PA Request

- Transaction ID: ERTYRTY432156
- Reject Error Category: Requester
- Reject Error Code(s):
  - An address component is missing or invalid
  - First and/or Last name is/are missing
  - Not a pilot participant State
  - NPI does not match the Name of the Physician
  - NPI is missing or invalid
  - Provider is exempted from submitting this PA request

Reason Code(s): GEX19

Add Remove
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| **Step 1.** Entering an Inbound Submission 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. |

![Administrative Error Response to Inbound Submissions Tab](image)

**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.
Step 3. Entering an Inbound Submission Error

After selecting Save, the “The Administrative Error Response to Inbound Submission has been successfully Saved for Submission” message is displayed.

Technical Note: After selecting Save, the Notification API invokes the Auth API to authenticate the user using the encrypted user credentials stored in memory. After authenticating the user, an Admin Error JSON message is created and sent to esMD. The Admin Error is delivered to HIH and the delivery confirmation in JSON format is returned to the RC Client. The notification API process converts the JSON message to XML file and saves in the notification folder.

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 esMD Cloud, Using the Advanced/Debugging Tab

This section provides step-by-step instructions on how to verify connection to the esMD Cloud, using the Advanced/Debugging tab.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1.</td>
<td><strong>Checking Connection to esMD Cloud</strong></td>
</tr>
<tr>
<td></td>
<td>Select the Advanced/Debugging tab.</td>
</tr>
<tr>
<td></td>
<td>The Advanced/Debugging tab fields are displayed.</td>
</tr>
<tr>
<td></td>
<td>On the Advanced/Debugging tab, enter your IDM User ID and password. (This is required on the Advanced/Debugging tab.)</td>
</tr>
<tr>
<td></td>
<td>Select Test Connection.</td>
</tr>
</tbody>
</table>

After selecting Test Connection, the “Connectivity Successful. 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 Microsoft .NET 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 .NET Client is 50 Megabytes (MB) for the RC Client itself. The documents that the RC Client pulls from the esMDCloud server may require additional disk space.

7.3 Memory

The RC .NET Client requires a minimum of 128 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 Microsoft .NET Framework

The RC .NET Client requires Microsoft .NET Framework 4.5 to run properly.

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>
<thead>
<tr>
<th>Library</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common.Logging.dll</td>
<td>2.1.1</td>
<td>Logging Framework</td>
</tr>
<tr>
<td>Common.Logging.Log4Net1211.dll</td>
<td>2.1.1</td>
<td>Logging Framework</td>
</tr>
<tr>
<td>edtFTPnetPRO.dll</td>
<td>8.6.4</td>
<td>the Secure Shell (SSH) File Transfer Protocol (SFTP) Library</td>
</tr>
<tr>
<td>Library</td>
<td>Version</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Ionic.Zip.dll</td>
<td>1.9.1.8</td>
<td>Compression Library</td>
</tr>
<tr>
<td>log4net.dll</td>
<td>1.2.11.0</td>
<td>Logging Framework</td>
</tr>
</tbody>
</table>
8. How to Install and Configure a Microsoft .NET 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 (.NET).

8.1 Out-of-the-Box

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

8.1.1 Keystore Set Up

Important: The RC .NET Client uses asymmetric encryption to store the IDM user credentials securely. For this encryption to work, the RC must use the machine-level Rivest, Shamir & Adleman (RSA) key container provided by Microsoft Windows. Please refer to Section 21.2 Security for more details on the Security framework used by the RC Client.

8.1.1.1 Microsoft Windows Machine-Level RSA Key Container

Microsoft Windows provides machine-level RSA key containers to all users who can log in to a computer by default. RSA key containers are used to encrypt or decrypt protected configuration sections while logged in with an administrator account. You can use a machine-level RSA key container to protect information for a single application, all the applications on a server, or a group of applications on a server that runs under the same user identity. Although machine-level RSA key containers are available to all users, they can be secured with New Technology File System (NTFS) Access Control Lists (ACL) so that only required users can access them. You can use the aspnet_regiis.exe tool to create, export, import, or delete an RSA key container:

- Type the command below at a command console to create a new RSA key container.

  Cd C:\Windows\Microsoft.NET\Framework64\<v4.xxxxxxxx>

  aspnet_regiis –pc <yourKeyName>
Note: Replace <v4.xxxxxxx> with the actual .NET framework version on your machine, and the <yourKeyName> with a name for your key so that you can retrieve it later.

8.1.1.2 Key Handling
The RC .NET Client delegates the key handling to the Windows Operating System environment.

8.1.2 Configuring the RC Client
Once the keystore is created, the RC Client is ready to be configured to use the keystore.

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

   Note: The certAlias name in the configuration file should be unique for each instance running on the same machine to avoid any encryption errors.

   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. The ESMDSFTPServer section in the config file is replaced with the ESMDAuthAPI configuration which is used by the esMD Auth API to connect to esMD Cloud.

   Note: Contact Helpdesk to get the ClientID, ClientSecret, hostname used by the API’s.

Use the comments for each configuration parameter shown in Table 4: Sample RC Client Configuration File as a guide for entering your data.

Table 4: Sample RC Client Configuration File

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://esmd.ois.cms.hhs.gov/v1/rc/config esmd-config.xsd">
    <!--The Authentication Server Configuration-->
    <ESMDAuthAPI>
        <!-- Update: Use T for VAL, P for PROD-->
        <environment>T</environment>
        <!--Client ID to connect to Auth server-->
        <!-- Update: The Client ID-->
        <ClientID>3fis5e1mln49c3fk1kfs5v20ml</ClientID>
        <!--Client Secret to connect to Auth Server -- >
        <!-- Update: The Client Secret-->
```
<clientSecret>1ad4odli16jur6qfbaoa7lo52rt8</clientSecret>
</ESMDAuthAPI>

<!--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>ItwdafsdviaZnpvV54aRM9ZzQiw==</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 esMD Cloud-->
  <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 esMD Cloud. IMPORTANT: Replace ES#### with your own mailbox number-->
  <remoteInboundDir>/ES####</remoteInboundDir>
</InboundConfig>
3. The api.properties file contains all the information used by API’s which is required to connect to esMD cloud environment.

Table 5: Sample RC Client API Properties File

```
DOWNLOAD_SCOPE=devesmdrc/download
UPLOAD_SCOPE=devesmdrc/upload
NOTIFICATION_SCOPE=devesmdrc/notification
STATUS_SCOPE=devesmdrc/status

# Update env value (use dev for DEV, val for VAL, uat for UAT and prod for PROD)
API_BASE_URL=https://esmdc{env}.cms.hhs.gov/esmd/v1

AUTH_URL=/auth
UPLOAD_URL=/objects
DOWNLOAD_URL=/objects
NOTIFICATION_URL=/objects/notification
```
8.1.3 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 4: Sample RC Client Configuration File.

1. To run the sample RC Client, run the "RcClientUI.exe" utility provided in the distribution package.
2. Start the RC Client by providing the IDM login credentials (i.e., IDM User ID and password) for the Login tab and select the “Login and Run RC Client” button.

8.2 Custom RC Client

The RC .NET 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 esMD Cloud environment using esMD Auth API.
2. Get Notifications using esMD Notifications API (refer to Section 21.3.2 Download).
3. Decrypt/encrypt and store the login credentials using a secure RSA algorithm (refer to Section 21.3.7 Utilities – Encryption).
4. Pull medical documentation from the esMD Cloud (refer to Section 21.3.2 Download).
5. Extract the downloaded packages (refer to Section 21.3.2 Download).
6. Check the payloads using checksums in the metadata (refer to Section 21.3.2 Download).
7. Push the outbound files from the output directory (refer to Section 21.3.3 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 documentation needed for integrating the API.)
9. esMD Cloud File Transfers

Table 6: Inbound and Outbound File Formats provides example file names and descriptions of Inbound and Outbound transactions.

1. ES0001 is a sample mailbox number that is used to identify the RC and YSQ0002051701EC is a sample fifteen-character esMD transaction ID.
2. The esMD transaction ID is included in the zip file name and will also be included in the RC metadata XML file.
### Table 6: Inbound and Outbound File Formats

<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Inbound    | `<ReceiverRoutingId>>.L<<CTC>>.E<<esMDTransactionID>>.<<SenderRoutingID>>.DMMddyy.THHmmssS` | Submissions received from esMD to the RC:  
1. `T` – Environment ID. T is for User Acceptance Testing (UAT) and P is for Production (PROD);  
2. `<ReceiverRoutingId>> – RC Routing ID;  
3. L – The Line of Business;  
4. `<CTC>> – Content Type Code (CTC) of the program;  
5. E – Delivery type of the inbound request;  
6. `<esMD TransactionID>> – 15-character esMD Transaction ID;  
7. `<SenderRoutingID>> – Sender Routing ID which is ESMD2;  
8. DMMddyy – Date format in MMDDYY.  
9. THHmmssS – Time format in THHmmssS |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Inbound   | <<ReceiverRoutingId>>.T.L<<CTC>>.A<<esMD TransactionID>>.<SenderRoutingID>>.DMMddyy.THHmmssS | Acknowledgments received from esMD to the RC:  
1. T – Environment ID. T is for UAT and P is for PROD;  
2. <<ReceiverRoutingId>> – RC Routing ID;  
3. L – The Line of Business;  
4. <<CTC>> – CTC of the program;  
5. A – Delivery type of the Acknowledgments;  
6. <<esMD TransactionID>> – 15-character esMD Transaction ID;  
7. <<SenderRoutingID>> – Sender Routing ID which is ESMD2;  
8. DMMddyy – Date format in MMDDYY.  
9. THHmmssS – Time format in THHmmssS. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound</td>
<td>&lt;&lt;ReceiverRoutingId&gt;&gt;.T.L&lt;&lt;CTC&gt;&gt;.N&lt;&lt;esMD TransactionID&gt;&gt;.&lt;&lt;SenderRoutingID&gt;&gt;.DMMddyy.THHmmssS</td>
<td>HIH delivery notification from esMD to RC:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. T – Environment ID. T is for UAT and P is for PROD;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. &lt;&lt;ReceiverRoutingId&gt;&gt; – RC Routing ID;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. L – The Line of Business;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. &lt;&lt;CTC&gt;&gt; – CTC of the program;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. N – Delivery type of the HIH Notifications;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. &lt;&lt;esMD TransactionID&gt;&gt; – 15-character esMD TransactionID;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7. &lt;&lt;SenderRoutingID&gt;&gt; – Sender Routing ID which is ESMD2;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8. DMMddyy – Date format in MMDDYY.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9. THHmmssS – Time format in THHmmssS.</td>
</tr>
<tr>
<td>Type</td>
<td>Example File Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Inbound | `<<ReceiverRoutingId>>.T.L<<CTC>>.F<<esMD TransactionID>>.<<SenderRoutingID>>.DMMddyy.THHmmssS` | Any validation failures from esMD:  
1. T – Environment ID. T is for UAT and P is for PROD;  
2. `<<ReceiverRoutingId>>` – RC Routing ID;  
3. L – The Line of Business;  
4. `<<CTC>>` – CTC of the program;  
5. F – Delivery type of the esMD validation failures;  
6. `<<esMD TransactionID>>` – 15-character esMD TransactionID;  
7. `<<SenderRoutingID>>` – Sender Routing ID which is ESMD2;  
8. DMMddyy – Date format in MMDDYY.  
9. THHmmssS – Time format in THHmmssS. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound</td>
<td>&lt;&lt;ReceiverRoutingId&gt;&gt;.T.ADM.D&lt;&lt;esMD TransactionID&gt;&gt;.&lt;&lt;SenderRoutingID&gt;&gt;.DMMddyy.THHmmssS</td>
<td>Administrative Error Response from RC to esMD to RC:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. T – Environment ID. T is for UAT and P is for PROD;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. &lt;&lt;ReceiverRoutingId&gt;&gt; – RC Routing ID;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. ADM – Administrative error response;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. D – Delivery type of the Administrative error response;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. &lt;&lt;esMD TransactionID&gt;&gt; – 15-character esMD TransactionID;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. &lt;&lt;SenderRoutingID&gt;&gt; – Sender Routing ID which is ESMD2;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7. DMMddyy – Date format in MMDDYY.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8. THHmmssS – Time format in THHmmssS.</td>
</tr>
<tr>
<td>Type</td>
<td>Example File Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Inbound</td>
<td>&lt;&lt;ReceiverRoutingId&gt;&gt;.T.MISC.Y&lt;&lt;esMD TransactionID&gt;&gt;.&lt;&lt;SenderRoutingID&gt;&gt;.DMMddyy.THHmmssS</td>
<td>Virus scan failure errors from esMD to RC:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. T – Environment ID. T is for UAT and P is for PROD;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. &lt;&lt;ReceiverRoutingId&gt;&gt; – RC Routing ID;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. MISC – Miscellaneous Type;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Y – Delivery type of the virus scan failures;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. &lt;&lt;esMD TransactionID&gt;&gt; – 15-character esMD TransactionID;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. &lt;&lt;SenderRoutingID&gt;&gt; – Sender Routing ID which is ESMD2;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7. DMMddyy – Date format in MMDDYY.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8. THHmmssS – Time format in THHmmssS.</td>
</tr>
<tr>
<td>Type</td>
<td>Example File Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Inbound    | <<ReceiverRoutingId>>.T.MISC.X<<esMD TransactionID>>.<<SenderIdRoutingID>>.DMMddyy.THHmssS | Virus found or infected file sent by RC to esMD:  
1. T – Environment ID. T is for UAT and P is for PROD;  
2. <<ReceiverRoutingId>> – RC Routing ID;  
3. MISC– Miscellaneous Type;  
4. X – Delivery type of the virus-scan-infected errors;  
5. <<esMD TransactionID>> – 15-character esMD TransactionID;  
6. <<SenderIdRoutingID>> – Sender Routing ID which is ESMD2;  
7. DMMddyy – Date format in MMDDYY.  
8. THHmssS – Time format in THHmssS. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Inbound/ Inbound  | <<ReceiverRoutingId>>.T.L<<CTC>>.Q<<esMD TransactionID>>.<<SenderRoutingID>>.DMMddyy.THHmmssS | ICDT Solicited Request from RC to esMD to RC:  
1. T – Environment ID. T is for UAT and P is for PROD;  
2. <<ReceiverRoutingId>> – RC Routing ID;  
3. L – The Line of Business;  
4. <<CTC>> – CTC of the program;  
5. Q – Delivery type of the ICDT Solicited request;  
6. <<esMD TransactionID>> – 15-character esMD TransactionID;  
7. <<SenderRoutingID>> – Sender Routing ID which is ESMD2;  
8. DMMddyy – Date format in MMDDYY.  
9. THHmmssS – Time format in THHmmssS. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Inbound/ Outbound    | <<ReceiverRoutingId>>.T.L<<CTC>>.R<<esMD TransactionID>>.<<SenderRoutingID>>.DMMddyy.THHmmssS | ICDT Solicited and Unsolicited Response from RC to esMD to RC.  
1. T – Environment ID. T is for UAT and P is for PROD;  
2. <<ReceiverRoutingId>> – RC Routing ID;  
3. L – The Line of Business;  
4. <<CTC>> – CTC of the program;  
5. R – Delivery type of the ICDT Solicited and Unsolicited Response;  
6. <<esMD TransactionID>> – 15-esMD TransactionID;  
7. <<SenderRoutingID>> – Sender Routing ID which is ESMD2;  
8. DMMddyy – Date format in MMDDYY.  
9. THHmmssS – Time format in THHmmssS. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Inbound/ Outbound  | <<ReceiverRoutingId>>.T.ADM.C<<esMD TransactionID>>.<<SenderRoutingID>>.DMMddyy.THHmmSSS | ICDT Administrative error Response from RC to esMD to RC:  
1. T – Environment ID. T is for UAT and P is for PROD;  
2. <<ReceiverRoutingId>> – RC Routing ID;  
3. ADM – Administrative error response;  
4. C – Delivery type of the Administrative error response;  
5. <<esMD TransactionID>> – 15-esMD TransactionID;  
6. <<SenderRoutingID>> – Sender Routing ID which is ESMD2;  
7. DMMddyy – Date format in MMDDYY.  
8. THHmmSSS – Time format in THHmmSSS. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Inbound/ Outbound    | <<ReceiverRoutingId>>.T.ICDT.B<<esMD TransactionID>>.<<SenderRoutingID>>.DMMddyy.THHmmssS | ICDT Pickup Notification from RC to esMD to RC and ICDT acknowledgments from esMD to RC:  
1. T – Environment ID. T is for UAT and P is for PROD;  
2. <<ReceiverRoutingId>> – RC Routing ID;  
3. ICDT – Type of response;  
4. B – Delivery type of the ICDT Pickup Notification;  
5. <<esMD TransactionID>> – 15-character esMD TransactionID;  
6. <<SenderRoutingID>> – Sender Routing ID which is ESMD2;  
7. DMMddyy – Date format in MMDDYY.  
8. THHmmssS – Time format in THHmmssS. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Inbound    | <<ReceiverRoutingId>>.T <<CTC>> .V <<esMD TransactionID>>. <<SenderRoutingID>>.DMMddyy.THHmmssS | esMD validation errors from esMD to RC for ICDT Request or Solicited or Unsolicited Response.  
1. T – Environment ID. T is for UAT and P is for PROD;  
2. <<ReceiverRoutingId>> – RC Routing ID;  
3. L – The Line of Business;  
4. <<CTC>> – CTC of the program;  
5. V – Delivery type of the ICDT Solicited and Unsolicited Response;  
6. <<esMD TransactionID>> – 15-character esMD TransactionID;  
7. <<SenderRoutingID>> – Sender Routing ID which is ESMD2;  
8. DMMddyy – Date format in DMMDDYY.  
9. THHmmssS – Time format in THHmmssS. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound</td>
<td>&lt;&lt;ReceiverRoutingId&gt;&gt;.T.L&lt;&lt;CTC&gt;&gt;.U&lt;&lt;esMD TransactionID&gt;&gt;.&lt;&lt;SenderRoutingID&gt;&gt;.DMMddyy.THHmmssS</td>
<td>Pre-Pay and Post-Pay eMDR ADR letters from RC to esMD:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. T – Environment ID. T is for UAT and P is for PROD;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. &lt;&lt;ReceiverRoutingId&gt;&gt; – RC Routing ID;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. L – The Line of Business;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. &lt;&lt;CTC&gt;&gt; – CTC of the program;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. U – Delivery type;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. &lt;&lt;esMD TransactionID&gt;&gt; – 15-character esMD TransactionID;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7. &lt;&lt;SenderRoutingID&gt;&gt; – Sender Routing ID which is ESMD2;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8. DMMddyy – Date format in MMDDYY.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9. THHmmssS – Time format in THHmmssS.</td>
</tr>
<tr>
<td>Type</td>
<td>Example File Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Inbound</td>
<td>&lt;&lt;ReceiverRoutingId&gt;&gt;.T.L&lt;&lt;CTC&gt;&gt;.E&lt;&lt;esMD TransactionID&gt;&gt;.&lt;&lt;SenderRoutingID&gt;&gt;.DMMddyy.THHmmssS</td>
<td>Service Registration flat file from esMD to RC:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. T – Environment ID. T is for UAT and P is for PROD;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. &lt;&lt;ReceiverRoutingId&gt;&gt; – RC Routing ID;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. L – The Line of Business;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. &lt;&lt;CTC&gt;&gt; – CTC of the program which is “5”;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. E – Delivery type of the Service Registration Request;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. &lt;&lt;esMD TransactionID&gt;&gt; – 15-character esMD TransactionID;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7. &lt;&lt;SenderRoutingID&gt;&gt; – Sender Routing ID which is ESMD2;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8. DMMddyy – Date format in MMDDYY.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9. THHmmssS – Time format in THHmmssS.</td>
</tr>
<tr>
<td>Type</td>
<td>Example File Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>Example File Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Outbound     | `<<ReceiverRoutingId>>T..L<<CTC>>.P<<esMD TransactionID>>.<<SenderRoutingID>>.DMMddyy.THHmmssS` | Pickup notification file name from RC to esMD for the Service Registration:  
1. T – Environment ID. T is for UAT and P is for PROD;  
2. `<<ReceiverRoutingId>>` – RC Routing ID;  
3. L – The Line of Business;  
4. `<<CTC>>` – CTC of the program which is “17”;  
5. P – Delivery type of the Pickup Notification for Service Registration Request;  
6. `<<esMD TransactionID>>` – 15-character esMD TransactionID;  
7. `<<SenderRoutingID>>` – Sender Routing ID which is ESMD2;  
8. DMMddyy – Date format in MMDDYY.  
9. THHmmssS – Time format in THHmmssS. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Inbound  | <<ReceiverRoutingId>>.T.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;  
  - <<esMD TransactionID>> – 15-character esMD TransactionID;  
  - <<SenderRoutingID>> – Sender Routing ID which is ESMD2;  
  - DMMddyy – Date format in MMDDYY.  
  - THHmmssS – Time format in THHmmssS. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Example File Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Outbound      | <<ReceiverRoutingId>>.T.L<<CTC>>.P<<3CharRandom>>.<<SenderRoutingID>>.DMMddyy.TH HmmSSS | 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;  
• <<esMD TransactionID>> – 15-character esMD TransactionID;  
• <<SenderRoutingID>> – Sender Routing ID which is ESMD2;  
• DMMddyy – Date format in MMDDYY.  
• THHmmSS – Time format in THHmmSS. |
10. XML Schema Definitions

The following schema definitions are updated for the new format of the esMD Transaction ID.

1. esMD-businessstypes.xsd;
2. esMD-config.xsd;
3. esMD-rc.xsd;
4. esMDProcessMetadata.xsd;
5. emdr-rcprocessmetadata.xsd.
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 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 continued to send decision indicators A, N, and M to esMD via the Workload Response files.

In the April 2022 release, the RCs will be able to capture the procedure code when the “Procedure code is invalid” reject error code is selected in the Review Decision Response to PA Request tab, as shown in Figure 3: Review Response To PA Request.

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

![Figure 3: Review Response To PA Request](image-url)
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 accept 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 ADR Response/eMDR Request;
11. esMD Validation Error Response for ADR Response/eMDR Request;
12. HIH Delivery Notification Response for 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. Service Registration Request;
22. HIH Delivery Notification for Service Registration Response;
23. esMD Validation Error Response for Service Registration Response;
24. esMD validation Error Response for Prepay eMDR letters;
25. esMD validation Error Response for Post-pay eMDR letters.
12.1.1 Payload Files

The RC Client will receive PDF files as payloads in the inbound documents with delivery type “E”. Example payload file names are E_185457-esmdQSSIVG0407141396893280928-0.pdf or E_185458-esmdVG0407141396893280928-0.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 the 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 7: E_L13_BGR000007095735_metadata.xml.

Note: The metadata file will remain the same for all lines of business

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

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.

As part of the October 2021 release, esMD will include the specific PA program Content Type instead of Content Type 13. In addition, a metadata element will be added to include the Workload Number in the metadata file.

Table 7: E_L13_BGR000007095735_metadata.xml

```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="BGR000007095735" DeliveryType="E"/>
    <SendingOID>urn:oid:123.456.657.126</SendingOID>
    <TargetOID>urn:oid:2.16.840.1.113883.13.34.13.34.110.1.999.1</TargetOID>
    <CompleteSubmission>true</CompleteSubmission>
    <RequestType>X12-XDR</RequestType>
    <SubmissionMetadata>
      <BusinessType>XDR X12</BusinessType>
    </SubmissionMetadata>
  </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 HIH 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 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.

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 8: N_L8_1_KBW000000006908_Delivery_Acknowledgement.xml.

*Note: The HIH Pickup Status Response will remain the same for all lines of business.*
### 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 9: F_L13_PDW000000007903_Pickup_Validation_Error.xml, and sends it to the RC. The RC will correct the pickup notification and resubmits it to esMD.

### Table 9: F_L13_PDW000000007903_Pickup_Validation_Error.xml

```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>Review Contractor Response Transaction ID does not exist</ErrorName>
  <ErrorDescription>esMD validation error. Please correct and resubmit.</ErrorDescription>
  <Status>FAILED</Status>
  <StatusDesc>esMD validation error. Please correct and resubmit.</StatusDesc>
</ErrorInfo>
<fileName>ESMD2.D.L13.PWBT0002051601EC.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 10: N_L1_IUC00000006217_Delivery_Acknowledgement.xml.

**Note: The Administrative Error HIH Status Response will remain the same for all lines of business.**

**Table 10: N_L1_IUC00000006217_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="IUC00000006217" 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 11: F_ADMIN_123456788912345_Validation_Error.xml, and sends it to the RC. The RC will correct the administrative error response and resubmits it.

**Table 11: 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>ESMD2.D.ADM.DWBT0002051601EC.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 12: Y_L151Q5DNES9996D030819T1405310<<RandomNo>>_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 12: Y_L151Q5DNES9996D030819T1405310<<RandomNo>>_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 scanning. If the file is found to be virus infected, the esMD application sends the message detailed in Table 13: X_<<RandomNo>>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 the esMD Service Desk (esMD_Support@cms.hhs.gov). Refer to the code located in Table 13: X_<<RandomNo>>L151Q5DNES9996D030819T1405310_Virus_Scan_Error.xml

Table 13: X_<<RandomNo>>L151Q5DNES9996D030819T1405310_Virus_Scan_Error.xml

```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 Reject response Delivery Acknowledgment for X12N 278 or XDR

When the RC Client sends a PA reject response to esMD for X12N 278 or XDR PA Programs like Repetitive Scheduled Non-Emergent Ambulance Transport (CTC 8.1), HHPCR Requests (CTC 8.3), DME Phone Discussion Requests (CTC 8.4), or HOPD (CTC 8.5), the esMD application processes the file and sends the PA reject response to the HIH. The esMD application submits the PA reject response, HIH Status Response, as detailed in Table 14: N_L<<ContentTypeCode>_<TransactionID>_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 Reject Response Delivery to HIH</statusDescription>
  <ESMDTransaction TransactionId="RAI000007098495" DeliveryType="N"/>
</ns0:SubmitPADeterminationResponseResult>

Table 14: N_L<ContentTypeCode>_<TransactionID>_Delivery_Acknowledgement.xml and sends it to the RC, indicating the reject response was sent to the HIH.

Note: For the HOPD PA program CTC 8.5, the file naming convention would be N_L8_5_RAI000007098495_Delivery_Acknowledgement.xml. For the X12N 278 transaction, the file naming is N_L13_<TransactionID>_Delivery_Acknowledgment.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 Reject Response Delivery to HIH</statusDescription>
  <ESMDTransaction TransactionId="RAI000007098495" DeliveryType="N"/>
</ns0:SubmitPADeterminationResponseResult>

12.1.10 PA Review Response 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 15: F_PA_321312313112312_Review_Response_Validation_Error.xml, and sends it to the RC. The RC will correct the response and resubmit the PA Review Result.

Table 15: F_PA_321312313112312_Review_Response_Validation_Error.xml

<statusDescription>esMD validation error. Please correct and resubmit.</statusDescription>
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 underscore (i.e., “_”).

Table 16: Q_L1518DMESD0020315191038490_ICDTSolicitedRequest.xml shows the XML message generated for an ICDT Request XML from RCs.

Table 16: Q_L1518DMESD0020315191038490_ICDTSolicitedRequest.xml

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
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 CDT 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 17: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml shows the sample XML Message of the ICDT Response sent from the RCs.

Table 17: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml

```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>L1521R7ESD002130191302560</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>
```
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 “R”. The file name of the ICDT Response Document should contain only alphanumeric characters and underscore (i.e., “_”).

Table 18: R_L153PQQES99960308191418450_ICDTUnSolicitedResponse.xml shows the sample XML Message of the ICDT Unsolicited Response sent from the RCs.

```xml
<xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ICDTResponse xmlns="http://cms.hhs.gov/esmd/icdt">
    <receiverOID>urn:oid:2.16.840.1.113883.13.34.13.34.110.1.999.4</receiverOID>
    <receiverID>01232</receiverID>
    <senderOID>urn:oid:2.16.840.1.113883.13.34.13.34.110.1.999.3</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>File Size</FieldName>
            <FieldValue>7945</FieldValue>
        </OptionalMetadata>
    </Documentation>
</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 its notification or acknowledgment for the particular transaction.
The esMD system generates the batch acknowledgement response to the RC as shown in Table 19: ESD002.T.ICDT.BWBT0002051601EC.ESMD2.D031519.T1201000 and sends it to the RC.

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

Table 19: ESD002.T.ICDT.BWBT0002051601EC.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:creationTime>2019-03-15T12:01:00.335-04:00</ns0:creationTime>
      <ns0:fileName>ESMD2.T.L15_1.QWBT0002051601EC.ESD002.D031519.T1038490</ns0:fileName>
      <ns0:Status><ns0:serviceSuccessful>true</ns0:serviceSuccessful></ns0:Status>
    </ns0:ICDTMetaData>
  </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:creationTime>2019-03-15T12:01:00.335-04:00</ns0:creationTime>
      <ns0:fileName>ESMD2.D.L15_1.QWBT0002051601EC.ESD002.D031519.T1038580</ns0:fileName>
      <ns0:Status><ns0:serviceSuccessful>true</ns0:serviceSuccessful></ns0:Status>
    </ns0:ICDTMetaData>
  </ns0:ICDTNotification>

  <ns0:ICDTNotification RequestType="SOLIC_REQ_ACK" id="3">
    <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:creationTime>2019-03-15T12:01:00.335-04:00</ns0:creationTime>
      <ns0:fileName>ESMD2.D.L15_1.QWBT0002051601EC.ESD002.D031519.T1038580</ns0:fileName>
      <ns0:Status><ns0:serviceSuccessful>true</ns0:serviceSuccessful></ns0:Status>
    </ns0:ICDTMetaData>
  </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 20: V_L153RLELESD002D031519T1040260_Validation_Error.xml, and sends it to the RC.

Table 20: V_L153RLELESD002D031519T1040260_Validation_Error.xml

```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>ESMD2.D.L15_3.RWBT0002051601EC.ESD002.D031519.T1040260</ns0:fileName>
        <ns0:Status>
            <ns0:description>esMD validation error. Please correct and resubmit.</ns0:description>
            <ns0:serviceSuccessful>false</ns0:serviceSuccessful>
            <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:Status>
    </ns0:ICDTNotificationFailure>
</ns0:ICDTCommunication>
```

12.1.16 ICDT Administrative Error Response

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

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


Table 21: ESD002.T.ADM.CWBT0002051601EC. ESMD2.D031919.T1334570

```xml
<?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>ESD002.D.L15_2.VWBT0002051601EC.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 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 22: esMD Validation Error Response for Pre-Pay eMDR Letters.

Table 22: esMD Validation Error Response for Pre-Pay eMDR Letters

```xml
<?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>ESMD2.D.L1_5.UWBT0002051601EC.ESD002.D040819.T1607070</ReceivedFileName>
    </submissionMetadata>
    <Status>
      <description>esMD validation error. Please correct and resubmit.</description>
      <serviceSuccessful>true</serviceSuccessful>
    </Status>
  </ESMDTransaction>
</ns0:SubmitOutboundRequestOrResponseResult>
```
12.1.18  **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 23: esMD Validation Error Response for Post Pay eMDR Letters.

As part of the October 2021 release, when there are validation errors in the eMDR Post-Pay structured XML, for the required elements, specific error message(s) will be sent with the name of the data element that failed the validation, to allow the RC to correct any errors and resubmit the package.

**Table 23: esMD Validation Error Response for Post Pay eMDR Letters**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ns0:SubmitOutboundRequestOrResponseResult
xmlns:ns0="http://esmd.ois.cms.hhs.gov/v2/rc">
  <ESMDTransaction TransactionId="EXJ000007095380" DeliveryType="F"
RoutingId="ESD002="/>
  <UniqueID>L16U1K6ES9996D0729211241140</UniqueID>
  <submissionMetadata>
    <RCOID>urn:oid:2.16.840.1.113883.13.34.110.1.999.1</RCOID>
    <RCName>Test Review Contractor</RCName>
    <ContentTypeCode>1.6</ContentTypeCode>
    <LetterId>gurramvt77kaojevyk6as3pyz8mk81uo</LetterId>
  </submissionMetadata>
  <ReceivedFileName>ESMD2.D.L1_6.UWBT0002051601EC.ESD002.D072921.T1241140</ReceivedFileName>
  <Status>
    <description>esMD validation error. Please correct and resubmit.</description>
    <serviceSuccessful>true</serviceSuccessful>
  </Status>
  <ValidationFailure>
    <FailureCode>1082</FailureCode>
    <FailureReason>esMD Validation Error: EITHER THE PROVIDER CITY IS INVALID OR MISSING</FailureReason>
  </ValidationFailure>
</ns0:SubmitOutboundRequestOrResponseResult>
```
### 12.1.19 Document Code File

Document code file is a flat file sent by esMD on a Quarterly basis. Table 24: Document Code File shows the sample Document Code File.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U20200131143015DOCUCODE</td>
<td>The long description of the document which is getting requested</td>
</tr>
<tr>
<td>V0000001UBPDDJVT</td>
<td>The long description of the document which is getting requested</td>
</tr>
<tr>
<td>V0000001UBPDDJVT</td>
<td>The long description of the document which is getting requested</td>
</tr>
<tr>
<td>V0000001UBPDDJVT</td>
<td>The long description of the document which is getting requested</td>
</tr>
<tr>
<td>V0000001UBPDDJVT</td>
<td>The long description of the document which is getting requested</td>
</tr>
<tr>
<td>V0000001UBPDDJVT</td>
<td>The long description of the document which is getting requested</td>
</tr>
<tr>
<td>V0000001UBPDDJVT</td>
<td>The long description of the document which is getting requested</td>
</tr>
</tbody>
</table>

### 12.2 Outbound

Note for API users: 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. ICDT Request.
7. ICDT Solicited Response.
8. ICDT Unsolicited Response.
9. ICDT Pickup/ Pickup Error Notification.
10. ICDT Administrative Error Notification.
11. Pickup Notification for Service Registration.
13. eMDR Process Metadata.
14. eMDR Structured File for Post-Pay ADR letters.
15. API Error Messages for Pre-Pay and Post-Pay.
16. DCF Pickup Notification.
17. DCF Error Pickup Notification.
18. HOPD Pickup Notification.

12.2.1 Pickup Notification
The RC Client generates pickup notifications for all inbound files with delivery type “E” pulled from the esMD Cloud and processed successfully, as detailed in Table 25: Table 4: P_L1_DLL00000009524_Pickup.xml.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:RCPickupNotification
xmlns:ns2="http://esmd.ois.cms.hhs.gov/v1/rc/config">
<ESMDTransactionId>DLL000000009524</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>ESD002.L1.EWB002051601EC.ESMD2.D190315.T120200</fileName>
</ns2:RCPickupNotification>
```

12.2.2 Error Pickup Notification
The RC Client generates pickup error notifications for all inbound files pulled from esMD Cloud and processed unsuccessfully, as detailed in Table 26: P_L1_WBT0002051601EC_Pickup_Error_Notification.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).

2. Extraction was unsuccessful (i.e., the RC client could not successfully unzip the file received from the server).
Table 26: P_L1_WBT0002051601EC_Pickup_Error_Notification.xml

```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>ESD002.D.I1.EWBT0002051601EC.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 error response as detailed in Table 27: R_PA_LQA000000006905_Response.xml.

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 27: R_PA_LQA000000006905_Response.xml

```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>
        <ErrorCode>AG</ErrorCode>
        <ErrorCodeDescription>Procedure Code is invalid - PC001, PC002</ErrorCodeDescription>
      </RejectErrorCodeRecord>
    </RejectErrorCodeRecordList>
  </ErrorResponseDetail>
</PAReviewResponse>
```

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.*

The Administrative Error Response to Inbound Submissions XML is detailed in Table 28: D_ADM_EPP000000008983_AdminResponse.xml.

Table 28: D_ADM_EPP000000008983_AdminResponse.xml

```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 29: Q_L1518DMESD0020315191038490_ICDTSolicitedRequest.xml shows the XML message generated for an ICDT Request XML from RCs.

```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 30: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml shows the sample XML Message of the ICDT Response sent from the RCs.
Table 30: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml

```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 31: R_L152PXHES99960308191419170_ICDTSolicitedResponse.xml

Error! Reference source not found. shows the sample XML Message of the ICDT UnSolicited Response sent from the RCs.

```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>
</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 32: T.ESD002.ICDT.BJ99.ESMD2.D031519.T1201000.

Table 32: T.ESD002.ICDT.BJ99.ESMD2.D031519.T1201000

```xml
<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>ESMD2.D.L15_1.QWBT002051601EC.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>L151ZWESD0020315191038580</ns0:requestID>
      <ns0:contentType>15.1</ns0:contentType>
    </ns0:ICDTMetaData>
    <ns0:creationTime>2019-03-15T12:01:00.335-04:00</ns0:creationTime>
    <ns0:fileName>ESMD2.D.L15_1.QWBT002051601EC.ESD002.D031519.T1038490</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.
2. A virus was found.

Table 33: ESMMD2.T.ADM.CWBT0002051601EC.ESD002.D031919.T1436180 shows the sample Administrative error response XML file.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ICDTCommunication xmlns="http://cms.hhs.gov/esmd/icdt">
  <ICDTNotification FailureRequestType="ADMIN_ERROR">
    <creationTime>2019-03-19T14:36:18.854-04:00</creationTime>
    <fileName>ESD002.D.L15_2.VWBT0002051601EC.ESMD2.D030719.T1309450</fileName>
    <Status>
      <description>ERROR: Admin error notification</description>
      <serviceSuccessful>false</serviceSuccessful>
    </Status>
  </ICDTNotification>
</ICDTCommunication>
```

12.2.10 Service Registration Pickup Notification

After picking up the Service Registration Request flat file, the RC generates the success/failure pickup notification Refer to Table 34: P_L5_WBT0002051601EC_Pickup.xml for successful pickup notification.
Table 34: P_L5_WBT0002051601EC_Pickup.xml

```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>ESD002.D.L5.EWBT0002051601EC.ESMD2.D190315.T1202000</fileName>
</ns2:RCPickupNotification>
```

12.2.11 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 35: Sample eMDRProcessMetadata XML. The same eMDR Process Metadata schema is used for Prepay and Post-Pay functionality.

Table 35: Sample eMDRProcessMetadata XML

```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>
<DocumentUniqueIdentifier="TESTLETTERID4478_20190405&_REVIEW3"
MimeType="application/pdf"
FileName="TESTLETTERID4478_20190405&_REVIEW3.pdf">
<OptionalMetadata>
<FieldName>CheckSum</FieldName>
<FieldValue>8c84d07d505a1429725e5f09a79daf96465592d</FieldValue>
</OptionalMetadata>
<OptionalMetadata>
<FieldName>File Size</FieldName>
<FieldValue>105255200</FieldValue>
</OptionalMetadata>
</Document>
```
12.2.12 eMDR StructuredFile for Post-Pay ADR Letters

Table 36: U_L16Y5XESD0020726192103460_eMDRStructuredFile.xml shows the sample eMDR structured ADR letter file for Post-Pay eMDR letters.

```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>
</eMDRPostPayRequest>
```
12.2.13 API Error Messages for Prepay and Post-pay

Table 37: RC Client Error Codes and Error Messages lists the validation error messages that may be received while generating the Pre-Pay and Post-Pay zip packages in the RC Client API.

<table>
<thead>
<tr>
<th>ID</th>
<th>Scenario</th>
<th>Error Code</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Attachments other than PDF format for Prepay package</td>
<td>ADR_ESMD_LETTERS_FILE_INVALID_ATTACHMENT</td>
<td>ADR esMD Letters File must be in PDF format</td>
</tr>
<tr>
<td>2</td>
<td>File size is 0 MB or greater than 140 MB in size for Prepay Package</td>
<td>ADR_ESMD_LETTERS_FILE_EXCEEDS_MAX_LIMIT</td>
<td>ADR esMD PDF Letters exceeds 140 MB</td>
</tr>
</tbody>
</table>
### 12.2.14 DCF Pickup Notification

RC Client after downloading the Document Code file, generates the successful pickup notification when all the edits are validated successfully. Table 38: P_WBT0002051601EC_3C8_Pickup.xml shows the sample Pickup notification XML.

Table 38: P_WBT0002051601EC_3C8_Pickup.xml

```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>ESD002.T.L17.EWBT0002051601EC.ESMD2.D041020.T160610</fileName>
</ns2:RCPickupNotification>
```

### 12.2.15 DCF Error Pickup Notification

If there are any validation edits failure, RC Client API will generate error pickup notification XML back to esMD. Table 39: DCF Error Pickup Notification shows the sample error pickup notification XML.
### Table 39: DCF Error Pickup Notification

```xml
<?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=""">ESD002.T.L17.EWBT0002051601EC.ESMD2.D200131.T0803010_Invalid</fileName>
</RCPickupNotification>
```

### 12.2.16 HOPD Pickup notification

RC Client API after downloading the HOPD PA program generates the pickup notification back to esMD. Table 40: HOPD Pickup Notification shows the sample HOPD pickup notification.

### Table 40: HOPD Pickup Notification

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

    <ESMDTransactionId xmlns=""">PWI000007041335</ESMDTransactionId>
    <RoutingId xmlns=""">ESD002</RoutingId>
    <PickupTime xmlns=""">2020-04-13T15:44:36.638-04:00</PickupTime>
    <SubmissionTime xmlns=""">2020-04-13T12:19:03.093-04:00</SubmissionTime>
    <fileName xmlns=""">ESD002.T.L17.EWBT0002051601EC.ESMD2.D200131.T0803010_Invalid</fileName>
</ns2:RCPickupNotification>
```
13. RC Client Components

Figure 4: RC Client Components shows the internal components of RC Client application. The following sections describe each component in detail.

**Figure 4: RC Client Components**

![Diagram of RC Client Components]

**Figure 5: RC Client APIs implemented in the esMD Cloud** Illustrates the RC Client APIs which are implemented in the esMD Cloud.
13.1 esMD API’s

The following esMD APIs are used to authenticate the user, transfer files between esMD and the RC Client, send notifications to the HIH in real time and send acknowledgments/responses from the HIH.

- Auth API
- Upload API
- Download API
- Notification API
- Status API

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 esMD Cloud. 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 21.2 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 (e.g., 1 hour or 4 hours) through the RC Client property file. The Scheduler 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 4: RC Client Components is broken down in Figure 6: RC Client Workflow which is followed by a detailed description of the workflow.
Figure 6: RC Client Workflow
15. **Auth API Overview**

1. The esMD Authentication API is used to authenticate the RC Client user by verifying the user id, password, client id and client secret. The Auth API receives a token after the user is successfully authenticated. The token is used by other APIs to upload and download files from esMD Cloud.

2. If the user is not found in the system, the API returns an error message back to the user.

3. The Authentication Token is valid for only 30 minutes after the user makes the API call.

4. If the Authentication token expires, the user is allowed to make another Auth API call.

**Figure 7: esMD Auth API Sequence Diagram**

**Auth Endpoint URL:**

https://esmdcdev.cms.hhs.gov/esmd/v1/auth?clientid=3fis5elmln49c3fkfks5v20ml&clientsecret=1ad4odhli6jur6qfbaoa71c52rt836b0v8l3jgh5tivosvmkc48&username=123&passwd=tsdf&scope=devesmdrc/download

**REQUEST BODY**

None
REQUEST PARAMETERS

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Data Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>User ID</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>password</td>
<td>Password</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>clientid</td>
<td>Client ID to authorize the user</td>
<td>header</td>
<td>string</td>
<td>yes</td>
</tr>
<tr>
<td>clientsecret</td>
<td>Client Secret to authorize the user</td>
<td>header</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>scope</td>
<td>Scope to limit user access to specific API. EX: devesmdrc/download</td>
<td>header</td>
<td>String</td>
<td>yes</td>
</tr>
</tbody>
</table>

Request:
GET /esmd/v1/auth
uid: testuserid
passwd: xxxxxx
clientid: xxxxxxxx
clientsecret: xxxxxxxx
scope: devesmdrc/download

Response Success:
application/json

```json
{
    "access_token": "xxxxxxxxxxxxxxxxxxxxxx",
    "token_type": "Bearer",
    "expires_in": 3600
}
```

HTTP/1.1 200 Accepted

Response Failure:
application/json

HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store

```json
{
    "error": "Authentication Failed.Invalid UserID/Password"
}
```
16. Upload API Overview

The Upload API allows the RC Clients to upload files to esMD Cloud. The Authentication Token will be generated through the Auth API. Once authentication is successful, the RCs can upload the files successfully using the presigned URL generated with filename and token.

If the metadata validations fail, the Upload API will reject the request and send an error message.

Once the Upload API call is successful, a presigned URL is generated with the following metadata elements:
- Filename
- contentChecksum
- content type

1. The presigned URL is valid for 15 minutes for the user to upload the documents.
2. The system cross-validates the metadata information against the file name with the presigned URL and the files are uploaded to esMD once the validation is successful.
3. If the cross validation between the presigned URL and Metadata information is failed, an error notification will be generated and send to the review contractor.

Figure 8: esMD Upload API Sequence Diagram
Endpoint URL: https://esmdc-dev.cms.hhs.gov/esmd/v1/objects

REQUEST BODY

None

REQUEST PARAMETERS

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Data Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>senderroutingID</td>
<td>Sender Routing ID</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>content-type</td>
<td>Content Type</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>Authorization</td>
<td>Token</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>contentchecksum</td>
<td>checksum (autogenerated)</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>filename</td>
<td>Filename</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>uid</td>
<td>username</td>
<td>header</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>size</td>
<td>Content Length</td>
<td>header</td>
<td>String</td>
<td>Yes</td>
</tr>
<tr>
<td>Content-md5</td>
<td>Checksum is autogenerated and received in md5 binary converted to base 64 format</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>Scope</td>
<td>Scope to limit user access to specific API. EX: devesmdrc/download</td>
<td>header</td>
<td>String</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request:

POST /esmd/v1/objects

size: 100 mb
Content-md5: xdfdsasdfsfrergewq
Content-type: application/zip
Authorization:
eb0c4513bb3d8b265e4f7a33cebeb61f5c527a74b8d516efe8465e2378cc5853
uid:testuserid
senderroutingid: ES0001
contentchecksum: xdfdsasdfsfrergewq
scope: devesmdrc/upload

Response (esMD Received):
application/json
HTTP/1.1 200 Ok

```json
{
   "status": "SUCCESS",
   "message": "FILE UPLOADED SUCCESSFULLY",
   "filename": "esd001.D.guid1234.esmd2.D123234.T121212.zip"
}
```

Error Response (Upload failed):
HTTP/1.1 202

```json
{
   "errorDetails": [{
      "filename": " ESMD2.D.L1_5.GUID01020304051.ES0001.D040622.T1051500.zip",
      "message": "VALIDATION FAILED",
      "status": "failed",
      "errorMessage": "Invalid request - File already available",
      "errorCode": "<DUPLICATE_ERROR_CODE>"
   }]
}
```

File Format:

<<ReceiverRoutingId>>.<<environment>>.L<<CTC>>.<<DeliveryType>>GUID.<<SenderRoutingID>>.DMMdyy.TTHHmmsS.zip

Elements Description Inside the File Name:
- Receiver Routing ID:  RC/esMD Mailbox/sub_id (Ex: ESMD2)
- CTC:  Content Type Code (associated CTC code, numeric and period allowed characters)
- DeliveryType:  Transaction types (is 1char and allowed value is Alphabet only)
Ex: (Q- ICDT Request. R-> ICDT Solicited response/unsolicited response)
- **GUID**: esMD Transaction ID or Global Unique Id (15 AN RC Client generated random id)
- **Sender Routing ID**: RC/esMD Mailbox
- **DMMddyy**: Current date in Mmddyy format
- **THHmmssS**: Current time in HHmmssS format

File Name Example:
ESMD2.L11_2.EGUID01020304051.ESMD01.D040322.T0732220.zip
17. Download API Overview

The Download API is used to download the files from esMD cloud. The API gets the list of files ready to be downloaded using the token returned by the Auth API. The Download API gets a presigned URL for each file and downloads the file to esMD Cloud using the presigned URL.

1. The user can send the request to the Download API to get the list of files available for download in esMD with the valid metadata below provided in the API request.
   - uid
   - senderroutingid
   - authorization

2. Once the Download API call is successful, the list of files available for download will be sent to the user.

3. The Download API will reject the API request and send an error message if the metadata validations fail.

4. If the file name is validated successfully, a presigned URL is generated with the following metadata elements:
   - Filename

5. The Filename should be sent in the format below in the Download API request.
   Format: <<ReceiverRoutingId>>.<<environment>>.L<<CTC>>.<<Delivery_type>>GUID. <<SenderRoutingID>>.DMMddyy.THHmmssS.zip

Elements Description Inside the File Name:
- Receiver Routing ID: RC/esMD Mailbox/sub_id (Ex: ESD0001)
- CTC: Content Type Code (associated CTC code, numeric and period allowed characters)
- DeliveryType: transaction types ( E – xdr/x12 package, Q – ICDT Request …etc) (is 1char and allowed value is Alphabet only)
- GUID: esMD Transaction ID or Global Unique Id (15 AN RC Client generated random id)
- Sender Routing ID: RC/esMD Mailbox
- DMMddyy: current date in Mmddyy format
- THHmmssS: current time in HHmmssS format

File Name Example:
ES0001.D.L11_2.EGUID01020304051.ESMD01.D040322.T0732220.zip
6. Using the presigned URL, the RCs will receive an attachment in zip format and must be able to download the files.
7. The Presigned URL is valid for 15 minutes for the user to download the documents.

Figure 9: esMD Download API Sequence Diagram

17.1.1 Get List of Files

Endpoint URL: [https://esmdc{env}.cms.hhs.gov/esmd/v1/objects](https://esmdc{env}.cms.hhs.gov/esmd/v1/objects)

Method: GET

This API is used to get the list of files ready to download in esMD.
REQUEST BODY

None

REQUEST PARAMETERS

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Data Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>User ID</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>senderroutingID</td>
<td>Sender Routing ID</td>
<td>header</td>
<td>string</td>
<td>yes</td>
</tr>
<tr>
<td>authorization</td>
<td>Sender Organization ID</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>scope</td>
<td>Scope to limit user access to specific API. EX: devesmdrc/download</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
</tbody>
</table>
RESPONSES

application/json

200 OK
Success

EXAMPLE

Request:
GET /esmd/v1/objects

uid: {uid}
senderroutingid: ESD001
authorization: xxxxxxxxxxxxxxxxxxxxxxx

Response:
HTTP/1.1 200 OK

```json
{
  "status": "SUCCESS",
  "message": "",
  "objects": [
    {
      "filename": "T#EFT.ON.D.SD002.L7.EAZV.D070122.T0640010",
      "size": "",
      "statusFlag": "",
      "createdOn": "2022-07-01T06:39:49.279534",
      "id": "",
      "lastDownloaded": ""
    }
  ],
  "count": 1
}
```

Error Response:

```json
{
  "status": "failed",
  "error_message": "<<Error Message>>",
  "error_code": "<<ERROR_CODE>>"
}
```

17.1.2 Generate Presigned URL for zip file:
This API returns the presigned URL to download a zip file from esMD Cloud.
Figure 11: esMD Generate PresignedURL Sequence Diagram

REQUEST PARAMETERS

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Data Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>User ID</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>senderroutingID</td>
<td>Sender Routing ID</td>
<td>header</td>
<td>string</td>
<td>yes</td>
</tr>
<tr>
<td>authorization</td>
<td>Sender Organization ID</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request


**Method:** GET

**Response (Review Contractor Received):**
application/json

uid: {uid}
senderRoutingid: ESD001
authorization: xxxxxxxxxxxxxxxxxxx

HTTP/1.1 202 Accepted

```
{
  "status": "SUCCESS",
  "message": "",
  "contents": [
    {
      "lastDownloaded": "",
      "comments": ""
    }
  ]
}
```

Error Response:
HTTP/1.1 404 Not Found

```
{
  "status": "FAILED",
  "message": "VALIDATION FAILED",
  "filename": "",
  "errorDetails": [{
    "errorCode": "E0001",
    "errorMessage": "Required header missing"
  }]
}
```
18. Notification API

When the RC downloads a package and processes it, the Notification API is invoked to send either a Successful or Error pickup notification to the HIH. RCs receive the Acknowledgement for the Pickup Notification in real time. The Notification API is used to process the messages below from the RC to the HIH.

- Pickup Success Notification
- Pickup Failure Notification
- PA reject response
- Admin Error
- ICDT Pickup Notification
- ICDT Admin Error

The PA Reject Response and Admin Error notifications are processed asynchronously using a scheduler which runs at specific intervals of time. Currently the notification process runs when the inbound or outbound process is enabled and uses the “pushFrequency” value from the configuration to send the notifications.

**Figure 12: esMD Notification API Sequence Diagram**

**Endpoint URL:** [https://esmdc-dev.cms.hhs.gov/esmd/v1/objects/notification/pickup](https://esmdc-dev.cms.hhs.gov/esmd/v1/objects/notification/pickup)

**Method:** POST

**REQUEST PARAMETERS**
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Data Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization</td>
<td>Token</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>scope</td>
<td>Scope</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>content-type</td>
<td>Content Type</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Request:**
POST /esmd/v1/objects

Content-type: application/json
Authorization: eb0c4513bb3d8b265e4f7a33cebeb61f5c527a74b8d516efe8465e2378cc5853
scope: devesmdrc/notification

**Pickup Notification from RC to HIH**

```json
{
  "notificationType": "PICKUP",
  "senderRoutingId": "ESD002",
  "contenttypecd": "1",
  "notification": [
    {
      "esMDTransactionId": "KSZ0002296501EC",
      "pickupTime": "2022-07-14T14:46:32.9061123-04:00",
      "submissionTime": "2022-07-14T14:46:32.9031133-04:00",
      "status": "Success",
      "errorMessages": []
    }
  ]
}
```

**Pickup Success Response:**
application/json
HTTP/1.1 200 Ok

```json
{
  "senderRoutingId": "ESD002",
  "statusDetails": [
    {
      "esMDTransactionId": "WEB0002251201EC",
      "contenttypecd": "1",
      "status": "Success",
```

---

AR2023.01.0 Review Contractor (RC) Client Microsoft .NET User Guide and Installation Handbook

Version 11.0 102 Electronic Submission of Medical Documentation (esMD)
"statusDescription": "Successfully delivered to HIH",
"errorMessages": []
}
]
}

Pickup Failure Response:
HTTP/1.1 202

{
  "senderRoutingId": "ESD002",
  "statusDetails": [
    {
      "esMDTransactionId": "KSZ0002296501EC",
      "contenttypecd": "",
      "status": "Failed",
      "statusDescription": "Metadata Validation Failed",
      "errorMessages": [
        {
          "errorCode": "ESMD_PROCESSED_PICKUP_NOTIFICATION",
          "errorName": "",
          "errorDescription": "ESMD validation error: esMD processed pickup notification for this transaction"
        }
      ]
    }
  ]
}
19. Status API

The Status API is a scheduled process that runs at specific intervals of time and retrieves the status from esMD for the scenarios below and updates the RC on the Successful notification delivery status to the HIH/RC.

- Pickup Success Notification
- Pickup Failure Notification
- PA reject response
- Admin Error
- ICDT
  - Validation Errors
  - Acknowledgement

The Status messages are processed asynchronously using a scheduler which runs at specific intervals of time. Currently the Status API runs when the inbound or outbound process is enabled and uses the “pushFrequency” value from the configuration to receive statuses.

**Figure 13: esMD Status API Sequence Diagram**

**Endpoint URL:** [https://esmdc-dev.cms.hhs.gov/esmd/v1/objects/status/rc](https://esmdc-dev.cms.hhs.gov/esmd/v1/objects/status/rc)

**Method:** GET

**REQUEST BODY**
REQUEST PARAMETERS

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Data Type</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization</td>
<td>Token</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>scope</td>
<td>Scope</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>content-type</td>
<td>Content Type</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
<tr>
<td>uid</td>
<td>Username</td>
<td>header</td>
<td>string</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Request:

POST / api/esmd/v1/objects/status/rc

Content-type: application/json
Authorization:
eb0c4513bb3d8b265e4f7a33cebeb61f5c527a74b8d516efe8465e2378cc5853
scope: devesmdrc/status

Response:

application/json

Success Response:

HTTP/1.1 200

```json
{
    "status": "SUCCESS",
    "message": "THE FOLLOWING objects ARE delivered to HIH at this point of time",
    "statusDetails": [
        {
            "notification_type": "PICKUP",
            "delivery_type": "N",
            "esmdtransactionid": "ETI0005027701EC",
            "status": "Success",
            "statusDescription": "SENT PICKUP STATUS TO HIH"
        },
        {
            "notification_type": "ADMIN",
            "delivery_type": "N",
            "esmdtransactionid": "ETI0005027701EC",
```
"status": "Success",
"statusDescription": "SENT PICKUP STATUS TO HIH"
},
{
"notification_type": "PAREJECT",
"delivery_type": "N",
"esmdtransactionid": "ETI0005027701EC",
"contenttypecd": 1
"statusDescription": "SENT PICKUP STATUS TO HIH"
},
"count": 3
}

Error Response:

{
"message": "client sent an invalid request, such as lacking required request header or parameter"
}

Note: Please refer to the ICD document for validations and additional error details about the APIs.

19.1 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 14: Request/Response Business Process Flow Diagram shows the process flow of ICDT Request and Response, and Table 41: ICDT Request/Response Business Process Flow Steps provides the detailed steps.
Figure 14: Request/Response Business Process Flow Diagram

Table 41: ICDT Request/Response Business Process Flow Steps

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The Review Contractor ‘A’ creates the ICDT Request package, which consists of an ICDT Request file, and sends the package to esMD via esMD Upload API to be delivered to another RC.</td>
</tr>
<tr>
<td>1.2</td>
<td>The 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.</td>
</tr>
<tr>
<td>1.3</td>
<td>The esMD system delivers the ICDT Request package to the Review Contractor ‘B’ if the validation is successful.</td>
</tr>
<tr>
<td>1.4</td>
<td>The Review Contractor ‘B’ downloads the ICDT Request Package and sends the Successful Pickup notification, Error pickup notification via esMD Notification API, and admin errors to esMD via the esMD Status API.</td>
</tr>
<tr>
<td>1.5</td>
<td>The esMD system delivers the Pickup notification, error pickup notification, or admin error to the Review Contractor ‘A’.</td>
</tr>
<tr>
<td>2.1</td>
<td>The Review Contractor ‘B’ sends the ICDT Response package to the esMD system.</td>
</tr>
<tr>
<td>2.2</td>
<td>The esMD system 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.</td>
</tr>
<tr>
<td>2.3</td>
<td>The esMD system delivers the ICDT Response package to the Review Contractor ‘A’.</td>
</tr>
<tr>
<td>2.4</td>
<td>The Review Contractor sends the successful pickup notification, error pickup notification, or admin error to the esMD system via esMD notification API and esMD Status API.</td>
</tr>
<tr>
<td>Message Sequence</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>2.5</td>
<td>The esMD system validates and delivers the Pickup notification, error pickup notification, or admin errors to the Review Contractor ‘B’.</td>
</tr>
</tbody>
</table>

### 19.2 Service Registration Processing Overview

Table 42: Service Registration Flow Steps describes the typical Service Registration flow interaction as shown in Figure 15: Service Registration Process Flow.
Figure 15: Service Registration Process Flow

Table 42: Service Registration Flow Steps

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>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.</td>
</tr>
</tbody>
</table>
### 19.3 Document Codes Processing Overview

Table 42: Service Registration Flow Steps describes the typical Document Codes flow interaction.

The RC Client API shall download/pull the new DCF Flat File from the esMD Cloud and initially move the file to the Temp Folder. The RC Client shall continue processing with the header, body, and trailer validation. The RC Client shall move the DCF file to the input folder and send a Success pickup notification on successful validation of the file. The RC Client shall push the error pickup notification to esMD if there is any validation failure, 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 20.1 DCF 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.
20. Schema Definition and Sample Files

20.1 DCF File Format

Figure 16: Document code file (DCF) Flow

Table 43: Document Code File Process Flow Steps

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The RC shall receive the DCF flat file from the esMD system via esMD Download API.</td>
</tr>
<tr>
<td>1.2</td>
<td>The RC system processes the DCF flat file received from the esMD and generates the appropriate acceptance or rejection response acknowledgement to the esMD</td>
</tr>
</tbody>
</table>

20.2 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.
20.2.1 Logical Process Flow

20.2.1.1 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 17: eMDR Pre-Pay Process Flow depicts the logical processing of the eMDR (Pre-Pay) process, and Table 44: eMDR Pre-Pay Logical Process Flow Steps details the eMDR process.
Table 44: eMDR Pre-Pay Logical Process Flow Steps

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>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.</td>
</tr>
<tr>
<td>2.1</td>
<td>The RC sends ADR letters (PDF) matching the eMDR requests in the zip request package to esMD.</td>
</tr>
<tr>
<td>2.2</td>
<td>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.</td>
</tr>
<tr>
<td>3.1</td>
<td>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.</td>
</tr>
<tr>
<td>3.2</td>
<td>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. 4. Error.</td>
</tr>
<tr>
<td>4.1</td>
<td>The HIH sends the package delivery confirmation to esMD after the ADR PDF letter and eMDR structured XML are successfully transmitted to the Provider.</td>
</tr>
<tr>
<td>4.2</td>
<td>esMD acknowledges the delivery confirmation received from the HIH.</td>
</tr>
<tr>
<td>5.1</td>
<td>The esMD system sends the transaction details only when HIH delivery failed due to validation error or transmission error.</td>
</tr>
<tr>
<td>5.2</td>
<td>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. 4. Error.</td>
</tr>
</tbody>
</table>

20.2.2  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 is driven by RC submissions. The Content Type Code for this program will be 1.6.

Figure 18: eMDR Post-Pay Process Flow depicts the logical processing of eMDR (Post-Pay) process, and Table 45: eMDR Post-Pay Logical Process Flow Steps details the sequence of interaction between esMD and HIH.
Figure 18: eMDR Post-Pay Process Flow

Table 45: eMDR Post-Pay Logical Process Flow Steps

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The esMD system receives the ADR Letter PDF and structured eMDR Post-Pay XML within a zip file from the RC.</td>
</tr>
<tr>
<td>1.2</td>
<td>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.</td>
</tr>
<tr>
<td>2.1</td>
<td>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.</td>
</tr>
<tr>
<td>Message Sequence</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 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.  
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.  
4. Error. |

As part of the October 2021 release, only one .PDF letter can be submitted in the eMDR Post-Pay RC package. If more than one PDF letter is included in the RC package, the request will be rejected with an error message. In addition, the eMDR Post-Pay schema definition will be updated to remove the restriction for the Zip code under the Sender Details.

### 20.3 Start RC Client

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

#### 20.3.1 Login and Encryption

The RC Client prompts the user for the following details:

1. IDM User ID.
2. IDM Password.

After successful login, IDM login credentials are encrypted in memory and used when needed to authenticate by the Auth API for upload and download files from esMD Cloud. The RC Client initiates two threads, one for the inbound process and one for the outbound process. These processes are described on in Sections 20.4 Upload Process and 20.5 Download Process, respectively.
20.4 Upload Process

20.4.1 Outbound Start
The esMD Auth API authenticates the RC Client user with their IDM credentials and returns a token which is valid for 15 minutes. The RCClient Upload API calls the esMD Upload API with the token and other header information to get the presigned URL to upload the file. RC Client loads the parameters for the outbound process from the RC Client 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. Upload API URL and other details to upload the file to esMD Cloud.
5. Auth API details for the chosen environment (esMD Auth API).

20.4.2 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.

20.4.3 Connect
The RC Client connects to the esMD Cloud environment using IDM login credentials. The Encryption utility decrypts the credentials in memory and authenticates the user using esMD Auth API. If the user password is expired, the connection fails, prompting the user to provide the login information again.

20.4.4 Upload
The RC Client uploads the outbound files to the esMD Cloud. 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.

20.5 Download Process

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

- Pull frequency.
- Directories used by the RC Client to save the downloaded files (inputDirectory).
- Download API and Notification URLs and other details from properties file to upload the file to esMD Cloud and also to send notifications in real time.
• The ClientID and clientSecret used by the Auth API.

20.5.2 Extraction Failure
If the extraction process was interrupted during extraction during the run, then there will be files sitting in the local "temp" directory, which needs to be manually moved/deleted.

20.5.3 Extraction
When the RC Client runs the inbound process and downloads the files. It will then extract the downloaded file, if the extraction is successful, RC Client proceeds to "checksum verification"; otherwise, RC Client creates an error pickup notification.

20.5.4 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.

20.6 Acknowledgements

20.6.1 Pickup Notification
If the RC Client successfully extracts and verifies the compressed files, the RC Client sends a SUCCESS notification by invoking esMDNotification API to inform the HIH that the document has been received and successfully processed. The esMDNotification API is a synchronous call which receives the acknowledgement response back from the HIH and saves the XML file in the notification folder.

To generate this SUCCESS notification, the RC Client should:

1. Get the Transaction ID from the compressed file name.
2. Prepare the notification with a SUCCESS message and generate an XML notification file.
3. Post the pickup notification message in JSON format to esMD by calling the esMD Notification API.
4. Convert the HIH Acknowledge JSON message received in the same synchronous call to XML format and save the file in the notification folder.
20.6.2 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 which asks the HIH to resubmit the package.

In order to generate this error notification, the RC Client must:

- Obtain the Transaction Identifier (TID) from the compressed file name.
- Prepare the notification with an error message.
- Generate a JSON notification file.
- Post the JSON notification message to esMD.

20.7 Connect

After the Housekeeping Manager completes preprocessing, the RC Client authenticates the user by calling the Auth API with the encrypted username password stored in memory. The RC Client also enables the User ID and Password fields to log in into the RC Client. Once the RC Client is logged in successfully, the processing will start.

20.8 Get Notifications

The RC Client uses the esMDDownload API to get a list of the available inbound documents for the RC on the esMD Cloud.

20.9 Process Document

If any documents are available for the RC Client in the esMD Cloud, the RC Client will go through the list to pull each document using the esMD Download API and process the document.

20.10 Download Document

The RC Client uses the esMDDownload API to pull each inbound document from esMD cloud. The RC Client then extracts the contents of the zip file and continues processing.
21. .NET Client API

21.1 Push Frequency for Outbound Responses

.NET RC Client V5.2 has a new method included in the Login Process for the Outbound check frequency. Users (RCs) can use a separate Outbound check interval (set in the esmd-rc-client-config file) independent of the Inbound check frequency.

21.2 Security

When the RC Client starts, the user credentials are provided because they are stored in encrypted form in memory. Figure 19: Encryption and Decryption Process shows the processes used to safeguard the IDM user credentials from exposure.

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

![Encryption and Decryption Process](image)}
21.3 .NET API Documentation

This section discusses API methods that can be called for a custom solution to interface with the esMD Cloud. RCs who choose to use the RC .NET client out-of-the-box may skip this section.

21.3.1 Auth

Table 46: The esMD.RC Client.Login.LoginProcess Methods describes the RC Client Login process.

<table>
<thead>
<tr>
<th>No.</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>public bool Authenticate()</td>
<td>User login procedure. The username and password properties are encrypted and set only when this method returns TRUE. Returns: TRUE if the user logs in successfully.</td>
</tr>
<tr>
<td>2.</td>
<td>MessageDTO ValidateLogin(LoginDTO logindto);</td>
<td>This method validates the User ID and Password For Login Tab. Returns: FALSE if error Messages are found, returns with a list of Error Messages.</td>
</tr>
<tr>
<td>3.</td>
<td>MessageDTO LoginProcessAPI(LoginDTO logindto);</td>
<td>This method does the Login Process. Returns: TRUE if the user logs in successfully. Returns: FALSE if the user fails to login with an Error Message.</td>
</tr>
<tr>
<td>4.</td>
<td>AuthResponse GetToken(String username, String password, String scope)</td>
<td>This method returns the token if the user is authenticated successfully.</td>
</tr>
</tbody>
</table>

21.3.2 Download

Table 47: The esMD.RC Client.Inbound.Inbound Methods details the RC Client Inbound process.
### Table 47: The esMDRC Client.Inbound.Inbound Methods

<table>
<thead>
<tr>
<th>No.</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>public SortedList&lt;long, string&gt; GetNotifications(string remoteDownloadDirectoryPath, string filePattern)</td>
<td>This method connects to the esMD cloud and checks for any available notifications. Parameters: 1. remoteDownloadDirectoryPath – The remote directory path to download from as a String. 2. filePattern – The File Name Pattern to look for as a String. Returns: A list for file names sorted by last modified time, oldest first.</td>
</tr>
<tr>
<td>2.</td>
<td>public string PullDocument(string remoteDocName, string localDocName)</td>
<td>This method is used to pull the document (i.e., zip file) from the esMD Cloud using the remoteDocName and saves it locally in the “temp” directory as the localDocName. Parameters: 1. remoteDocName – The remote file to pull as a String. 2. localDocName – The local file name to save as a String. Returns: Error message if any errors. A null return value means downloading succeeded.</td>
</tr>
<tr>
<td>3.</td>
<td>public string ExtractDocument(string zipFileName, string targetDirectory)</td>
<td>Extracts the zip file that was downloaded from the esMD cloud. Parameters: 1. zipFileName – The local zip file to extract. 2. targetDirectory – The target directory to place the extracted contents. Returns: The directory name – the location where the extracted file(s) stored in the local file system.</td>
</tr>
<tr>
<td>4.</td>
<td>objectPresignedURL GenerateDownloadPresignedURLforZipfile (string filename, string token, string uid, string senderroutingid)</td>
<td>Gets the presigned URL for the given filename.</td>
</tr>
<tr>
<td>5.</td>
<td>DownloadZipfileUsingPresignedURL(string presignedurl, string token, string filename, string localtemppath)</td>
<td>This method downloads the file using the presigned URL.</td>
</tr>
<tr>
<td>No.</td>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>6.</td>
<td>GetReadyToDownloadFilesFromsmd(string token, string uid, string senderroutingid)</td>
<td>This method gets the list of files ready to be downloaded from esMD.</td>
</tr>
<tr>
<td>7.</td>
<td>public bool ProcessMedicalDocumentation(string remoteDocumentName)</td>
<td>This is the “housekeeping” method. It does the following: 1. Downloads the zip file from the esMD cloud using the PullDocument() method based on the name passed to the “temp” directory. 2. Extracts the zip file into the “download” directory using the ExtractDocument() method. 3. If extraction fails, calls the Acknowledge() method with an error event. 4. After successful extraction, verifies the extracted payloads against the checksum in the metadata file using the CheckPayloads() method. 5. If checksum fails, calls the Acknowledge() method with an error event. 6. If checksum passes, calls the Acknowledge() method with a success event. Parameter: 1. remoteDocumentName – The remote document name to pull and process. Returns: The Boolean status of the processing for that document.</td>
</tr>
<tr>
<td>8.</td>
<td>SubmitPickupNotification(string token, string jsonString, string type)</td>
<td>Submits the Pickup Notification JSON message to esMD.</td>
</tr>
<tr>
<td>9.</td>
<td>prepareNotificationRequestInfo(RCPickupNotification rcPickup, string type)</td>
<td>Prepares the Pickup Notification object.</td>
</tr>
<tr>
<td>10.</td>
<td>public bool CheckPayloads(string localExtractedDirectory, ESMDDocument[] esmdDocuments)</td>
<td>Checks the payload files against the metadata from the package. Parameters: 1. localExtractedDirectory – The directory in which the payloads were extracted to as a File. 2. esmdDocuments – The payloads metadata captured in ESMDDocument objects. Returns: The status of the checksum verification.</td>
</tr>
</tbody>
</table>
21.3.3 Outbound


<table>
<thead>
<tr>
<th>No.</th>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1.  | public SortedList<long, string> GetOutboundDocuments(string outboundDir, string filePattern) | This method is used to retrieve the list of outbound documents in the “output” directory to be pushed. Parameters:  
1. outboundDir – The local “output” directory to push files from as a String.  
2. filePattern – The file name pattern to push as a String.  
Returns: A list of file names (without a directory path). |
| 2.  | public string PushDocument(string localDocName, string remoteDirectory)   | This method used to push a local compressed document from the “output” directory to the esMD cloud. Parameters:  
1. localDocName – The name of the file to push as a String.  
2. remoteDirectory – The remote directory name to push to as a String.  
Returns: An error message if any. A null return value means uploading succeeded. |
|     | generatePresignedURLforUploadingDocument(UploadRequestInfo uploadRequestInfo) | Generate the presigned URL to upload the document to esMD. Parameters:  
1. uploadRequestInfo – The upload request info to generate the presigned URL. |
| 2.  | uploadDocumentUsingPresignedURL(string filelocation, string token, UploadPreSignedURLResponse uploadRequestInfo, string checksum, string size) | Upload the document using presigned URL. Parameters:  
2. filelocation – The path of the file to push as a String.  
3. token – The token returned by Auth API.  
4. uploadRequestInfo – Upload Request Header info.  
5. checksum – Checksum calculated for the file.  

21.3.4 PA Error (Rejected Decision) Response

Table 49: Manual Submission of PA Error (Rejected Decision) Response details the methods to submit the PA Error (Rejected Decision) Response.
Table 49: Manual Submission of PA Error (Rejected Decision) Response

<table>
<thead>
<tr>
<th>No.</th>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1.  | MessageDTO validateErrorResponse(ErrorResponseToPARequestDTO ErrorRespDTO); | This method takes ErrorResponseToPARequestDTO object as input, which has the review decision response information provided by user and validates all information before generating the XML response.  
Parameter: ErrorResponseToPARequestDTO – The ErrorResponseToPARequestDTO object to use.  
Returns: The Message Object which has status of validations result and also the list of Validation Failure Data Transfer Objects (DTO) if there is any validation failure with the data provided by the user. |
| 2.  | string CreateErrorResponseAPI(ErrorResponseToPARequestDTO ErrorRespDTO); | This method takes ErrorResponseToPARequestDTO object as input which has the review error (rejected decision) response information provided by user and creates the SubmitPADeterminationResponse object.  
Parameter:  
1. ErrorResponseToPARequestDTO – The ErrorResponseToPARequestDTO object to use.  
Returns: The SubmitPADeterminationResponse object populated with the data provided by the user. |

21.3.5 Administrative Error Response to Inbound Submissions

Table 50: Administrative Error Response to Inbound Submissions details the methods to submit the Administrative Error Response to inbound submission.

Table 50: Administrative Error Response to Inbound Submissions

<table>
<thead>
<tr>
<th>No.</th>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1.  | MessageDTO ValidateAdminResponseAPI(AdministrativeErrorPropertiesDTO adminErrDTO); | This method takes AdministrativeErrorPropertiesDTO 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. AdministrativeErrorPropertiesDTO – the AdministrativeErrorPropertiesDTO object to use.  
Returns: The Message Object which has status of validations result and also the list of Validation Failure DTO object if there is any validation failure with the data provided by the user. |
21.3.6 Status

Table 51: Status messages from esMD

<table>
<thead>
<tr>
<th>No.</th>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>GetStatusForTransaction(string token, string uid, string senderroutingid, string requestType)</td>
<td>Gets the Status from esMD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. uid – Username.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. token – The Token returned by Auth API.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Senderroutingid – Sender Routing ID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. requestType – Type of request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns: The Status Message.</td>
</tr>
<tr>
<td>2.</td>
<td>GetStatusForTransactionById(string token, string uid, string senderroutingid, string esmdtransactionid)</td>
<td>Gets the Status from esMD for the transactionID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Parameter:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. uid – Username.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. token – The Token returned by Auth API.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6. Senderroutingid – The Sender Routing ID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7. esmdtransactionid – The esMD TransactionID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns: The Status Message.</td>
</tr>
</tbody>
</table>

21.3.7 Utilities – Encryption

*Note: The .NET 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 52: EMSD.RC Client.Encryption.EncryptionUtil Methods

<table>
<thead>
<tr>
<th>No.</th>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1.  | public string EncryptCredential(string credential) | This method is used to encrypt the IDM login credentials using an RSA Public Key from the key container. Parameter:  
1. credential – User’s login name or password to encrypt as a String. Returns: The encrypted credential. |
| 2.  | public string DecryptCredential(string credential) | This method is used to decrypt the IDM login credentials using an RSA Private Key from the key container. Parameter:  
1. credential – User’s encrypted login name or password. Returns: The decrypted credential. |

### 21.3.8 Advanced / Debugging API

Refer to Table 53: Remote Troubleshooting for details on the Execute Handshake method.

### Table 53: Remote Troubleshooting

<table>
<thead>
<tr>
<th>No.</th>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>public bool ExecuteHandshake()</td>
<td>This sample method invokes a call to the esMD cloud to pass login information to assist in remote troubleshooting. Returns: TRUE if handshake succeeded.</td>
</tr>
</tbody>
</table>
| 2.  | MessageDTO ValidateTestConnection(TestConnectionDTO TestConnDTO); | This method takes TestConnectionDTO object as input which has the Advanced/Debugging information provided by user and validates all the information before testing the connection to esMD Cloud. Parameter:  
1. TestConnectionDTO – the TestConnectionDTO object to use. Returns: The Message Object which has status of validations result and also the list of Validation Failure DTO object if there is any validation failure with the data provided by the user. |
21.3.9 Validation API

Refer to Table 54: Validation Methods used for validating fields on the Review Response screen.

Table 54: Validation Methods

<table>
<thead>
<tr>
<th>No.</th>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>ValidationUtil</td>
<td>ValidationUtil Class has two methods</td>
</tr>
<tr>
<td>1.</td>
<td>public bool ValidateFormatAlphaNumeric(string TextToValidate)</td>
<td>1.  public bool ValidateFormatAlphaNumeric(string TextToValidate)- This method will be used Validate the input for Alphanumeric characters(Used for Procedure Code &amp; Unique Tracking Number (UTN) Returns True, if the string is alphanumeric or else returns False</td>
</tr>
<tr>
<td>2.</td>
<td>public bool ValidateFormatNumeric(string TextToValidate)</td>
<td>2.  public bool ValidateFormatNumeric(string TextToValidate) - This method will be used Validate the input for Numeric characters( Used for Transaction ID and Number of units), Returns True, if the string is Numeric or else returns False</td>
</tr>
<tr>
<td>3.</td>
<td>public void validateReasonCodesInPAErrorResponse(List&lt;String&gt; ReasonCodesList, ref List&lt;ValidationFailureDTO&gt; validationFailureListDTO)</td>
<td>3. This method will be used to validate the missing reason codes for PA Review response (Decision 'N' only) and for PA Error response.</td>
</tr>
<tr>
<td>4.</td>
<td>private bool validateIndividualReasonCodes(List&lt;String&gt; ReasonCodesList, ref List&lt;ValidationFailureDTO&gt; validationFailureListDTO)</td>
<td>4. This method will be used to validate the length of a reason code for PA review response (Decision 'M' and 'N') and PA Error Response.</td>
</tr>
<tr>
<td>5.</td>
<td>private bool validateReasonCodesExceedsMax(List&lt;String&gt; reasonCodesList, ref List&lt;ValidationFailureDTO&gt; validationFailureListDTO)</td>
<td>5. This method validates if more than 25 reason codes are provided.</td>
</tr>
<tr>
<td>6.</td>
<td>private bool isDuplicatesReasonCodesExists(List&lt;String&gt; ReasonCodesList, ref List&lt;ValidationFailureDTO&gt; validationFailureListDTO)</td>
<td>6. This method validates if Duplicate reason codes are provided.</td>
</tr>
</tbody>
</table>
22. API Methods

22.1 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 55: Example Unique ID Rules and Format.

1. ICDT Solicited Request – CTC 15.1
2. ICDT Solicited Response – CTC 15.2
3. ICDT Unsolicited Response – CTC 15.3

Table 55: Example Unique ID Rules and Format

<table>
<thead>
<tr>
<th>ID</th>
<th>Format</th>
<th>Example</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>L&lt;CTC&gt;&lt;3CharRandomID&gt;&lt;routingID&gt;&lt;date&gt;&lt;time&gt;</td>
<td>L13STSESD0020131191203070</td>
<td>1. CTC for Solicited Request – 151 (period in the CTC is removed in the Unique ID) 2. CTC for Solicited Response – 152 (period in the CTC is removed in the Unique ID) 3. CTC for UnSolicited Response – 153 (period in the CTC is removed in the Unique ID) 4. routingId -- RCs mailbox ID 5. date -- Date in MMDDYY format 6. time - Time in HHMMSS0</td>
</tr>
</tbody>
</table>

22.1.1 Unique ID Generation

Table 56: Unique ID Generation API Methods describes the API Methods available to generate the Unique ID.

Table 56: Unique ID Generation API Methods

<table>
<thead>
<tr>
<th>No.</th>
<th>Class/Interface Name</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ICDTUtils</td>
<td>Public String randomAlphaNumericValue() throws Exception;</td>
<td>This method is used to generate the 5-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.</td>
</tr>
<tr>
<td>2</td>
<td>ICDTUtils</td>
<td>String generateUniqueID(String randomAlphaNumericValue_.</td>
<td>This method is used to generate a Unique ID for each ICDT Request and ICDT Response sent from the RC.</td>
</tr>
</tbody>
</table>
### CMS XLC API Methods

<table>
<thead>
<tr>
<th>No.</th>
<th>Class/Interface Name</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>String date_, String timestamp_) throws Exception;</td>
<td>Parameter: 1. randomAlphaNumericValue_ - The value created using randomAlphaNumericValue() method; 2. Date_ - The current system date in MMddyy format. 3. Timestamp_ - The current system timestamp in HHmmss format. Returns: The String Object of Unique ID value for the ICDT Request and ICDT Response.</td>
</tr>
</tbody>
</table>

#### 22.2 ICDT Request

Table 57: ICDT Request API Methods details the methods available to submit the ICDT Request by different Review Contractors.

<table>
<thead>
<tr>
<th>No.</th>
<th>Class/Interface Name</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ICDTRequestProcessorImpl</td>
<td>MessageDTO generateICDTPackage(ICDT MetadataDTO_, boolean isUniqueldCreate_)</td>
<td>This method is used to create the Request XML based on the object values. icdtMetadataDTO_ - Object values to generate the Request XML file isUniqueldCreate_ - 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 MessageDTO Object - The messageDTO Object contains message, status, list of errors and desc, randomNumber, Request ID, and filename.</td>
</tr>
<tr>
<td>2</td>
<td>ICDTRequestProcessorImpl</td>
<td>MessageDTO generateICDTPackage(String icdtSolicitedRequestXMLFileLocation_, boolean isUniqueldCreate_, ICollection&lt;FileInfo&gt; icdtAttachmentFiles_)</td>
<td>This method is used to create the ICDTRequest XML based on the absolute path of the file. icdtSolicitedRequestXMLFileLocation_ - The absolute file path of the request XML isUniqueldCreate_ - 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 request package.</td>
</tr>
<tr>
<td>No.</td>
<td>Class/Interface Name</td>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>3</td>
<td>ICDTRequestProcessorImpl</td>
<td>MessageDTO generateICDTPackage(FileInfo icdtSolicitedRequestXMLFile Obj_, boolean isUniqueldCreate_, ICollection&lt;FileInfo&gt; icdtAttachmentFiles_)</td>
<td>This method is used to create the ICDT Request XML based on the file object icdtSolicitedRequestXMLFileObj_. The request XML file object passed by RCs isUniqueldCreate_ - 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 request package Returns MessageDTO Object - The messageDTO Object contains message, status, list of errors and desc, randomNumber, Request ID, and filename</td>
</tr>
<tr>
<td>4</td>
<td>ICDTRequestProcessorImpl</td>
<td>ICDTRequest readRequestXMLFile(String xmlFileNamewithAbsolutePath_)</td>
<td>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</td>
</tr>
</tbody>
</table>

### 22.3 ICDT Solicited Response

Table 58: ICDT Solicited Response API Methods details the methods available for sending ICDT Response by RCs.

#### Table 58: ICDT Solicited Response API Methods

<table>
<thead>
<tr>
<th>No.</th>
<th>Class/Interface Name</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ICDTSolicitedResponseProcessorImpl</td>
<td>MessageDTO generateICDTPackage(ICDT MetadataDTO_, boolean isUniqueldCreate_)</td>
<td>This method is used to generate the Solicited Response based on the object values icdtMetadataDTO_ - Object values for generating the Solicited Response isUniqueldCreate_ - 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 Response ID to the API</td>
</tr>
<tr>
<td>No.</td>
<td>Class/Interface Name</td>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2</td>
<td>ICDTSolicitedResponseProcessor</td>
<td>MessageDTO generateICDTPackage(String icdtSolicitedResponseXMLFileLocation_, boolean isUniqueIdCreate, ICollection&lt;FileInfo&gt; icdtAttachmentFiles_)</td>
<td>This method is used to create the Response package based on the absolute path of the file. icdtSolicitedResponseXMLFileLocation_ - The absolute file path of the request 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 Response ID to the API icdtAttachmentFiles_ - List of attachment files to be included in the Solicited response package Returns MessageDTO Object - The message Object contains message, status, list of errors and desc, randomNumber, Request ID, and filename</td>
</tr>
<tr>
<td>3</td>
<td>ICDTSolicitedResponseProcessor</td>
<td>MessageDTO generateICDTPackage(FileInfo icdtSolicitedRequestXMLFileObj_, boolean isUniqueIdCreate, ICollection&lt;FileInfo&gt; icdtAttachmentFiles_)</td>
<td>This method is used to create the Response package 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 MessageDTO Object - The message Object contains message, status, list of errors and desc, randomNumber, Request ID, and filename</td>
</tr>
<tr>
<td>4</td>
<td>ICDTSolicitedResponseProcessor</td>
<td>ICDTResponse readSolicitedResponseXMLFile(String xmlFileNamewithAbsolutePath_)</td>
<td>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</td>
</tr>
</tbody>
</table>
### 22.4 ICDT Unsolicited Response

Table 59: ICDT Unsolicited Response API Methods details the methods available for sending ICDT Response by RCs.

<table>
<thead>
<tr>
<th>No.</th>
<th>Class/Interface Name</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ICDTUnsolicitedResponseProcessor Impl</td>
<td>MessageDTO generateICDTPackage(ICDTMetadataDTO_, boolean isUniqueIdCreate_)</td>
<td>This method is used to generate the Unsolicited Response based on the object values icdtMetadataDTO_ - 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 Response ID to the API Returns MessageDTO Object - The message Object contains message, status, list of errors and desc, randomNumber, Response ID, and the filename</td>
</tr>
<tr>
<td>2</td>
<td>ICDTUnsolicitedResponseProcessor Impl</td>
<td>MessageDTO generateICDTPackage(String icdtUnsolicitedResponseXMLFileLocation_, boolean isUniqueIdCreate, ICollection&lt;FileInfo&gt; icdtAttachmentFiles_)</td>
<td>This method is used to create the Response package based on the absolute path of the file. icdtUnsolicitedResponseXMLFileLocation_ - The absolute file path of the request 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 Response ID to the API icdtAttachmentFiles_ - List of attachment files to be included in the UnSolicited response package Returns MessageDTO Object - The message Object contains message, status, list of errors and desc, randomNumber, Request ID, and the filename</td>
</tr>
<tr>
<td>No.</td>
<td>Class/Interface Name</td>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>3</td>
<td>ICDTSolicitedResponseProcessorImpl</td>
<td>generateICDTPackage(FileInfo icdtUnsolicitedRequestXMLFileObj_, boolean isUniqueIdCreate_, ICollection&lt;FileInfo&gt; icdtAttachmentFiles_)</td>
<td>This method is used to create the Response package based on the file object icdtUnsolicitedRequestXMLFileObj_. The request 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 MessageDTO Object - The message Object contains message, status, list of errors and desc, randomNumber, Request ID, and filename</td>
</tr>
<tr>
<td>4</td>
<td>ICDTSolicitedResponseProcessorImpl</td>
<td>readUnsolicitedResponseXML(String xmlFileNamewithAbsolutePath_)</td>
<td>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</td>
</tr>
</tbody>
</table>

Refer to the Figure 20: High-level ICDT API Architecture for all the classes and method signatures for ICDT Request Solicited Response and Unsolicited Response.
Figure 20: High-level ICDT API Architecture

```
@Message
generateICDTPackage(
    ICDTMetaBean
    boolean isUniqueIDCreate_)

@Message
generateICDTPackage(
    String
    ICDTMetaBean fileLocation_,
    boolean isUniqueIDCreate_,
    Collection<File>
    attachmentFiles_)

@Message
generateICDTPackage(
    File
    ICDTMetaBean xmlFileObject_,
    boolean isUniqueIDCreate_,
    Collection<File>
    attachmentFiles_)

ICDTPResponse
readICDTRequestXMLFile(String
    xmlFileNamewithAbsolutePath_)
```

The Message Object that has a status of true or false based on the successful creation of the ICDT Request Package. The Message object also returns the list of the Validation Failure Bean object in case of a validation failure with the data provided by the user.

```java
class Message
    String message;
    boolean status;
    List<ValidationFailureBean> validationFailureBeanList;
    String randomAlphaNumeric;
    String uniqueId;
    String fileName;
```
22.5 ICDT 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.
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 60: Administrative Error Response API Methods.

<table>
<thead>
<tr>
<th>No.</th>
<th>Class/Interface Name</th>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ICDTAdminErrorNotifi cationProcessImpl</td>
<td>public MessageDTO generateICDTNotification(NotificationDTO notification)</td>
<td>This method is used to generate the administrative error response based on the object values notification - Object that holds the administrative error response details</td>
</tr>
<tr>
<td>2</td>
<td>ICDTAdminErrorNotifi cationProcessImpl</td>
<td>public ICDTCommunication readICDTNotification(string fileNameWithAbsolutePath _)</td>
<td>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.</td>
</tr>
</tbody>
</table>

22.6 Pre-Pay and Post-Pay eMDR Letters

Table 61: Pre-Pay and Post-Pay API Methods lists the API methods for generating the Pre-Pay and Post-Pay eMDR letter packages.
<table>
<thead>
<tr>
<th>No.</th>
<th>Class/Interface Name</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>ESMDManualSubmitADReMDRPrePayImpl</td>
<td>public Message generateEMDRPrepayPackage(String eMDRPrepayLetterDirectory_)</td>
<td>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.</td>
</tr>
<tr>
<td>2.</td>
<td>ESMDManualSubmitADReMDRPostPayImpl</td>
<td>public Message generateEMDRPostpayPackage(Object eMDRPostPayStructuredBean_, Collection&lt;File&gt; eMDRPDFFiles_)</td>
<td>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.</td>
</tr>
<tr>
<td>No.</td>
<td>Class/Interface Name</td>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------</td>
<td>--------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 3   | ESMDManualSubmitADRReMDRPostPayImpl | public Message generateEMDRPostpayPackage (File eMDRStructuredXMLFile_, Collection<File> eMDRPDFFiles_) | Used to generate the ADR Post-Pay letters zip package that contains PDF file, structured XML file, and the 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. |
| 4   | ESMDManualSubmitADRReMDRPostPayImpl | 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. eMDRPDFFilesPath_ - 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. |

### 22.7 Logs

The RC .NET Client Sample application is a Windows desktop application. All log messages are written to the RC Client.log file.
# 23. Error Codes

## 23.1 Errors: esMD to RC

Table 62: Error Codes Sent from the esMD to RC lists all the error codes sent from the esMD to the RC.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>File not formatted correctly</td>
</tr>
<tr>
<td>118</td>
<td>ESMD validation error: Error encountered while saving ReviewContractorPickUpStatus data</td>
</tr>
<tr>
<td>144</td>
<td>Failure in sending the Administrative error response to HIH</td>
</tr>
<tr>
<td>145</td>
<td>Failure in sending the Administrative PA response to HIH</td>
</tr>
<tr>
<td>146</td>
<td>Failure in sending the Pickup notification to HIH</td>
</tr>
<tr>
<td>222</td>
<td>Number of Records Mismatch</td>
</tr>
<tr>
<td>303</td>
<td>esMD validation error: Empty file received in the response.</td>
</tr>
<tr>
<td>305</td>
<td>esMD validation error: Review Contractor PickUp Timestamp is not a valid Timestamp. Correct and resubmit.</td>
</tr>
<tr>
<td>306</td>
<td>esMD validation error: esMD Delivery Timestamp is not a valid Timestamp. Correct and resubmit.</td>
</tr>
<tr>
<td>333</td>
<td>Duplicate File Sent</td>
</tr>
<tr>
<td>501</td>
<td>Missing Contractor/Workload number</td>
</tr>
<tr>
<td>502</td>
<td>Missing esMD Transaction ID</td>
</tr>
<tr>
<td>503</td>
<td>Missing Mode of Receipt</td>
</tr>
<tr>
<td>504</td>
<td>Missing Service Trace Number</td>
</tr>
<tr>
<td>516</td>
<td>ESMD validation error: Error encountered while storing PA Review Results Response</td>
</tr>
<tr>
<td>517</td>
<td>ESMD validation error: Error encountered while fetching PA Review Results Response Notification objects</td>
</tr>
<tr>
<td>518</td>
<td>ESMD validation error: Error encountered while updating PA Review Results Response Notification objects to DB</td>
</tr>
<tr>
<td>534</td>
<td>Unzip error</td>
</tr>
<tr>
<td>535</td>
<td>Checksum error</td>
</tr>
<tr>
<td>536</td>
<td>Metadata error</td>
</tr>
<tr>
<td>537</td>
<td>Registration Request error</td>
</tr>
<tr>
<td>539</td>
<td>esMD internal system error: Unzip failure. Resubmit.</td>
</tr>
<tr>
<td>541</td>
<td>esMD validation error: Transaction ID is invalid. Correct and resubmit.</td>
</tr>
<tr>
<td>542</td>
<td>ESMD validation error: Outbound Content Type Code does not match Inbound Content Type Code for this transaction ID</td>
</tr>
<tr>
<td>543</td>
<td>ESMD validation error: RC is not authorized to use this Content Type Code</td>
</tr>
<tr>
<td>544</td>
<td>esMD validation error: Reason Code is required when Decision Indicator is N or R. Correct and resubmit.</td>
</tr>
<tr>
<td>545</td>
<td>esMD validation error: Total number of Reason Codes cannot exceed 25. Reduce the number of Reason Codes and Resubmit.</td>
</tr>
<tr>
<td>546</td>
<td>ESMD validation error: Warning: Total number of Denial Codes exceeds 25</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>547</td>
<td>ESMD validation error: Denial Code Description must be 2 - 256 positions</td>
</tr>
<tr>
<td>555</td>
<td>ESMD validation error: Content Type Code does not exist</td>
</tr>
<tr>
<td>556</td>
<td>ESMD validation error: Decision Indicator must be A, N, M or R. Correct and resubmit.</td>
</tr>
<tr>
<td>557</td>
<td>ESMD validation error: Review Contractor Unique Tracking Number must be 1 - 50 alphanumeric characters with no special characters. Correct and resubmit.</td>
</tr>
<tr>
<td>558</td>
<td>ESMD validation error: Reason Code does not exist in the esMD database. Correct and resubmit.</td>
</tr>
<tr>
<td>559</td>
<td>ESMD validation error: Denial Code Description is required, if Decision Indicator is N or R and Denial Code is &quot;Other.&quot;</td>
</tr>
<tr>
<td>560</td>
<td>ESMD validation error: Submission is infected with virus. Correct and resubmit.</td>
</tr>
<tr>
<td>562</td>
<td>ESMD validation error: Unique Tracking Number is required when Decision Indicator is A, N, or M. Correct and resubmit.</td>
</tr>
<tr>
<td>563</td>
<td>Error encountered while validating the PA Review Results Response</td>
</tr>
<tr>
<td>564</td>
<td>Error occurred while storing the RC Response Virus Scan Result</td>
</tr>
<tr>
<td>565</td>
<td>ESMD Internal System Error: Unable to process your response. Correct and resubmit.</td>
</tr>
<tr>
<td>566</td>
<td>ESMD validation error: A required element is either missing, has an invalid element format, or has an invalid length. Correct and resubmit.</td>
</tr>
<tr>
<td>567</td>
<td>ESMD validation error: A Decision Indicator of 'M' is invalid for PMD PA or DMEPOS response. Provide a valid Decision Indicator and resubmit.</td>
</tr>
<tr>
<td>569</td>
<td>ESMD validation error: Number of Approved Units, Approved Service Date, and Date Range are not allowed for this response. Correct and resubmit.</td>
</tr>
<tr>
<td>572</td>
<td>ESMD validation error: Approved Service End Date is less than or equal to Approved Service Start Date. Correct and resubmit.</td>
</tr>
<tr>
<td>576</td>
<td>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.</td>
</tr>
<tr>
<td>600</td>
<td>ESMD validation error: Duplicate Reason Codes found. Correct and resubmit.</td>
</tr>
<tr>
<td>601</td>
<td>Invalid Contractor/Workload Number</td>
</tr>
<tr>
<td>602</td>
<td>ESMD validation error: Approved Service Date must be greater than or equal to current system date.</td>
</tr>
<tr>
<td>603</td>
<td>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.</td>
</tr>
<tr>
<td>604</td>
<td>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.</td>
</tr>
<tr>
<td>605</td>
<td>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.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>609</td>
<td>esMD Virus Scanning service is unavailable. Retry later</td>
</tr>
<tr>
<td>610</td>
<td>esMD validation error: Empty File Received in the Response. Correct and resubmit.</td>
</tr>
<tr>
<td>611</td>
<td>esMD validation error: Multiple Files Received in the Response. Resubmit with only one file</td>
</tr>
<tr>
<td>612</td>
<td>esMD validation error: Approved Service Date or Approved Service Date Range and Approved Unit are not allowed for this response. Correct and resubmit.</td>
</tr>
<tr>
<td>613</td>
<td>esMD validation error: Administrative error code is invalid. Correct and resubmit.</td>
</tr>
<tr>
<td>614</td>
<td>esMD validation error: Approved Service End Date is less than the Current Date. Correct and resubmit</td>
</tr>
<tr>
<td>615</td>
<td>esMD validation error: Invalid error in the pickup notification. Correct and resubmit.</td>
</tr>
<tr>
<td>616</td>
<td>esMD validation error: Intended Recipient OID is deactivated and cannot accept response. Correct and resubmit.</td>
</tr>
<tr>
<td>617</td>
<td>esMD validation error: Mailbox ID in the response does not match with the Mailbox ID that the request was sent.</td>
</tr>
<tr>
<td>618</td>
<td>esMD validation error: Intended Recipient OID is deactivated and cannot accept response.</td>
</tr>
<tr>
<td>619</td>
<td>esMD validation error: Mailbox ID in the response does not match with the Mailbox ID that the request was sent.</td>
</tr>
<tr>
<td>620</td>
<td>esMD validation error: Invalid Review Response Creation Time format</td>
</tr>
<tr>
<td>621</td>
<td>esMD validation error: Invalid review Response Submission Time Format</td>
</tr>
<tr>
<td>622</td>
<td>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.</td>
</tr>
<tr>
<td>623</td>
<td>esMD validation error: Both Approved Service Date and Approved Service Date range cannot exists in same response. Correct and Resubmit.</td>
</tr>
<tr>
<td>624</td>
<td>esMD validation error: Approved Service Start Date cannot be greater than the Approved Service End Date. Correct and resubmit</td>
</tr>
<tr>
<td>625</td>
<td>esMD validation error: Reason Code is not allowed for Decision Indicator A. Correct and resubmit.</td>
</tr>
<tr>
<td>631</td>
<td>esMD validation error: A Review or Error Response is not allowed for this transaction.</td>
</tr>
<tr>
<td>632</td>
<td>esMD validation error: Total number of Industry Codes cannot exceed 5. Reduce the number of Industry Codes and resubmit.</td>
</tr>
<tr>
<td>633</td>
<td>esMD validation error: Either HIH is not active or agreement has expired to receive the response.</td>
</tr>
<tr>
<td>634</td>
<td>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.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>637</td>
<td>esMD validation error: Outbound response received for the submission that failed for the Inbound</td>
</tr>
<tr>
<td>638</td>
<td>OUTBND_EMPTY_PICKUP_FILE_ERROR</td>
</tr>
<tr>
<td>640</td>
<td>esMD validation error: Intended recipient OID and Procedure Code is not a valid combination. Correct and resubmit</td>
</tr>
<tr>
<td>701</td>
<td>Missing esMD Transaction ID</td>
</tr>
<tr>
<td>702</td>
<td>Missing Procedure Code</td>
</tr>
<tr>
<td>703</td>
<td>Missing Decision Indicator</td>
</tr>
<tr>
<td>704</td>
<td>Missing Subscriber ID</td>
</tr>
<tr>
<td>705</td>
<td>Missing Workload Number</td>
</tr>
<tr>
<td>706</td>
<td>Missing Service Trace Number</td>
</tr>
<tr>
<td>800</td>
<td>ESMD validation error: Error occurred while storing the Review Contractor Status PickUp</td>
</tr>
<tr>
<td>801</td>
<td>Invalid esMD Transaction ID</td>
</tr>
<tr>
<td>801</td>
<td>ESMD validation error: Error occurred while validating the Review Contractor PickUp Status Data</td>
</tr>
<tr>
<td>802</td>
<td>Invalid Procedure Code</td>
</tr>
<tr>
<td>803</td>
<td>Invalid Decision Indicator</td>
</tr>
<tr>
<td>804</td>
<td>Invalid Subscriber ID</td>
</tr>
<tr>
<td>805</td>
<td>Invalid Workload Number</td>
</tr>
<tr>
<td>806</td>
<td>Invalid Service Trace Number</td>
</tr>
<tr>
<td>901</td>
<td>Invalid AAA codes</td>
</tr>
<tr>
<td>902</td>
<td>Invalid PA Program Reason Code</td>
</tr>
<tr>
<td>903</td>
<td>Invalid Review Decision Reason Code</td>
</tr>
<tr>
<td>904</td>
<td>esMD Validation Error: The ICDT Request Zip File received from RC is Zero Byte in size. Correct and resubmit.</td>
</tr>
<tr>
<td>905</td>
<td>esMD validation error: The Checksum received does not match the Checksum in the zip file. Correct and resubmit.</td>
</tr>
<tr>
<td>907</td>
<td>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.</td>
</tr>
<tr>
<td>908</td>
<td>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.</td>
</tr>
<tr>
<td>909</td>
<td>esMD validation error: The Content type code received for the eMDR Request is incorrect. Correct and resubmit.</td>
</tr>
<tr>
<td>910</td>
<td>esMD Validation Error: The ICDT Request Zip File received from RC exceeded the maximum allowable size. Correct and resubmit.</td>
</tr>
<tr>
<td>911</td>
<td>esMD Validation Error: The Unique ID received in the eMDR Request Metadata XML File already exists in the database. Correct and resubmit.</td>
</tr>
<tr>
<td>912</td>
<td>esMD validation error: The name of the document does not match the document ID in the Metadata zip file. Correct and resubmit.</td>
</tr>
<tr>
<td>913</td>
<td>esMD validation error: The Size of Document received does not match with the Size of Document in the zip file. Correct and resubmit.</td>
</tr>
<tr>
<td>914</td>
<td>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.</td>
</tr>
<tr>
<td>915</td>
<td>esMD validation error: Unable to encode the response</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>916</td>
<td>esMD Validation Error: The eMDR request Zip file extraction failed. Correct and resubmit.</td>
</tr>
<tr>
<td>917</td>
<td>esMD validation error: Unable to parse request XML file. Correct XML and Resubmit.</td>
</tr>
<tr>
<td>918</td>
<td>esMD validation error: The Sender OID received from the Review Contractor for the eMDR Request is Invalid. Correct and resubmit.</td>
</tr>
<tr>
<td>919</td>
<td>esMD validation error: RC type provided in the metadata is Invalid for the eMDR request. Correct and resubmit.</td>
</tr>
<tr>
<td>920</td>
<td>esMD validation error: The name of the document does not match the document ID in the Metadata zip file. Correct and resubmit.</td>
</tr>
<tr>
<td>921</td>
<td>esMD validation error: The HIH OID received from the Review Contractor for the eMDR request is Invalid. Correct and resubmit.</td>
</tr>
<tr>
<td>922</td>
<td>esMD Validation Error: A Duplicate RC Unique ID received in the ADR Review Result Response XML File already exists. Correct and resubmit.</td>
</tr>
<tr>
<td>923</td>
<td>esMD Validation Error: The File received from RC exceeded the maximum allowable size for ADR Review Result Response. Correct and resubmit.</td>
</tr>
<tr>
<td>926</td>
<td>esMD validation error: The name of the document does not match the document ID in the Metadata zip file. Correct and resubmit.</td>
</tr>
<tr>
<td>928</td>
<td>esMD validation error: The Checksum received does not match the Checksum in the ADR Review Result Response zip file. Correct and resubmit.</td>
</tr>
<tr>
<td>929</td>
<td>esMD validation error: The HIH OID received from the Review Contractor for the ADR Review Result Response is Invalid. Correct and resubmit.</td>
</tr>
<tr>
<td>930</td>
<td>esMD validation error: The Sender OID received from the Review Contractor for the ADR Review Result Response is Invalid. Correct and resubmit.</td>
</tr>
<tr>
<td>931</td>
<td>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.</td>
</tr>
<tr>
<td>932</td>
<td>esMD validation error: The Content type code received for the ADR Review Result Response is incorrect. Correct and resubmit.</td>
</tr>
<tr>
<td>933</td>
<td>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.</td>
</tr>
<tr>
<td>934</td>
<td>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.</td>
</tr>
<tr>
<td>936</td>
<td>esMD Validation Error: The Document Unique ID received from RC for the eMDR Request exceeds the maximum length. Correct and resubmit.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>937</td>
<td>esMD Validation Error: The Document Unique ID received from RC for the ADR Review Result Response exceeds the maximum length. Correct and resubmit.</td>
</tr>
<tr>
<td>938</td>
<td>esMD validation error: The MIME type is missing in the ADR Review Result Response Process Metadata. Correct and resubmit.</td>
</tr>
<tr>
<td>939</td>
<td>esMD validation error: Unable to parse {0} XML file. Correct XML and Resubmit</td>
</tr>
<tr>
<td>940</td>
<td>esMD Validation Error: The {0} received in the {1} XML File already exists in the database. Correct and resubmit</td>
</tr>
<tr>
<td>941</td>
<td>esMD validation error: The Receiver OID received from the Review Contractor for the {0} is non-participating. Correct and resubmit</td>
</tr>
<tr>
<td>942</td>
<td>esMD validation error: The Sender OID received from the Review Contractor for the {0} is non-participating. Correct and resubmit</td>
</tr>
<tr>
<td>943</td>
<td>esMD validation error: The Content type code received for the {0} is incorrect. Correct and resubmit</td>
</tr>
<tr>
<td>944</td>
<td>esMD validation error: The combination of Sender OID and the Content type code received in the {0} from RC is incorrect. Correct and resubmit</td>
</tr>
<tr>
<td>945</td>
<td>esMD validation error: The combination of Receiver OID and the Content type code received in the {0} from RC is incorrect. Correct and resubmit.</td>
</tr>
<tr>
<td>946</td>
<td>esMD Validation Error: The Claim ID received in the {0} is Invalid. Correct and Resubmit</td>
</tr>
<tr>
<td>947</td>
<td>esMD Validation Error: Missing Claim ID in the {0}. Correct and Resubmit</td>
</tr>
<tr>
<td>948</td>
<td>esMD Validation Error: The Case ID received in the {0} is Invalid. Correct and Resubmit.</td>
</tr>
<tr>
<td>949</td>
<td>esMD Validation Error: The NPI received in the {0} is Invalid. Correct and Resubmit</td>
</tr>
<tr>
<td>950</td>
<td>esMD Validation Error: Missing NPI in the {0}. Correct and Resubmit</td>
</tr>
<tr>
<td>951</td>
<td>esMD Validation Error: The HICN received in the {0} is Invalid. Correct and Resubmit</td>
</tr>
<tr>
<td>952</td>
<td>esMD Validation Error: Missing HICN in the {0}. Correct and Resubmit</td>
</tr>
<tr>
<td>953</td>
<td>esMD Validation Error: The OCN received in the {0} is Invalid. Correct and Resubmit</td>
</tr>
<tr>
<td>954</td>
<td>esMD Validation Error: Missing OCN in the {0}. Correct and Resubmit</td>
</tr>
<tr>
<td>955</td>
<td>esMD Validation Error: Sender OID and Receiver OID received in the {0} match. Correct and Resubmit</td>
</tr>
<tr>
<td>956</td>
<td>esMD Validation Error: Internal System issue</td>
</tr>
<tr>
<td>957</td>
<td>esMD validation error: The Checksum received does not match the Checksum in the Zip file. Correct and resubmit</td>
</tr>
<tr>
<td>958</td>
<td>esMD validation error: The MIME type is missing in the {0} Metadata. Correct and resubmit.</td>
</tr>
<tr>
<td>959</td>
<td>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</td>
</tr>
<tr>
<td>960</td>
<td>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</td>
</tr>
<tr>
<td>961</td>
<td>esMD Validation error: The Mime type {0} is invalid. Correct and resubmit</td>
</tr>
<tr>
<td>962</td>
<td>esMD Validation Error: Missing NPI in the ICDT Request. Correct and Resubmit.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>963</td>
<td>esMD Validation error: The name of the document does not match the name of the document received in the {0} in the zip file</td>
</tr>
<tr>
<td>964</td>
<td>esMD Validation error: The Request ID provided in the {0} is either missing or not exist in the esMD database</td>
</tr>
<tr>
<td>965</td>
<td>esMD Validation Error: Invalid Admin Error Code received from the Review Contractor</td>
</tr>
<tr>
<td>966</td>
<td>esMD Validation Error: The ICDT Request Zip File received from RC exceeded the maximum allowable size. Correct and resubmit</td>
</tr>
<tr>
<td>967</td>
<td>esMD Validation Error: The ICDT Request Zip File received from RC is Zero Byte in size. Correct and resubmit</td>
</tr>
<tr>
<td>968</td>
<td>esMD Validation Error: The {0} Zip file extraction failed. Correct and resubmit</td>
</tr>
<tr>
<td>969</td>
<td>esMD Validation Error: The documentation type received in the {0} XML is invalid. Correct and Resubmit.</td>
</tr>
<tr>
<td>970</td>
<td>esMD validation error: The MIME type is missing in the esMD Process Metadata. Correct and resubmit.</td>
</tr>
<tr>
<td>971</td>
<td>esMD Validation Error: The Document Unique ID received from RC for the ADR Review Result Letter exceeds the maximum length. Correct and resubmit.</td>
</tr>
<tr>
<td>972</td>
<td>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.</td>
</tr>
<tr>
<td>1032</td>
<td>esMD Validation Error: esMD system missed or received more than one {0} from the RC for the {1}. Correct and resubmit</td>
</tr>
<tr>
<td>1034</td>
<td>esMD Validation Error: Unable to parse the eMDR Process Metadata XML File for {0}. Correct and resubmit.</td>
</tr>
<tr>
<td>1035</td>
<td>esMD Validation Error: The Unique Id {0} received in the eMDR Process Metadata XML File for the {1} already exists in the database. Correct and resubmit.</td>
</tr>
<tr>
<td>1036</td>
<td>esMD Validation Error: The number of documents in the eMDR Process Metadata XML File does not match the number of documents in the zip file for the {0}.</td>
</tr>
<tr>
<td>1037</td>
<td>esMD Validation Error: The Checksum received does not match the Checksum calculated for one or more attachments in the zip file for the {0}. Correct and resubmit.</td>
</tr>
<tr>
<td>1038</td>
<td>esMD Validation Error: The name of the document for one or more attachments in the zip file does not match the name of the document in the {0} for the {1}. Correct and resubmit.</td>
</tr>
<tr>
<td>1039</td>
<td>esMD Validation Error: The size of the document for one or more attachments in the zip file does not match the size of the document received in the Metadata for the {0}. Correct and resubmit.</td>
</tr>
<tr>
<td>1040</td>
<td>esMD validation error: The Sender OID received from the RC for the {0} is invalid. Correct and resubmit.</td>
</tr>
<tr>
<td>1041</td>
<td>esMD Validation Error: The combination of Review Contractor OID and Content type code received for the {0} from RC is incorrect. Correct and resubmit.</td>
</tr>
<tr>
<td>1049</td>
<td>esMD Validation Error: Date of Service (To) is less then Date of Service (From) in the {0} for the {1}. Correct and resubmit.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>1050</td>
<td>esMD Validation Error: The NPI {0} received from the Review Contractor is missing provider consent.</td>
</tr>
<tr>
<td>1051</td>
<td>esMD Validation Error: The NPI {0} received from the Review Contractor is not associated with any HIH in esMD. Correct and resubmit.</td>
</tr>
<tr>
<td>1052</td>
<td>esMD Validation Error: The combination of HIH OID and Content type code received for the {0} from RC is incorrect. Correct and resubmit.</td>
</tr>
<tr>
<td>1053</td>
<td>esMD Validation Error: Prefix for the {0} File Name does not match with the {1} in the {2} for the {3}. Correct and resubmit.</td>
</tr>
<tr>
<td>1054</td>
<td>esMD Validation Error: Invalid File extension for {0} received from RC for the {1}. Correct and resubmit.</td>
</tr>
<tr>
<td>1055</td>
<td>RC Client API Validation Error: {0} file size must be greater than {1}</td>
</tr>
<tr>
<td>1056</td>
<td>RC Client API Validation Error: {0} file size must not exceed {1}</td>
</tr>
<tr>
<td>1057</td>
<td>RC Client API Validation Error: Unable to parse the {0} for the {1}</td>
</tr>
<tr>
<td>1058</td>
<td>esMD Validation Error: The combination of HIH OID and Content type code received for the {0} from RC is incorrect. Correct and resubmit.</td>
</tr>
<tr>
<td>1059</td>
<td>The file attached to ADR eMDR Letter File received from Review Contractor is not a PDF or XML file</td>
</tr>
<tr>
<td>1063</td>
<td>esMD Validation Error: Either The Unique Letter ID Is Invalid OR Missing. Correct and resubmit.</td>
</tr>
<tr>
<td>1064</td>
<td>esMD Validation Error: The NPI {0} received from the Review Contractor is not associated with any HIH in esMD. Correct and resubmit.</td>
</tr>
<tr>
<td>1066</td>
<td>esMD validation error: The Sender OID received from the RC for the {0} is invalid. Correct and resubmit.</td>
</tr>
<tr>
<td>1068</td>
<td>esMD Validation Error: The MIME type is either missing or invalid in the ADR esMD Letters File Metadata XML File. Correct and resubmit.</td>
</tr>
<tr>
<td>1069</td>
<td>esMD Validation Error: The Document Unique ID received from the RC for the ADR esMD Letters File exceeds the maximum length. Correct and resubmit.</td>
</tr>
<tr>
<td>1070</td>
<td>esMD Validation Error: NPPES gateway response time out received for {0}. Please resubmit.</td>
</tr>
<tr>
<td>1071</td>
<td>esMD Validation error: EITHER THE TYPE OF EMDR IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1072</td>
<td>esMD Validation error: EITHER THE ANALYSIS ID IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1073</td>
<td>esMD Validation error: EITHER THE LETTER DATE IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1074</td>
<td>esMD Validation error: EITHER THE ORGANIZATION NAME/RC DETAILS IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1075</td>
<td>esMD Validation error: EITHER THE RC ADDRESS 1 IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1076</td>
<td>esMD Validation error: EITHER THE RC CITY IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1077</td>
<td>esMD Validation error: EITHER THE RC STATE IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1078</td>
<td>esMD Validation error: EITHER THE RC ZIP CODE IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1079</td>
<td>esMD Validation error: EITHER THE SENDER OR ORGANIZATION NAME IS INVALID OR MISSING</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>1080</td>
<td>esMD Validation error: EITHER THE PROVIDER LAST NAME OR ORGANIZATION NAME IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1081</td>
<td>esMD Validation error: EITHER THE PROVIDER ADDRESS 1 IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1082</td>
<td>esMD Validation error: EITHER THE PROVIDER CITY IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1083</td>
<td>esMD Validation error: EITHER THE PROVIDER STATE IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1084</td>
<td>esMD Validation error: EITHER THE PROVIDER ZIP CODE IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1085</td>
<td>esMD Validation error: EITHER THE PROVIDER NPI IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1086</td>
<td>esMD Validation error: EITHER THE RESPONSE DATE IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1087</td>
<td>esMD Validation error: EITHER THE JURISDICTION OR ZONE INVALID OR MISSING</td>
</tr>
<tr>
<td>1088</td>
<td>esMD Validation error: EITHER THE PROGRAM NAME IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1089</td>
<td>esMD Validation error: EITHER THE DOCUMENT CODE IS MISSING OR THE FORMAT IS INVALID</td>
</tr>
<tr>
<td>1090</td>
<td>esMD Validation error: EITHER THE CLAIM ID IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1091</td>
<td>esMD Validation error: EITHER THE BENEFICIARY ID IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1092</td>
<td>esMD Validation error: EITHER THE BENEFICIARY NAME IS INVALID OR MISSING</td>
</tr>
<tr>
<td>1094</td>
<td>esMD Validation Error: More than 5 Procedure Codes are received for a single error description. Maximum 5 Procedure codes are allowed. Correct and resubmit.</td>
</tr>
<tr>
<td>1095</td>
<td>esMD Validation Error: Procedure Code received in the RC Reject Response is not of a valid length and format. Correct and resubmit.</td>
</tr>
<tr>
<td>1098</td>
<td>esMD Validation Error: Duplicate Procedure Codes received in RC Reject response. Correct and resubmit.</td>
</tr>
</tbody>
</table>

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

### 23.2 Errors: RC to esMD

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

1. Administrative Errors.
2. Pickup Errors.

---

23.2 Errors: RC to esMD

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

1. Administrative Errors.
2. Pickup Errors.
23.2.1 Administrative Errors:

Table 63: Administrative Error Codes lists the error codes used to report unexpected errors related to the payload received in a downloaded file from the esMD system. For more details, please refer to section 12.2.4 Administrative Error Response to Inbound Submissions.

<table>
<thead>
<tr>
<th>Administrative Error</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>corrupt files/cannot read files</td>
<td>ESMD_410</td>
<td>ESMD_410- Administrative Error (corrupt files/cannot read files).</td>
</tr>
<tr>
<td>Submission Sent to Incorrect RC</td>
<td>ESMD_411</td>
<td>ESMD_411- Administrative Error (Submission Sent to Incorrect RC).</td>
</tr>
<tr>
<td>Virus Found</td>
<td>ESMD_412</td>
<td>ESMD_412- Administrative Error (Virus Found).</td>
</tr>
<tr>
<td>Other</td>
<td>ESMD_413</td>
<td>ESMD_413- Administrative Error (Other).</td>
</tr>
<tr>
<td>Incomplete File</td>
<td>ESMD_414</td>
<td>ESMD_414- Administrative Error (Incomplete File).</td>
</tr>
<tr>
<td>Unsolicited Response</td>
<td>ESMD_415</td>
<td>ESMD_415- Administrative Error (Unsolicited Response).</td>
</tr>
<tr>
<td>Documentation cannot be matched to a case/claim</td>
<td>ESMD_416</td>
<td>ESMD_416- Administrative Error (Documentation cannot be matched to a case/claim).</td>
</tr>
<tr>
<td>Duplicate</td>
<td>ESMD_417</td>
<td>ESMD_417- Administrative Error (Duplicate).</td>
</tr>
<tr>
<td>The date(s) of service on the cover sheet received is missing or invalid</td>
<td>GEX10</td>
<td>The date(s) of service on the cover sheet received is missing or invalid.</td>
</tr>
<tr>
<td>The NPI on the cover sheet received is missing or invalid</td>
<td>GEX11</td>
<td>The NPI on the cover sheet received is missing or invalid.</td>
</tr>
<tr>
<td>The state where services were provided is missing or invalid on the cover sheet received</td>
<td>GEX12</td>
<td>The state where services were provided is missing or invalid on the cover sheet received.</td>
</tr>
<tr>
<td>The Medicare ID on the cover sheet received is missing or invalid</td>
<td>GEX13</td>
<td>The Medicare ID on the cover sheet received is missing or invalid.</td>
</tr>
<tr>
<td>The billed amount on the cover sheet received is missing or invalid</td>
<td>GEX14</td>
<td>The billed amount on the cover sheet received is missing or invalid.</td>
</tr>
<tr>
<td>The contact phone number on the cover sheet received is missing or invalid</td>
<td>GEX15</td>
<td>The contact phone number on the cover sheet received is missing or invalid.</td>
</tr>
<tr>
<td>The beneficiary name on the cover sheet received is missing or invalid</td>
<td>GEX16</td>
<td>The beneficiary name on the cover sheet received is missing or invalid.</td>
</tr>
</tbody>
</table>
### Administrative Error

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEX17</td>
<td>The claim number on the cover sheet received is missing or invalid.</td>
</tr>
<tr>
<td>GEX18</td>
<td>The ACN on the coversheet received is missing or invalid.</td>
</tr>
</tbody>
</table>

### 23.2.2 Pickup Errors

Table 64: Pickup Error Codes lists the pickup error codes and their descriptions. These codes are used to populate the ErrorInfo object inside the error pickup notification XML. Please refer to Section 12.2.2 Error Pickup Notification for more details.

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNZIP ERROR</td>
<td>534</td>
<td>ESMD_534 – RC Client processing error (Unzip failure). Please resubmit.</td>
</tr>
<tr>
<td>METADATA ERROR</td>
<td>536</td>
<td>ESMD_536 – RC Client processing error (Metadata issue). Please resubmit.</td>
</tr>
<tr>
<td>ERROR DOCUMENT CODES</td>
<td>515</td>
<td>INVALID LINE LENGTH FOR LINE 1; EXPECTED: 1035, ACTUAL: 1021</td>
</tr>
<tr>
<td>VALIDATE FILE</td>
<td></td>
<td>INVALID LINE LENGTH FOR LINE 2; EXPECTED: 1035, ACTUAL: 1028</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INVALID LINE LENGTH FOR LINE 7; EXPECTED: 1035, ACTUAL: 1029</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INVALID LINE LENGTH FOR LINE 8; EXPECTED: 1035, ACTUAL: 1029</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INVALID LINE LENGTH FOR LINE 9; EXPECTED: 1035, ACTUAL: 1025</td>
</tr>
</tbody>
</table>

*Note: This is dynamic error message based on the edit validation.*
24. PA Requests and Responses Automation with Shared Systems

24.1 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.

24.1.1 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 24.3.1 Logical Workflow and 24.3.2 Application Workflow for detailed information on the automation processing of PA requests and responses with Shared System/Workloads.

24.1.2 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.

24.1.2.1 PA Review Response

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

24.2 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).
2. No User Interface feature will be available for eMDR and ICDT; only the API will be provided to support the eMDR and ICDT.
24.3 Automation of PA Requests/Responses – Application Workflow

24.3.1 Logical Workflow

Figure 21: esMD Shared System/Workload Integration – Logical provides an overview of the logical flow of PA Requests/Responses between esMD and Shared System/Workload.

![Figure 21: esMD Shared System/Workload Integration – Logical](image)

24.3.2 Application Workflow

Figure 22: 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 22: Information Flow – X12N 278 PA Request/Response Integration with Shared Systems
25. esMD Cloud File Transfers

Table 6: Inbound and Outbound File Formats lists the zip/XML files that will be transferred between esMD and the RCs.

Notes:

1. ES0001 is a sample mailbox number that the esMD Cloud environment uses to identify the RC and “EPP000000008983” is a sample fifteen-character alphanumeric value.

2. The esMD transaction ID will be included in the zip file name and will also be included in the RC metadata XML file.
## 26. Inbound/Outbound File Names and Data Directories

Table 65: 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>
<thead>
<tr>
<th>Data Directory</th>
<th>Program Type</th>
<th>Folder Names</th>
<th>XML File Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>ADR Request (CTC-1)</td>
<td>E_L1_GUID</td>
<td>E_L1_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>Error</td>
<td>ADR Validation error response</td>
<td>F_L1_GUID</td>
<td>F_L1_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>ADR HIH delivery notification</td>
<td>N_L1_GUID_MMD_DYY_HHMM</td>
<td>N_L1_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>ADR Pickup Notification</td>
<td>P_L1_GUID</td>
<td>P_L1_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>input</td>
<td>PWK Unsolicited documents (CTC-7)</td>
<td>E_L7_GUID</td>
<td>E_L7_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>PWK Unsolicited Validation error response</td>
<td>F_L7_GUID</td>
<td>F_L7_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>PWK Unsolicited HIH delivery notification</td>
<td>N_L7_GUID_MMD_DYY_HHMM</td>
<td>N_L7_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>PWK Unsolicited Pickup Notification</td>
<td>P_L7_GUID</td>
<td>P_L7_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>input</td>
<td>First Level Appeals (CTC-9)</td>
<td>E_L9_GUID</td>
<td>E_L9_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>First Level Appeals Validation error response</td>
<td>F_L9_GUID</td>
<td>F_L9_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>First Level Appeals HIH Delivery Notification</td>
<td>N_L9_GUID_MMD_DYY_HHMM</td>
<td>N_L9_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>Pickup Notification</td>
<td>P_L9_GUID</td>
<td>P_L9_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>Data Directory</td>
<td>Program Type</td>
<td>Folder Names</td>
<td>XML File Name</td>
<td>Notes</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>--------------</td>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td>input</td>
<td>Second Level Appeals (CTC 9.1)</td>
<td>E_L9_1_GUID</td>
<td>E_L9_1_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>Second Level Appeals Validation error response</td>
<td>F_L9_1_GUID</td>
<td>F_L9_1_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>Second Level Appeals HIH delivery notification</td>
<td>N_L1_GUID_MMD DYY_HHMM</td>
<td>N_L9_1_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Second Level Appeals Pickup Notification</td>
<td>P_L9_1_GUID</td>
<td>P_L9_1_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>input</td>
<td>ADMC (CTC 10)</td>
<td>E_L10_GUID</td>
<td>E_L10_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>ADMC Validation error response</td>
<td>F_L10_GUID</td>
<td>F_L10_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>ADMC HIH delivery notification</td>
<td>N_L10_GUID_MMD DYY_HHMM</td>
<td>N_L10_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ADMC Pickup Notification</td>
<td>P_L10_GUID</td>
<td>P_L10_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>input</td>
<td>RAC Discussion Request (CTC 11)</td>
<td>E_L11_GUID</td>
<td>E_L11_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>RAC Discussion Validation error response</td>
<td>F_L11_GUID</td>
<td>F_L11_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>RAC Discussion HIH delivery notification</td>
<td>N_L11_GUID_MMD DYY_HHMM</td>
<td>N_L11_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RAC Discussion Pickup Notification</td>
<td>P_L11_GUID</td>
<td>P_L11_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>input</td>
<td>Phone Discussion Request (CTC 11.1)</td>
<td>E_L11_1_GUID</td>
<td>E_L11_1_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>Phone Discussion Validation error response</td>
<td>F_L11_1_GUID</td>
<td>F_L11_1_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>Phone Discussion HIH delivery notification</td>
<td>N_L11_1_GUID_MMD DYY_HHMM</td>
<td>N_L11_1_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phone Discussion Pickup Notification</td>
<td>P_L11_1_GUID</td>
<td>P_L11_1_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>input</td>
<td>Ambulance (CTC 8.1)</td>
<td>E_L8_1_GUID</td>
<td>E_L8_1_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>Data Directory</td>
<td>Program Type</td>
<td>Folder Names</td>
<td>XML File Name</td>
<td>Notes</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>--------------</td>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td>error</td>
<td>Ambulance Validation error response</td>
<td>F_L8_1_GUID</td>
<td>F_L8_1_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>Ambulance HIH delivery notification</td>
<td>N_L8_1_GUID_MM DDYY_HHMM</td>
<td>N_L8_1_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>Ambulance Pickup Notification</td>
<td>P_L8_1_GUID</td>
<td>P_L8_1_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>input</td>
<td>HHPCR (CTC 8.3)</td>
<td>E_L8_3_GUID</td>
<td>E_L8_3_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>HHPCR Validation error response</td>
<td>F_L8_3_GUID</td>
<td>F_L8_3_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>HHPCR HIH delivery notification</td>
<td>N_L8_3_GUID_MM DDYY_HHMM</td>
<td>N_L8_3_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>HHPCR Pickup Notification</td>
<td>P_L8_3_GUID</td>
<td>P_L8_3_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>input</td>
<td>DMEPOS (CTC 8.4)</td>
<td>E_L8_4_GUID</td>
<td>E_L8_4_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>DMEPOS Validation error response</td>
<td>F_L8_4_GUID</td>
<td>F_L8_4_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>DMEPOS HIH delivery notification</td>
<td>N_L8_4_GUID_MM DDYY_HHMM</td>
<td>N_L8_4_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>DMEPOS Pickup Notification</td>
<td>P_L8_4_GUID</td>
<td>P_L8_4_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>input</td>
<td>X12 XDR (CTC 12)</td>
<td>E_L12_GUID</td>
<td>E_L12_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>Validation error response</td>
<td>F_L12_GUID</td>
<td>F_L12_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>HIH delivery notification</td>
<td>N_L12_GUID_MM DDYY_HHMM</td>
<td>N_L12_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>Pickup Notification</td>
<td>P_L12_GUID</td>
<td>P_L12_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>input</td>
<td>Additional Documentation X12 XDR (CTC 13)</td>
<td>E_L13_GUID</td>
<td>E_L13_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>Validation error response</td>
<td>F_L13_GUID</td>
<td>F_L13_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>HIH delivery notification</td>
<td>N_L3_GUID_MMD DDMM_HHMM</td>
<td>N_L3_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>Pickup Notification</td>
<td>P_L13_GUID</td>
<td>P_L13_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>Admin Error Response</td>
<td>D_ADM_GUID</td>
<td>D_ADM_GUID_AdminResponse.xml</td>
<td>The Folder Name and XML File Name are the</td>
</tr>
<tr>
<td>Data Directory</td>
<td>Program Type</td>
<td>Folder Names</td>
<td>XML File Name</td>
<td>Notes</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------</td>
<td>--------------</td>
<td>------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>N/A</td>
<td>Reject Review Response</td>
<td>R_PA_GUID</td>
<td>R_PA_GUID_Response.xml</td>
<td>The Folder Name and XML File Name are the same for the following Content Type Codes: CTC-8.1, CTC-8.3, CTC-8.4, and CTC-13</td>
</tr>
<tr>
<td>error</td>
<td>Virus Scan Timeout</td>
<td>Y_&lt;&lt;RandomNo&gt;&gt;</td>
<td>Y_&lt;&lt;PackageUniqueID&gt;&gt;_Virus_Scan_Gateway_Failure.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>Virus Found or Infected File</td>
<td>X_&lt;&lt;RandomNo&gt;&gt;</td>
<td>X_&lt;&lt;PackageUniqueID&gt;&gt;_Virus_Scan_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>ICDT/input</td>
<td>ICDT - Solicited Request</td>
<td>Q_UniqueID</td>
<td>&lt;&lt;ReceiverRoutingId&gt;&gt;.T.L15_1.Q&lt;&lt;esMDTransactionID&gt;&gt;.&lt;&lt;SenderRoutingID&gt;&gt;.DMMddy.THHmmssS</td>
<td>Q_&lt;&lt;RCUniqueId&gt;&gt;_ICDTSolicitedRequest.xml</td>
</tr>
<tr>
<td>ICDT/input</td>
<td>ICDT - Solicited Response</td>
<td>R_UniqueID</td>
<td>&lt;&lt;ReceiverRoutingId&gt;&gt;.T.L15_2.R&lt;&lt;esMDTransactionID&gt;&gt;.&lt;&lt;SenderRoutingID&gt;&gt;.DMMddy.THHmmssS</td>
<td>R_&lt;&lt;RCUniqueId&gt;&gt;_ICDTSolicitedResponse.xml</td>
</tr>
<tr>
<td>Icdt\ntfn_ack</td>
<td>ICDT – Pickup Notifications/Acknowledgment</td>
<td>B_3CharRandomId</td>
<td>N/A</td>
<td>T#EFT.ON.&lt;ReceiverRoutingId&gt;&gt;.ICDT.B&lt;&lt;3CharRandom&gt;&gt;.&lt;&lt;SenderRoutingID&gt;&gt;.DMMddy.THHmmssS</td>
</tr>
<tr>
<td>Icdt\error</td>
<td>ICDT - Admin Error Response</td>
<td>C_3CharRandomId</td>
<td>N/A</td>
<td>T#EFT.ON.&lt;ReceiverRoutingId&gt;&gt;.ADM.C&lt;&lt;3CharRandom&gt;&gt;.&lt;&lt;SenderRoutingID&gt;&gt;.DMMddy.THHmmssS</td>
</tr>
<tr>
<td>Data Directory</td>
<td>Program Type</td>
<td>Folder Names</td>
<td>XML File Name</td>
<td>Notes</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------</td>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Icdt\error</td>
<td>ICDT Solicited Request - Validation</td>
<td>V_PackageUniqueID</td>
<td>V_Packageuniqueid_Validation_Error.xml</td>
<td>V_Packageuniqueid_Validation_Error.xml</td>
</tr>
<tr>
<td></td>
<td>errors from esMD</td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ICDT Solicited Response - Validation</td>
<td>V_PackageUniqueID</td>
<td>V_Packageuniqueid_Validation_Error.xml</td>
<td>V_Packageuniqueid_Validation_Error.xml</td>
</tr>
<tr>
<td></td>
<td>errors from esMD</td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ICDT Unsolicited Response - Validation</td>
<td>V_PackageUniqueID</td>
<td>V_Packageuniqueid_Validation_Error.xml</td>
<td>V_Packageuniqueid_Validation_Error.xml</td>
</tr>
<tr>
<td></td>
<td>errors from esMD</td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eMDRRегистration</td>
<td>eMDR Registration Batch File from esMD</td>
<td>E_3CharRandomID</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>Pickup Notification for eMDR</td>
<td>P_L5_3CharRandomID</td>
<td>P_L5_3CharRandomID_Pickup.xml</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Registered Provider Batch File</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>Pickup Notification for eMDR</td>
<td>P_L5_3CharRandomID</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Registered Provider Batch File</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>PrePay eMDR letters (CTC 1.5)</td>
<td>U_UniqueId</td>
<td>U_UniqueID_eMDR_ProcessMetadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>Acknowledgment</td>
<td>PrePay eMDR letters Acknowledgment</td>
<td>A_L1_5_GUID</td>
<td>A_L1_5_GUID_Receipt_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>PrePay eMDR letters Validation error</td>
<td>F_L1_5_GUID</td>
<td>F_L1_5_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>response</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>PostPay eMDR letters (CTC 1.6)</td>
<td>U_UniqueId</td>
<td>U_UniqueID_eMDR_ProcessMetadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>Acknowledgment</td>
<td>PostPay eMDR letters Acknowledgment</td>
<td>A_L1_6_GUID</td>
<td>A_L1_6_GUID_Receipt_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>PrePay eMDR letters Validation error</td>
<td>F_L1_6_GUID</td>
<td>F_L1_6_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>response</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input</td>
<td>Document Code Batch File from esMD</td>
<td>E_3CharRandomID</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>Data Directory</td>
<td>Program Type</td>
<td>Folder Names</td>
<td>XML File Name</td>
<td>Notes</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------</td>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>N/A</td>
<td>Pickup Notification for document Code Batch File</td>
<td>P_L17_3CharRandomID</td>
<td>&lt;&lt;ReceiverRoutingId&gt;&gt;.T.L17.P&lt;&lt;esMDTransactionID&gt;&gt;.&lt;&lt;SenderRoutingID&gt;&gt;.DMMddyy.THHmmsS</td>
<td>P_L17_3CharRandomID_Pickup.xml</td>
</tr>
<tr>
<td>input</td>
<td>HOPD (CTC 8.5)</td>
<td>E_L8_5_GUID</td>
<td>E_L8_5_GUID_metadata.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>error</td>
<td>HOPD Validation error response</td>
<td>F_L8_5_GUID</td>
<td>F_L8_5_GUID_Validation_Error.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>notification</td>
<td>HOPD HIH delivery notification</td>
<td>N_L8_5_GUID_MMDDYY_HHMM</td>
<td>N_L8_5_GUID_Delivery_Acknowledgement.xml</td>
<td>N/A</td>
</tr>
<tr>
<td>N/A</td>
<td>HOPD Pickup Notification</td>
<td>P_L8_5_GUID</td>
<td>P_L8_5_GUID_Pickup.xml</td>
<td>N/A</td>
</tr>
</tbody>
</table>
27. Contacts

Table 66: Support Points of Contact provides the contact list for the esMD Service Desk.

<table>
<thead>
<tr>
<th>Contact</th>
<th>Phone</th>
<th>Email</th>
<th>Hours of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMS esMD Service Desk</td>
<td>(443) 832-1856</td>
<td><a href="mailto:esMD_Support@cms.hhs.gov">esMD_Support@cms.hhs.gov</a></td>
<td>Regular Business Hours: 8 a.m. to 8 p.m. Eastern Time (ET).</td>
</tr>
</tbody>
</table>
Appendix A: Description of Fields on RC Client Tabs

Table 67: 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>
<thead>
<tr>
<th>Name of Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction ID</td>
<td>The esMD TransactionId format is as follows:</td>
</tr>
<tr>
<td></td>
<td>• The length of TransactionId will be 15 alphanumeric characters</td>
</tr>
<tr>
<td></td>
<td>• TransactionId will consist of alphabetic (a-z, A-Z) and numeric (0-9) characters.</td>
</tr>
<tr>
<td>Reject Error Category</td>
<td>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).</td>
</tr>
<tr>
<td>Reject Error Code</td>
<td>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.</td>
</tr>
<tr>
<td>Decision Indicator</td>
<td>Decision provided for the Error Response should be the following:</td>
</tr>
<tr>
<td></td>
<td>• &quot;R&quot;- Decision Indicator: &quot;R&quot; is provided in the response when the Decision is &quot;Rejected&quot; for the request received.</td>
</tr>
<tr>
<td>Reason Code</td>
<td>5-character reason code is provided. Minimum of 1 and up to maximum of 25 reason codes can be provided. (Required Element).</td>
</tr>
<tr>
<td>Request Level UTN</td>
<td>UTN is provided for each response. (Optional Element). Format of the unique tracking number is 14 Alpha Numeric Characters.</td>
</tr>
</tbody>
</table>

Table 68: 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>
<thead>
<tr>
<th>Name of Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction ID</td>
<td>The esMD TransactionId format is as follows:</td>
</tr>
<tr>
<td></td>
<td>• The length of TransactionId will be 15 alphanumeric characters</td>
</tr>
<tr>
<td></td>
<td>• TransactionId will consist of alphabetic (a-z, A-Z) and numeric (0-9) characters.</td>
</tr>
<tr>
<td>Error Situation</td>
<td>Error code/situation; can be one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Corrupt Files/Cannot read file.</td>
</tr>
<tr>
<td></td>
<td>• Virus found.</td>
</tr>
<tr>
<td></td>
<td>• Submission sent to incorrect RC or Other.</td>
</tr>
<tr>
<td></td>
<td>• Incomplete File.</td>
</tr>
<tr>
<td></td>
<td>• Unsolicited Response.</td>
</tr>
<tr>
<td></td>
<td>• Documentation cannot be matched to a case/claim.</td>
</tr>
<tr>
<td></td>
<td>• Duplicate.</td>
</tr>
</tbody>
</table>
Table 69: Descriptions of Fields on Advanced/Debugging Tab lists the descriptions of the fields on the Advanced/Debugging tab.

Table 69: Descriptions of Fields on Advanced/Debugging Tab

<table>
<thead>
<tr>
<th>Name of Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td>IDM User ID is a Required Element for testing the connectivity to esMD Cloud Environment.</td>
</tr>
<tr>
<td>Password</td>
<td>IDM password is a Required Element for testing the connectivity to esMD Cloud Environment.</td>
</tr>
</tbody>
</table>
Appendix B:  Reject Error Codes

Use the link below to obtain an up-to-date list of Reject Error Codes from the esMD Downloads section:


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

Use the link below to obtain an up-to-date list of Industry Codes from the esMD Downloads section:


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

Table 70: Content Type Code Descriptions provides the description of the Content Type Codes.

Table 70: Content Type Code Descriptions

<table>
<thead>
<tr>
<th>Content Type Code</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Response to ADR</td>
<td>N/A</td>
</tr>
<tr>
<td>1.5</td>
<td>Prepay eMDR letters</td>
<td>Prepay eMDR letters from RC to esMD</td>
</tr>
<tr>
<td>1.6</td>
<td>Post-pay eMDR letters</td>
<td>Post-pay eMDR letters from RC to esMD</td>
</tr>
<tr>
<td>5</td>
<td>Service Registration</td>
<td>Service Registration</td>
</tr>
<tr>
<td>17</td>
<td>Document Codes</td>
<td>Document Codes</td>
</tr>
<tr>
<td>7</td>
<td>PWK Unsolicited documents</td>
<td>PWK Unsolicited documents</td>
</tr>
<tr>
<td>8.1</td>
<td>Non-Emergent Ambulance Transport</td>
<td>N/A</td>
</tr>
<tr>
<td>8.3</td>
<td>HHPCR</td>
<td>N/A</td>
</tr>
<tr>
<td>8.4</td>
<td>DMEPOS</td>
<td>N/A</td>
</tr>
<tr>
<td>8.5</td>
<td>HOPD</td>
<td>N/A</td>
</tr>
<tr>
<td>9</td>
<td>First Level Appeal Requests</td>
<td>N/A</td>
</tr>
<tr>
<td>9.1</td>
<td>Second Level Appeal Requests</td>
<td>N/A</td>
</tr>
<tr>
<td>10</td>
<td>ADMC</td>
<td>N/A</td>
</tr>
<tr>
<td>11</td>
<td>RA Requests</td>
<td>N/A</td>
</tr>
<tr>
<td>11.1</td>
<td>DME Phone Discussion Requests</td>
<td>N/A</td>
</tr>
<tr>
<td>12</td>
<td>Supporting Documentation for the unsolicited X12N 278 Request</td>
<td>N/A</td>
</tr>
<tr>
<td>13</td>
<td>Supporting Documentation for the X12N 278 Request</td>
<td>N/A</td>
</tr>
<tr>
<td>15.1</td>
<td>ICDT Request</td>
<td>Supports requests for documentation from an RC to another RC.</td>
</tr>
<tr>
<td>15.2</td>
<td>ICDT Solicited Response</td>
<td>Supports responses from an RC for previously requested documentation to another RC.</td>
</tr>
<tr>
<td>15.3</td>
<td>ICDT Unsolicited Response</td>
<td>Supports an RC sending misdirected documentation to another RC.</td>
</tr>
</tbody>
</table>

Table 71: Content Type Codes and Business Types provides the description of the Content Type Codes and the Business Type associated with each Content Type Code.

Table 71: Content Type Codes and Business Types

<table>
<thead>
<tr>
<th>Content Type Code</th>
<th>Business Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Response message for additional documentation request</td>
</tr>
<tr>
<td>1.5</td>
<td>Pre-Pay eMDR</td>
</tr>
<tr>
<td>1.6</td>
<td>Post-Pay eMDR</td>
</tr>
<tr>
<td>5</td>
<td>Service Registration</td>
</tr>
<tr>
<td>17</td>
<td>Document Codes</td>
</tr>
<tr>
<td>7</td>
<td>PWK Unsolicited documents</td>
</tr>
<tr>
<td>8.1</td>
<td>Non-Emergent Ambulance Transport</td>
</tr>
<tr>
<td>8.3</td>
<td>HHPCR</td>
</tr>
<tr>
<td>Content Type Code</td>
<td>Business Type</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>8.4</td>
<td>DMEPOS PA</td>
</tr>
<tr>
<td>8.5</td>
<td>HOPD</td>
</tr>
<tr>
<td>9</td>
<td>First Level Appeal</td>
</tr>
<tr>
<td>9.1</td>
<td>Second Level Appeal</td>
</tr>
<tr>
<td>10</td>
<td>ADMC</td>
</tr>
<tr>
<td>11</td>
<td>RA Requests</td>
</tr>
<tr>
<td>11.1</td>
<td>DME Phone Discussion Requests</td>
</tr>
<tr>
<td>13</td>
<td>XDR X12</td>
</tr>
<tr>
<td>15.1</td>
<td>ICDT Request</td>
</tr>
<tr>
<td>15.2</td>
<td>ICDT Solicited Response</td>
</tr>
<tr>
<td>15.3</td>
<td>ICDT Unsolicited Response</td>
</tr>
</tbody>
</table>
**Appendix E: Acronyms**

Table 72: Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Literal Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACN</td>
<td>Attachment Control Number</td>
</tr>
<tr>
<td>ADMC</td>
<td>Advance Determination of Medicare Coverage</td>
</tr>
<tr>
<td>ADR</td>
<td>Additional Documentation Request</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ASC</td>
<td>Accredited Standards Committee</td>
</tr>
<tr>
<td>AUTH</td>
<td>Authentication</td>
</tr>
<tr>
<td>C-CDA</td>
<td>Consolidated Clinical Document Architecture</td>
</tr>
<tr>
<td>CMS</td>
<td>Centers for Medicare &amp; Medicaid Services</td>
</tr>
<tr>
<td>CTC</td>
<td>Content Type Code</td>
</tr>
<tr>
<td>DC</td>
<td>Data Center</td>
</tr>
<tr>
<td>DME</td>
<td>Durable Medical Equipment</td>
</tr>
<tr>
<td>DMEPOS</td>
<td>Durable Medical Equipment, Prosthetics, Orthotics and Supplies</td>
</tr>
<tr>
<td>DTO</td>
<td>Data Transfer Object</td>
</tr>
<tr>
<td>EDI</td>
<td>Electronic Data Interchange</td>
</tr>
<tr>
<td>EFT</td>
<td>Enterprise File Transfer</td>
</tr>
<tr>
<td>IDM</td>
<td>Identity Management.</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>eMDR</td>
<td>Electronic Medical Documentation Request</td>
</tr>
<tr>
<td>esMD</td>
<td>Electronic Submission of Medical Documentation</td>
</tr>
<tr>
<td>FFS</td>
<td>Fee-For-Service</td>
</tr>
<tr>
<td>GUID</td>
<td>esMD Transaction ID</td>
</tr>
<tr>
<td>HHPCR</td>
<td>Home Health Services Pre-Claim Review</td>
</tr>
<tr>
<td>HICN</td>
<td>Health Insurance Claim Number</td>
</tr>
<tr>
<td>HIH</td>
<td>Health Information Handler</td>
</tr>
<tr>
<td>HL7</td>
<td>Health Level Seven International</td>
</tr>
<tr>
<td>HOPD</td>
<td>Hospital Outpatient Department</td>
</tr>
<tr>
<td>ICDT</td>
<td>Inter Contractor Document Transfer</td>
</tr>
<tr>
<td>Kbps</td>
<td>Kilobits Per Second</td>
</tr>
<tr>
<td>LOB</td>
<td>Line of Business</td>
</tr>
<tr>
<td>MAC</td>
<td>Medicare Administrative Contractor</td>
</tr>
<tr>
<td>MB</td>
<td>Megabytes</td>
</tr>
<tr>
<td>MCS</td>
<td>Multi-Carrier System</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extension</td>
</tr>
<tr>
<td>NPI</td>
<td>National Provider Identifier</td>
</tr>
<tr>
<td>OID</td>
<td>Object Identifier or Organizational Identifier</td>
</tr>
<tr>
<td>PA</td>
<td>Prior Authorization</td>
</tr>
<tr>
<td>PCR</td>
<td>Pre-Claim Review</td>
</tr>
<tr>
<td>PDF</td>
<td>Portable Document Format</td>
</tr>
<tr>
<td>PROD</td>
<td>Production</td>
</tr>
<tr>
<td>PWK</td>
<td>Paperwork</td>
</tr>
<tr>
<td>RA</td>
<td>Recovery Auditor</td>
</tr>
<tr>
<td>RC</td>
<td>Review Contractor</td>
</tr>
<tr>
<td>RRL</td>
<td>Review Results Letter</td>
</tr>
<tr>
<td>RSA</td>
<td>Rivest, Shamir &amp; Adleman</td>
</tr>
<tr>
<td>SSH</td>
<td>Secure Shell</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>Acronym</td>
<td>Literal Translation</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>UAT</td>
<td>User Acceptance Test</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>URL</td>
<td>Universal Resource Locator</td>
</tr>
<tr>
<td>UTN</td>
<td>Universal Tracking Number</td>
</tr>
<tr>
<td>VDC</td>
<td>Virtual Data Center</td>
</tr>
<tr>
<td>XDR</td>
<td>Cross-Enterprise Document Reliable Interchange</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>ZPIC</td>
<td>Zone Program Integrity Contractor</td>
</tr>
</tbody>
</table>
## Appendix F: Glossary

### Table 73: Glossary

<table>
<thead>
<tr>
<th>Glossary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional Documentation Request (ADR)</td>
<td>Official letters sent to Providers from CMS RCs requesting additional documentation that is needed to process claims.</td>
</tr>
<tr>
<td>Advanced Determination of Medical Coverage (ADMC)</td>
<td>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.</td>
</tr>
<tr>
<td>CONNECT</td>
<td>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.</td>
</tr>
<tr>
<td>Electronic Submission of Medical Documentation (esMD)</td>
<td>A new mechanism for submitting medical documentation via a secure internet gateway connecting Providers to the Centers for Medicare &amp; 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.</td>
</tr>
<tr>
<td>Health Information Handler (HIH)</td>
<td>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.</td>
</tr>
<tr>
<td>Inter Contractor Document Transfer (ICDT)</td>
<td>A new functionality that allows RCs to exchange files/documents from one RC to another RC through the esMD system.</td>
</tr>
<tr>
<td>Interface</td>
<td>A well-defined boundary where direct contact between two different environments, systems, etc., occurs, and where information is exchanged.</td>
</tr>
<tr>
<td>Security</td>
<td>The physical, technological, and administrative safeguards used to protect individually identifiable health information.</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol (SOAP) is a message exchange format for web services.</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that &quot;provide communications security over the Internet&quot;. TLS and SSL encrypt the segments of network connections above the Transport Layer, using symmetric cryptography for privacy and a keyed message authentication code for message reliability. TLS is an Internet Engineering Task Force (IETF) standards track protocol, last updated in RFC 5246, and based on the earlier SSL specifications developed by Netscape Corporation.</td>
</tr>
<tr>
<td>Transaction</td>
<td>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.</td>
</tr>
</tbody>
</table>
Appendix G: Frequently Asked Questions (FAQs)

- 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, the 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 a 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 and every file is erroring out.
  - RC Client needs folder permissions 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 the 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 AR2023.01.0 Review Contractor (RC) Client Microsoft .NET User Guide and Installation Handbook, Version 10.0, 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: ______________________
Print Name: Ayana Chavis ___________________________
Title: Contracting Officer’s Representative ___________________________
Role: CMS Approving Authority ___________________________