List of the changes from the previous version

1. Following sections were modified for the current version:
   - Save manifest
   - Update manifest
   - Delete manifest
   - Correct manifest
   - Manifest entities and fields validation for Save service
   - Manifest entities and fields validation for Update service
   - Manifest entities and fields validation for Create Correction service

Introduction

Description

This document describes the use of and validations performed by the e-Manifest services. The document focuses on the Save, Update, Delete, Correct and Revert services. Save is the initial creation of a manifest in e-Manifest. The Save service may be used for a fully electronic manifest or to upload a paper manifest and its data. The Update service is used for subsequent changes to a previously submitted manifest and is based on the Manifest's Tracking Number provided. The Delete service is used to delete the manifest. The Correct service is used for creation and updating a manifest version. The Revert service is used to delete a manifest version under correction.

A user implementing these services must also submit an authentication token with each service call. The authentication token is obtained using the Authentication service which is covered in e-Manifest Authenticate Get and Lookup Services v4 05 12 2019 final.pdf document.

In order to use any service, the user must have the site manager role for at least one of the handlers listed on the manifest and have requested an API ID and API Key through the e-Manifest module of the RCRAInfo Industry Application.

Save manifest

Description

The Save Service creates the new Manifest. The service accepts Manifest data in JSON format compliant with e-Manifest JSON schema (e-Manifest JSON Schema). The following submission types are supported for this release:

- "FullElectronic"
- "DataImage5Copy"
- "Image"
- "Hybrid"

For the "FullElectronic" and "Hybrid" submission types, the Manifest can be submitted at either "Pending" (post June release) or "Scheduled" status. The transition to the statuses after "Scheduled" for these submission types will be done automatically by e-Manifest.

For the “DataImage5Copy” and “Image” submission types, the Manifest will be assigned the "ReadyForSignature" status. The transition to the next status for these submission type will be done automatically by e-Manifest.

Manifests with the “Mail” origin type cannot be saved via the Save service.

For the “DataImage5Copy” and “Image” submission types, the service requires receiving the scanned compressed document attachment (Printed/Paper, Signed, Scanned Manifest form-2050). Manifest attachment shall be passed as a multipart binary content. (See sample client implementation for details at: https://github.com/USEPA/e-manifest/tree/master/Services-Information/sample-client)

For the “DataImage5Copy” and “Image” submission types, if the Manifest contains an attachment, the following metadata JSON elements shall be presented in the Manifest JSON:

```
 "printedDocument": {
          "name": "user provided document name.pdf",
          "size": 23455, (size in bytes of the pdf document)
          "mimeType": " APPLICATION_PDF"
 }
```

The service will validate the submitted JSON and if:

- No Errors or Warnings were found: Service will save the manifest and return generated Manifest Tracking Number
- Only Warnings were found: Service will save the manifest and return generated Manifest Tracking Number and Warning Report containing all found Warnings
- Error(s) were found: Service won’t save the manifest and return Error Report containing all found errors.
- Error(s) and Warnings were found. Service won’t save the manifest and return Error and Warning Report containing all found errors and warnings

Service Type

e-Manifest Data Service

Parameters

- Security Token: Security Token received in Authentication. Will be passed in the Header of the request.
- Manifest JSON
- Attachment Document: If applicable, manifest attachment shall be passed as a multipart binary content.

Service Invocation Example

Request URLs:

PreProduction/ Test Environment:

https://rcrainfopreprod.epa.gov/rcrainfo/rest/api/v1/emanifest/manifest/save

Production:

https://rcrainfo.epa.gov/rcrainfoprod/rest/api/v1/emanifest/manifest/save

Request Body:

emanifest JSON

Request JSON Schema

emanfest.json

Manifest JSON examples

Valid Manifest JSON Example for “FullElectronic” submission type

```
{
  "submissionType" : "FullElectronic",
  "status" : "Scheduled",
  "import" : false,
  "rejection" : false,
  "generator" : {
     "epaSiteId" : "MDD981111081",
      "contact" : {
      "phone" : {
        "number" : "301-294-9727"
      }
    },
    "emergencyPhone" : {
      "number" : "301-423-5455"
    }
  },
  "transporters" : [ {
     "epaSiteId" : "CAR000189282",
     "name": "transporter 1",
     "order" : 1
  },
  {
    "epaSiteId" : "CAD982000564",
     "name": "transporter 2",
      "order" : 2
  } ],
  "designatedFacility" : {
     "epaSiteId" : "AK8570028649"
  },
  "wastes" : [ {
    "dotHazardous" : true,
     "epaWaste" : true,
     "pcb" : false,
    "dotInformation" : {
      "idNumber" : {
        "code" : "UN1988"
      },
      "printedDotInformation" : "UN1988, Waste Aldehydes, flammable, toxic, n.o.s., (technical name), 3,
I, RQ(rq description)"
    },
    "quantity" : {
      "containerNumber" : 100,
      "containerType" : {
        "code" : "BA"
      },
      "quantity" : 2,
      "unitOfMeasurement" : {
        "code" : "L"
      }
    },
    "hazardousWaste" : {
      "federalWasteCodes" : [ {
        "code" : "D023"
      }, {
        "code" : "D001"
      }, {
        "code" : "D021"
      } ],
      "tsdfStateWasteCodes" : [ ],
      "generatorStateWasteCodes" : [ ]
    },
    "additionalInfo" : {
      "comments" : [ ]
    },
    "lineNumber" : 1,
    "br" : true,
    "brInfo" : {
      "densityUnitOfMeasurement" : {
        "code" : "1"
      },
      "formCode" : {
        "code" : "W107"
      },
      "sourceCode" : {
        "code" : "G07"
      },
     "wasteMinimizationCode": {
        "code": "X"
      },
      "density" : 1.0
    },
    "managementMethod" : {
      "code" : "H020"
    }
  } ],
  "additionalInfo" : {
    "comments" : [ ]
  }

}
```

Sequence of steps

1. The System will validate the Security Token

1.1. If Web Security Token is invalid, the system stops the submission and generates the following error:

- E_SecurityApiTokenInvalid: Invalid Security Token

1.2. If Web Security Token expired, the system stops the submission and generates the following error:

- E_SecurityApiTokenExpired: Security Token is Expired

1.3. If Account was inactivated after the token was issued, the system stops the submission and generates the following error:

- E_SecurityApiInvalidStatus: This API Id is no longer active

2. The system will perform User Authorization

2.1. The system will check if the User has Site Manager (Industry) permissions for any Site.

2.2. If the User does not have permissions, the system will stop the processing and generates the following error:

- E_IndustryPermissions: The user does not have industry permissions for any Site

2.3. The system will check if the user has Site Manager Permissions for either the TSDF Site ID or Generator Site ID or Alternate TSDF site ID provided in the manifest.

2.3.1. If the user has Site Manager Permissions for the TSDF Site, then TSDF acts as "TSDF"

2.3.2. If the user has Site Manager Permissions for the Generator Site, then TSDF acts as "Generator".

2.3.3. If the user has Site Manager Permissions for the Alternate TSDF Site, then TSDF acts as "Alternate TSDF".

2.4. If the User does not have Site Manager Permissions for either the TSDF Site or for the Generator Site or Alternate TSDF Site, then the System will stop the processing and generates the following error:

- E_SitePermissions: The user does not have Industry Permissions for either the TSDF Site or Generator Site or Alternate TSDF Site specified in provided JSON

3. If the User was Authenticated and Authorized, the system will process the request

4. The system will validate the provided Manifest JSON and Attachment Document. The service will validate the e-Manifest according to the rules described in section "Manifest entities and fields validation for Save service". If no errors or warnings were generated during the validation process, the service will perform the following steps:

4.1.1. Generate Manifest Tracking Number for the provided Manifest data

4.1.2. Store provided Manifest information and Attachment Document (if applicable) into the e-Manifest database

4.1.3. Return Manifest Tracking Number to the Requestor

4.2. If a warning is generated during the validation process, the service will perform the following steps:

4.2.1. Generate Manifest Tracking Number for the provided Manifest data

4.2.2. Store the provided Manifest information and an Attachment Document into the e-Manifest database

4.2.3. Return the Manifest Tracking Number and Warning(s) Report to the Requestor

4.3. If an error is generated during the validation process, the service will perform following steps:

4.3.1. Generate Error Report with all errors found during validation process

4.3.2. Return Error Report to the Requestor

4.4. If any errors and warnings were generated during validation process, the service will perform the following steps:

4.4.1. Generate Error/Warning Report with all errors and warnings found during validation process

4.4.2. Return Error/Warning Report to the Requestor

Response JSON Schemas

crud-emanifest-return.json Schema for successful and failed submission response

Completed Response Examples

Response Without Warnings Example

```
{
  "manifestTrackingNumber": "100005103ELC"
  "reportId": "378a9fa2-db69-4a0a-a772-45ecd1ae67c8",
  "date": "2018-10-12T21:00:16.604+0000",
  "operationStatus": "Saved"
}
```

Response With Warnings Example

```
{
  "manifestTrackingNumber": "100005103ELC"
  "reportId": "378a9fa2-db69-4a0a-a772-45ecd1ae67c8",
  "date": "2018-10-12T21:00:16.604+0000",
  "operationStatus": "Saved",
   "warnings": [
     {
       "message": "Invalid Format. Text not exceeding 255 characters is expected",
       "field": "Emanifest.additionalInfo.comments.label",
       "value": "too many characters"
     }
   ],
   "generatorReport": {
     "entityId": {
       "entityIdField": "siteId",
       "entityIdValue": "MDR12345125"
     },
     "warnings": [
     {
       "message": "Invalid Format. Numbers not exceeding 6 characters is expected",
       "field": "Emanifest.generator.emergencyPhone.extension",
       "value": "N789"
     }
     ]
   },
      "tsdfReport": {
        "entityId": {
          "entityIdField": "siteId",
          "entityIdValue": "MDR12345125"
        },
        "warnings": [
        {
          "message": "Invalid Format. Alphanumeric characters not exceeding 50 characters is
          expected",
          "field": "Emanifest.designatedFacility.siteAddress.address2",
          "value": ";^"
        }
        ]
      },
      "wastesReports": [
      {
        "entityId": {
        "entityIdField": "lineNumber",
        "entityIdValue": "1"
      },
        "warnings": [
        {
        "message": "Invalid Format. Numeric characters is expected",
        "field": "Emanifest.wastes.additionalInfo.handlerId",
        "value": "FG056"
        }
        ]
        },
      {
        "entityId": {
        "entityIdField": "lineNumber",
        "entityIdValue": "2"
      },
      "warnings": [
      {
        "message": "Invalid Format. Text not exceeding 4000 is expected",
        "field": "Emanifest.wastes.additionalInfo.handlingInstructions",
        "value": "Way too long..."
      }
      ]
      }
      ]
      }
    }
```

Failed Response Example

```
{
    "reportId": "012345",
    "date": "2017-07-27T19:13:28.628+0000",
     "operationStatus": "Failed",
    "errors": [
    {
            "message": "Invalid Field Format. FullElectronic or DataImage5Copy is expected",
            "field": "Emanifest.submissionType",
            "value": "DataImage"
    },
    {
        "message": "Mandatory Field is not provided",
        "field": "Emanifest.designatedFacility"
    },
    {
        "message": "No Transporters specified. At least one Transporter is required for
Scheduled Status",
        "field": "Emanifest.transporter"
    }
    ],
    "manifestWarnings": [
        {
            "message": "Invalid Field Format. Text value not exceeding 255 characters is
expected",
            "field": "Emanifest.additionalInfo.comments.label",
            "value": "too many characters"
        }
    ],
    "generatorReport": {
        "entityId": {
            "entityIdField": "siteId",
            "entityIdValue": "MDD003473337"
        },
        "errors": [
            {
                "message": "Mandatory Field is not provided",
                "field": "Emanifest.generator.name"
            },
            {
                "message": "Lookup Value Not Found",
                "field": "Emanifest.generator.siteAddress.state.code",
                "value": "NA"
            },
            {
                "message": "Mandatory Field is not provided",
                "field": "Emanifest.generator.mailingAddress"
            }
        ],
        "warnings": [
            {
                "message": "Invalid Format",
                "field": "Emanifest.generator.emergencyPhone.extension",
                "value": "N789"
            }
        ]
    },
    "tsdfReport": {
        "entityId": {
            "entityIdField": "siteId",
            "entityIdValue": "N/A"
        },
        "errors": [
            {
                "message": "Mandatory Field is not provided",
                "field": "Emanifest.designatedFacility.siteId"
            },
            {
                "message": "Invalid Format",
                "field": "Emanifest.designatedFacility.siteAddress.zip",
                "value": "X345"
            },
            {
                "message": "Mandatory Field is not provided",
                "field": "Emanifest.designatedFacility.emergencyPhone.number"
            }
        ],
        "warnings": [
            {
                "message": "Invalid Format",
                "field": "Emanifest.designatedFacility.siteAddress.address2",
                "value": ";^"
            }
        ]
    },
    "wastesReports": [
        {
            "entityId": {
                "entityIdField": "lineNumber",
                "entityIdValue": "1"
            },
            "errors": [
                {
                    "message": "Lookup Value Not Found",
                    "field": "Emanifest.wastes.dotInformation.idNumber",
                    "value": "UN4345"
                },
                {
                    "message": "Mandatory Field is not provided",
                    "field": "Emanifest.wastes.quantity.quantity"
                },
                {
                "message": "Invalid Field Format.Integer number not exceeding 999 is
expected",
                    "field": "Emanifest.wastes.quantity.containerNumber",
                    "value": "n"
                }
            ],
            "warnings": [
                {
                    "message": "Invalid Field Format.",
                    "field": "Emanifest.wastes.additionalInfo.handlerId",
                    "value": "FG056"
                }
            ]
        },
        {
            "entityId": {
                "entityIdField": "lineNumber",
                "entityIdValue": "2"
            },
            "errors": [
                        {
                    "message": "Waste Description is not provided. Waste Description must be provided if hazardous is false",
                            "field": "Emanifest.wastes.wasteDescription"
                        },
                        {
                    "message": "Density is not provided. Density must be provided if quantity UOM is Gallons, Liters or Cubic meters",
                            "field": "Emanifest.wastes.br.density"
                        },
                        {
                            "message": "Lookup Value Not Found",
                            "field": "Emanifest.wastes.managementMethod.code",
                            "value": "X25"
                        }
                    ]
                }
            ]
  }
```

Errors

Type	code	Description
N/A	E_SecurityApiTokenInvalid	Invalid Security Token
N/A	E_SecurityApiTokenExpired	Security Token is Expired
N/A	E_SecurityApiInvalidStatus	This API Id is no longer active
Authorization	E_IndustryPermissions	The user does not have Industry permissions for any Site
Validation	E_SitePermissions	The user does not have Industry Permissions for either the TSDF Site or Generator Site or Alternate TSDF site specified in provided JSON
System	E_SystemError	Message describing system error

Update manifest

Description

The Update service updates an existing Manifest. The service requires that the industry system submits a complete Manifest JSON in order to update a manifest. Individual Manifest fields cannot be updated via this service, only the entire manifest. The update service accepts Manifest data in JSON format compliant with the e-Manifest JSON schema (e-Manifest JSON Schema) and can also accept a Manifest attachment for the "DataImage5Copy" and "Image" submission types (Scanned Signed Paper Manifest Image pdf). Manifest JSON must contain the Manifest Tracking Number generated for that manifest by the Save Manifest service. The Save manifest service must be invoked prior to the Update Manifest service for any manifest. The following submission types are supported:

- "FullElectronic"
- "DataImage5Copy"
- "Image"
- "Hybrid"

For “FullElectronic” and “Hybrid” submission types, the Manifest can be updated with “Pending” or “Scheduled” statuses. For these submission types the transition to the next statuses after “Scheduled” will be done by e-Manifest.
For “DataImage5Copy” and “Image” submission types the Manifest status shall not be submitted. For this submission type the status transition will be done by e-Manifest.
Signed manifests of any submission types cannot be Updated via Update service. “Image” Manifests submitted via mail cannot be updated via Update service at any time.For the “DataImage5Copy” and “Image” submission types the service supports receiving the scanned document attachment (Printed, Signed, Scanned Manifest form-2050). Manifest attachment shall be passed as a multipart binary content. (See sample client implementation for details). For the “DataImage5Copy” submission type if the Manifest contains an attachment the following metadata JSON elements shall be presented in the Manifest JSON:

```
 "printedDocument": {
          "name": "user provided document name.pdf",
          "size": 23455, (size in bytes of the pdf document)
          "mimeType": " APPLICATION_PDF"
 }
```

Service will validate submitted JSON (see section Manifest entities and fields validation for Update service) and if:

- No Errors or Warnings were found: The service will update the manifest and return the existing Manifest Tracking Number
- Only Warnings were found: The service will update the manifest and return the existing Manifest Tracking Number and Warning Report containing all found Warnings
- Error(s) were found: The service won’t update the manifest and returns Error Report containing all found errors.
- Error(s) and Warnings were found. The service won’t update the manifest and returns Error Report containing all found errors and warnings.

Service Type

e-Manifest Data Service

Parameters

- Security Token: Security Token received in Authentication. Will be passed in the Header of the request.
- Manifest JSON
- Attachment Document If applicable, manifest attachment shall be passed as a multipart binary content.

Service Invocation Example

Request URLs:

PreProduction/ Test Environment:

https://rcrainfopreprod.epa.gov/rcrainfo/rest/api/v1/emanifest/manifest/update

Production:

https://rcrainfo.epa.gov/rcrainfoprod/rest/api/v1/emanifest/manifest/update

Request Body:

emanifest JSON

Request JSON Schema

emanfest.json

Manifest JSON example

Valid Manifest JSON Example for ”FullElectronic” submission type

```
{
 "manifestTrackingNumber": "100000781ELC",
  "submissionType" : "FullElectronic",
  "status" : "Scheduled",
  "import" : false,
  "rejection" : false,
  "containsPreviousRejectOrResidue":false,
  "generator" : {
     "epaSiteId" : "MDD981111081",
      "contact" : {
      "phone" : {
        "number" : "301-294-9727"
      }
    },
    "emergencyPhone" : {
      "number" : "301-423-5455"
    }
  },
  "transporters" : [ {
     "epaSiteId" : "CAR000189282",
     "name": "transporter 1",
     "order" : 1
  },
  {
    "epaSiteId" : "CAD982000564",
     "name": "transporter 2",
      "order" : 2
  } ],
  "designatedFacility" : {
     "epaSiteId" : "AK8570028649"
  },
  "wastes" : [ {
    "dotHazardous" : true,
     "epaWaste" : true,
     "pcb" : false,
    "dotInformation" : {
        "idNumber" : {
          "code" : "UN1988"
        },
        "printedDotInformation" : "UN1988, Waste Aldehydes, flammable, toxic, n.o.s., (technical name), 3, I, RQ(rq description)"
      },
      "quantity" : {
        "containerNumber" : 100,
        "containerType" : {
          "code" : "BA"
        },
        "quantity" : 2,
        "unitOfMeasurement" : {
          "code" : "L"
        }
      },
      "hazardousWaste" : {
        "federalWasteCodes" : [ {
          "code" : "D023"
        }, {
          "code" : "D001"
        }, {
          "code" : "D021"
        } ],
        "tsdfStateWasteCodes" : [ ],
        "generatorStateWasteCodes" : [ ]
      },
      "additionalInfo" : {
        "comments" : [ ]
      },
      "lineNumber" : 1,
      "br" : true,
      "brInfo" : {
        "densityUnitOfMeasurement" : {
          "code" : "1"
        },
        "formCode" : {
          "code" : "W107"
        },
        "sourceCode" : {
          "code" : "G07"
        },
       "wasteMinimizationCode": {
          "code": "X"
        },
        "density" : 1.0
      },
      "managementMethod" : {
        "code" : "H020"
      }
    } ],
    "additionalInfo" : {
      "comments" : [ ]
    }
```

}

Sequence of steps

1. The System will validate the Security Token
    - 1.1. If the Web Security Token is invalid, the system stops the submission and generates the following error:
    - E_SecurityApiTokenInvalid: Invalid Security Token
    - 1.2. If the Web Security Token expired, the system stops the submission and generates the following error:
    - E_SecurityApiTokenExpired: Security Token is Expired
    - 1.3. If the Account was inactivated after the token was issued, the system stops the submission and generates the following error:
    - E_SecurityApiInvalidStatus: This API Id is no longer active
2. The system will perform User Authorization
    - 2.1. The system will check if the User has Site Manager (Industry) permissions for any Site
    - 2.2. If the User does not have permissions, the system will stop processing the manifest and generate the following error:
    - E_IndustryPermissions: The user does not have industry permissions for any Site
    - 2.3. The system will check if the user has the Industry Permissions for either the TSDF Site ID or Generator Site ID or Alternate TSDF Site or Transporter site(s) provided on the manifest.
    - 2.4. If the User does not have the appropriate permissions the system will stop the processing the manifest and generates the following error:
    - E_SitePermissions: The user does not have Industry Permissions for either the TSDF Site or Generator Site or Alternate TSDF Site or Transporter site(s) specified in provided JSON
3. If the User was Authenticated and Authorized, the system will process the request
    - 3.1. The system will validate the provided Manifest JSON and Attachment Document
        - 3.1.1. The system will validate the e-Manifest fields according to the rules described in section Manifest entities and fields validation for Update service
    - 3.2. If no errors or warnings are generated during the validation process, the service will perform the following steps:
        - 3.2.1. Update existing Manifest with provided Manifest information
        - 3.2.2. Create or Update Attachment Document into e-Manifest database
        - 3.2.3. Return Manifest Tracking Number to the Requestor.
    - 3.3. If only warnings are generated during the validation process, the service will perform following steps:
        - 3.3.1. Update existing Manifest with provided Manifest information
        - 3.3.2. Create or Update Attachment Document into e-Manifest database
        - 3.3.3. Return Manifest Tracking Number and Warning(s) Report to the Requestor
    - 3.4. If an error is generated during the validation process the system will perform the following steps:
        - 3.4.1. Generate Error Report with all errors found during validation process
        - 3.4.2. Return Error Report to the Requestor.

3.5. If any errors and warnings were generated during validation process the system will perform following steps

3.5.1. Generate Error/Warning Report with all errors and warnings found during validation process

3.5.2. Return Error/Warning Report to the Requestor

Response JSON Schema

crud-emanifest-return.json Schema for successful and failed submission response

Completed Response Examples

Response Without Warnings Example

```
{
  "manifestTrackingNumber": "100005103ELC"
  "reportId": "378a9fa2-db69-4a0a-a772-45ecd1ae67c8",
  "date": "2018-10-12T21:00:16.604+0000",
  "operationStatus": "Updated"
}
```

Response With Warnings Example

```
{
"manifestTrackingNumber": "100005103ELC"
  "reportId": "378a9fa2-db69-4a0a-a772-45ecd1ae67c8",
  "date": "2018-10-12T21:00:16.604+0000",
   "operationStatus": "Updated",
   "warnings": [
     {
       "message": "Invalid Format. Text not exceeding 255 characters is expected",
       "field": "Emanifest.additionalInfo.comments.label",
       "value": "too many characters"
     }
   ],
   "generatorReport": {
     "entityId": {
       "entityIdField": "siteId",
       "entityIdValue": "MDR12345125"
     },
     "warnings": [
     {
       "message": "Invalid Format. Numbers not exceeding 6 characters is expected",
       "field": "Emanifest.generator.emergencyPhone.extension",
       "value": "N789"
     }
     ]
   },
   "tsdfReport": {
     "entityId": {
       "entityIdField": "siteId",
       "entityIdValue": "MDR12345125"
```

```
        },
        "warnings": [
        {
          "message": "Invalid Format. Alphanumeric characters not exceeding 50 characters is
          expected",
          "field": "Emanifest.designatedFacility.siteAddress.address2",
          "value": "address value"
        }
        ]
      },
      "wastesReports": [
      {
        "entityId": {
        "entityIdField": "lineNumber",
        "entityIdValue": "1"
      },
        "warnings": [
        {
        "message": "Invalid Format. Numeric characters is expected",
        "field": "Emanifest.wastes.additionalInfo.handlerId",
        "value": "FG056"
        }
        ]
        },
      {
        "entityId": {
        "entityIdField": "lineNumber",
        "entityIdValue": "2"
      },
      "warnings": [
      {
        "message": "Invalid Format. Text not exceeding 4000 is expected",
        "field": "Emanifest.wastes.additionalInfo.handlingInstructions",
        "value": "Way too long..."
      }
      ]
      }
      ]
      }
    }
```

Failed Response Example

```
{
    "manifestTrackingNumber": "100005103ELC"
     "reportId": "378a9fa2-db69-4a0a-a772-45ecd1ae67c8",
     "date": "2018-10-12T21:00:16.604+0000",
      "operationStatus": "Failed",
    "errors": [
    {
        "message": "Invalid Field Format. FullElectronic or DataImage5Copy is expected",
        "field": "Emanifest.submissionType",
        "value": "DataImage"
    },
    {
        "message": "Mandatory Field is not provided",
        "field": "Emanifest.designatedFacility"
    },
    {
        "message": "No Transporters specified. At least one Transporter is required for
Scheduled Status",
        "field": "Emanifest.transporter"
    }
    ],
    "manifestWarnings": [
        {
            "message": "Invalid Field Format. Text value not exceeding 255 characters is
expected",
            "field": "Emanifest.additionalInfo.comments.label",
            "value": "too many characters"
        }
    ],
    "generatorReport": {
        "entityId": {
            "entityIdField": "siteId",
            "entityIdValue": "MDD003473337"
        },
        "errors": [
            {
                "message": "Mandatory Field is not provided",
                "field": "Emanifest.generator.name"
            },
            {
                "message": "Lookup Value Not Found",
                "field": "Emanifest.generator.siteAddress.state.code",
                "value": "NA"
            },
            {
                "message": "Mandatory Field is not provided",
                "field": "Emanifest.generator.mailingAddress"
            }
        ],
        "warnings": [
            {
                "message": "Invalid Format",
                "field": "Emanifest.generator.emergencyPhone.extension",
                "value": "N789"
            }
        ]
    },
    "tsdfReport": {
        "entityId": {
            "entityIdField": "siteId",
            "entityIdValue": "N/A"
        },
        "errors": [
            {
                "message": "Mandatory Field is not provided",
                "field": "Emanifest.designatedFacility.siteId"
            },
            {
                "message": "Invalid Format",
                "field": "Emanifest.designatedFacility.siteAddress.zip",
                "value": "X345"
            },
            {
                "message": "Mandatory Field is not provided",
                "field": "Emanifest.designatedFacility.emergencyPhone.number"
            }
        ],
        "warnings": [
            {
                "message": "Invalid Format",
                "field": "Emanifest.designatedFacility.siteAddress.address2",
                "value": ";^"
            }
        ]
    },
    "wastesReports": [
        {
            "entityId": {
                "entityIdField": "lineNumber",
                "entityIdValue": "1"
            },
            "errors": [
                {
                    "message": "Lookup Value Not Found",
                    "field": "Emanifest.wastes.dotInformation.idNumber",
                    "value": "UN4345"
                },
                {
                    "message": "Mandatory Field is not provided",
                    "field": "Emanifest.wastes.quantity.quantity"
                },
                {
                    "message": "Invalid Field Format.Integer number not exceeding
9999 is expected",
                    "field": "Emanifest.wastes.quantity.containerNumber",
                    "value": "n"
                }
            ],
            "warnings": [
                {
                    "message": "Invalid Field Format.",
                    "field": "Emanifest.wastes.additionalInfo.handlerId",
                    "value": "FG056"
                }
            ]
        },
        {
            "entityId": {
                "entityIdField": "lineNumber",
                "entityIdValue": "2"
            },
            "errors": [
                {
            "message": "Waste Description is not provided. Waste Description must be
provided if hazardous is false",
                    "field": "Emanifest.wastes.wasteDescription"
                },
                {
            "message": "Density is not provided. Density must be provided if quantity UOM
is Gallons, Liters or Cubic meters",
                    "field": "Emanifest.wastes.br.density"
                },
                {
                    "message": "Lookup Value Not Found",
                    "field": "Emanifest.wastes.managementMethod.code",
                    "value": "X25"
                }
            ]
        }
    ]
  }
```

Errors

Type	code	Description
N/A	E_SecurityApiTokenInvalid	Invalid Security Token
N/A	E_SecurityApiTokenExpired	Security Token is Expired
N/A	E_SecurityApiInvalidStatus	This API Id is no longer active
Authorization	E_IndustryPermissions	The user does not have Industry permissions for any Site
Validation	E_SitePermissions	The user does not have Industry Permissions

		for either the TSDF Site or Generator Site or Alternate TSDF site or Transporter site(s) specified in provided JSON
System	E_SystemError	Message describing system error

Delete manifest

Description

The Delete service deletes the manifest by provided Manifest Tracking Number. Depending on the manifest submission type, the manifest can be deleted in the following statuses:

- “FullElectronic “: can be deleted only if in the “Scheduled” status
- “Hybrid”: can be deleted only if in the “Scheduled” status
- “DataImage5Copy”: can be deleted only if in the “ReadyForSignature” status
- “Image”: can be deleted only if originType = “Web” or “Service” and status is in the “ReadyForSignature” status
- “NotSelected”: can be deleted in any possible status

Service Type

Emanifest Data Service.

Parameters

- Security Token: Security Token received in Authentication. Will be passed in the Header of the request.
- Manifest Tracking Number

Service Invocation Example

PreProduction/Test Environment:
https://rcrainfopreprod.epa.gov/rcrainfo/rest/api/v1/emanifest/manifest/delete/100001380ELC
Production:
https://rcrainfo.epa.gov/rcrainfoprod/rest/api/v1/emanifest/manifest/delete/100001380ELC

Sequence of steps

1. The System will validate the Security Token
   1.1. If the Web Security Token is invalid the system stops the submission and generates the following error:
   - E_SecurityApiTokenInvalid: Invalid Security Token

   1.2. If the Web Security Token expired the system stops the submission and generates the following error:
   - E_SecurityApiTokenExpired: Security Token is Expired

   1.3. If the Account was Inactivated after the token was issued the system stops the submission and generates the following error:
   - E_SecurityApiInvalidStatus: This API Id is no longer active
2. The system will perform User Authorization.

2.1. System will check if the User has Site Manager (Industry) permissions for any Site.

2.2. If the User does not have permission the System will stop the processing and generates the following error:

- E_IndustryPermissions: The user does not have industry permissions for any Site

3. The system will process the request

3.1. The system will check if the provided Manifest Tracking Number is valid and in the system.

3.1.1. Manifest Tracking Number shall be compliant with following rules:

3.1.1.1. Nine numeric characters + valid three character Manifest Tracking Number Suffix

3.1.1.2. If the provided Manifest Tracking Number does not have a valid format the processing will be stopped and system generates the following error:

- E_InvalidManifestTrackingNumber: Provided Manifest Tracking Number has invalid format

3.1.1.3. If the provided Manifest Tracking Number does not have a valid suffix the processing will be stopped and system generates the following error:

- E_InvalidManifestTrackingNumberSuffix: Provided Manifest Tracking Number has invalid suffix

3.1.2. If Manifest Tracking Number is not in the system the processing will stop and system generates the following error:

- E_ ManifestTrackingNumberNotFound: Provided Manifest Tracking Number was not found

3.2. The system will check if the user has Industry Permissions for the Designated Facility, Generator, Alternate TSDF or any of the Transporter Site Id(s) provided on the manifest. If the User does not have permission the system will stop the processing and generates the following error:

- E_SitePermissions: The user does not have Industry Permissions for either the TSDF Site or Generator Site or Alternate TSDF site or Transporter site(s) specified in provided JSON

3.3. The system will check if the manifest can be deleted

3.3.1. If Manifest submission type is “FullElectronic” or “Hybrid” or not selected and Manifest status is not “Pending” or “Scheduled” the system will stop the processing and generates the following error:

- E_IncorrectStatusForDelete: Manifest with submission type of [Full Electronic/Hybrid] can be deleted only in Pending or Scheduled status

3.3.2. If Manifest submission type is “DataImage5Copy” and Manifest status is not “ReadyForSignature” the system will stop the processing and generates the following error:

- E_IncorrectStatusForDelete: Manifest with submission type of DataImage5Copy can be deleted only in ReadyForSignature status

3.3.3. If Manifest submission type is “Image” following applies

3.3.3.1. If manifest origin type is “Web” or “Service” and Manifest status is not “ReadyForSignature” the system will stop the processing and generates the following error:

- E_IncorrectStatusForDelete: Manifest with submission type of Image can be deleted only in ReadyForSignature status

3.3.3.2. If manifest origin type is “Mail” the system will stop the processing and generates the following error:

- E_IncorrectOriginForDelete: Manifest with origin type of Mail cannot be deleted

3.4. The System will delete the Manifest

4. On success the system returns JSON containing
   - Manifest Tracking Number
   - Operation Status: “Deleted”
   - Date: Date/Time of the deletion
5. If any system errors were encountered during processing, the system will return:
   - error: containing error code, error message, error id and error date

Response JSON Schema

crud-emanifest-return.json Schema for successful and failed submission response

Completed Response Example

```
{
 "manifestTrackingNumber": "100001380ELC",
 "operationStatus": "Deleted",
  "date": "2017-07-27T19:13:28.628+0000"
}
```

Errors

Type	code	Description
N/A	E_SecurityApiTokenInvalid	Invalid Security Token
N/A	E_SecurityApiTokenExpired	Security Token is Expired
N/A	E_SecurityApiInvalidStatus	This API Id is no longer active
Authorization	E_IndustryPermissions	The user does not have Industry permissions for any Site
Validation	E_InvalidManifestTrackingNumber	Provided Manifest Tracking Number has invalid format
Validation	E_InvalidManifestTrackingNumberSuffix	Provided Manifest Tracking Number has invalid suffix
Validation	E_ManifestTrackingNumberNotFound	Provided Manifest Tracking Number was not found
Validation	E_IncorrectStatusForDelete	Manifest with submission type of [FullElectronic/ Hybrid] can be deleted only in Pending or Scheduled status
Validation	E_IncorrectStatusForDelete	Manifest with submission type of DataImage5Copy can be deleted only in ReadyForSignature status
Validation	E_IncorrectStatusForDelete	Manifest with submission type of Image can be deleted only in ReadyForSignature status
Validation	E_IncorrectOriginForDelete	Manifest with origin type of Mail

		cannot be deleted
System	E_SystemError	Message describing system error

Validation Error Return Example

```
{

      "code": "E_ManifestTrackingNumberNotFound ",
      "message": "Provided Manifest Tracking Number was not found"
      "errorId": "9v7a9808-s35a5-4874-928f-12fc3def4b56",
      "errorDate ": "2017-06-23T23:15:45.095+0000"
}
```

Correct manifest

Description

The Correct Manifest Service creates and updates a Manifest Correction (a new version of the manifest will be either created or updated by the service). If the current manifest status is either “Signed” or “Corrected” then a new version of the Manifest Correction will be created and automatically receives the status of “UnderCorrection”. If the current manifest status is “UnderCorrection” then the current Manifest Correction version can be updated with the same service. Note, there is only one version with the “UnderCorrection” status at any given time and this version is viewable and editable by all registered handlers on the manifest.

For the Image manifests submitted via the Industry application or save service, the Correct service will create a new version either before the manifest was processed by the by Paper Processing Center (PPC) or after the manifest was processed by PPC. If the manifest is currently being processed by PPC the service will return an error.

For manifests submitted by mail (Image) the Correct service will create a new version only after the manifest was processed by PPC. The service will not create new version if the manifest was not processed by PPC or is currently being processed by PPC. The service will return a specific error in both cases.

The service accepts Manifest data in JSON format compliant with e-Manifest JSON schema (e-Manifest JSON Schema). All submission types are supported by this service.For the “DataImage5Copy” and “Image” submission types, the service supports (but does not mandate) receiving the scanned compressed document attachment (Printed/Paper, Signed, Scanned Manifest form-2050). If a user wishes to send a manifest attachment, it shall be passed as a multipart binary content. (See sample client implementation for details at: https://github.com/USEPA/e-manifest/tree/master/Services-Information/sample-client)

For the “DataImage5Copy” and “Image” submission types, if the Manifest contains an attachment the following metadata JSON elements shall be presented in the Manifest JSON:

```
 "printedDocument": {
          "name": "user provided document name.pdf",
          "size": 23455, (size in bytes of the pdf document)
          "mimeType": " APPLICATION_PDF"
 }
```

The service will validate the submitted JSON (see details in section “Manifest entities and fields validation for Create Correction service”) and if:

- No Errors or Warnings were found: The service will create or update the manifest correction and return the Manifest Tracking Number, operation status, and operation date.
- Only Warnings were found: The service will create or update the manifest correction and return the Manifest Tracking Number, operation status, current version, operation date and Warning Report containing all found warnings.
- Error(s) were found: The service will not create or update the manifest correction and return Error Report containing all found errors.
- Error(s) and Warnings were found. Service will not create or update the manifest correction and return Error and Warning Reports containing all found errors and warnings.

Service Type

e-Manifest Data Service

Parameters

- Security Token: Security Token received in Authentication. Will be passed in the Header of the request.
- Manifest JSON passed as request body
- Attachment Document: If applicable, manifest attachment shall be passed as a multipart binary content.

Service Invocation Example

Request URLs:

PreProduction/Test Environment:
https://rcrainfopreprod.epa.gov/rcrainfo/rest/api/v1/emanifest/manifest/correct
Production:
https://rcrainfo.epa.gov/rcrainfoprod/rest/api/v1/emanifest/manifest /correct

Request Body:

emanifest JSON

Request JSON Schema

emanfest.json

Sequence of steps

1. The System will validate the Security Token

1.1. If the Web Security Token is invalid, the system stops the submission and generates the following error:

- E_SecurityApiTokenInvalid: Invalid Security Token

1.2. If the Web Security Token has expired, the system stops the submission and generates the following error:

- E_SecurityApiTokenExpired: Security Token is Expired

1.3. If the Account was inactivated after the token was issued, the system stops the submission and generates the following error:

- E_SecurityApiInvalidStatus: This API Id is no longer active

2. The system will perform User Authorization
    - 2.1. The system will check if the User has Site Manager (Industry) permissions for any Site
    - 2.2. If the User does not have permissions, the system will stop processing the manifest and generate the following error:
    - E_IndustryPermissions: The user does not have industry permissions for any Site
    - 2.3. The system will check if the user has the Industry Permissions for either the TSDF Site ID or Generator Site ID or Alternate TSDF Site or any Transporter site provided on the manifest.
    - 2.4. If the User does not have the appropriate permissions the system will stop the processing the manifest and generates the following error:
    - E_SitePermissions: The user does not have Industry Permissions for either the TSDF Site or Generator Site or Alternate TSDF Site or any Transporter site specified in provided JSON
3. If the User was Authenticated and Authorized, the system will process the request
    - 3.1. The system will validate the provided Manifest JSON and Attachment Document
        - 3.1.1. The system will validate the e-Manifest fields according to the rules described in section Manifest entities and fields validation for Create Correction service
    - 3.2. If no errors or warnings are generated during the validation process, the service will perform the following steps:
        - 3.2.1. If the current manifest status is either “Signed” or “Corrected” then the service creates a new manifest version with provided, the Manifest information. The service assigns the “UnderCorrection” status to the new manifest version.
        - 3.2.2. If the current manifest status is “UnderCorrection” then the service updates the existing Manifest with the provided Manifest information.
        - 3.2.3. Returns Manifest Tracking Number, operation status, and operation date to the Requestor.
    - 3.3. If only warnings are generated during the validation process, the service will perform the following steps:
        - 3.3.1. If the current manifest status is either “Signed” or “Corrected” then the service creates a new manifest version with the provided Manifest information. The service assigns the “UnderCorrection” status to the new manifest version.
        - 3.3.2. If the current manifest status is “UnderCorrection” then the service updates the existing Manifest with the provided Manifest information.
        - 3.3.3. Returns Manifest Tracking Number, operation status, operation date and Warning(s) Report to the Requestor.
    - 3.4. If an error is generated during the validation process, the service will perform the following steps:
        - 3.4.1. Generate Error Report with all errors found during the validation process.
        - 3.4.2. Return Error Report to the Requestor.
    - 3.5. If errors and warnings were generated during validation process, the service will perform the following steps:
        - 3.5.1. Generate Error/Warning Report with all errors and warnings found during validation process.

3.5.2. Return Error/Warning Report to the Requestor.

Response JSON Schema

crud-emanifest-return.json Schema for successful and failed submission response

Completed Response Examples

Correction Created Response Without Warnings Example

```
{
"manifestTrackingNumber" : "100001380ELC",
"reportId": "43f52e98-4b01-4b61-8c42-3d6ab42c5bf3",
"date": "2018-10-09T15:18:56.145+0000",
"operationStatus": "CorrectionCreated"
}
```

Correction Updated Response Without Warnings Example

```
{
"manifestTrackingNumber" : "100001380ELC",
 "reportId": "33f561fc-2142-48b5-97eb-c6de0899814f",
 "date": "2018-10-09T15:21:09.306+0000",
 "operationStatus": "CorrectionUpdated"
}
```

Response with Warnings and Response with Errors are identical to the Save and Update services Warning and Error Responses

Errors

Type	code	Description
N/A	E_SecurityApiTokenInvalid	Invalid Security Token
N/A	E_SecurityApiTokenExpired	Security Token is Expired
N/A	E_SecurityApiInvalidStatus	This API Id is no longer active
Authorization	E_IndustryPermissions	The user does not have Industry permissions for any Site
Validation	E_SitePermissions	The user does not have Industry Permissions for either the TSDF Site or Generator Site or Alternate TSDF site or any Transporter site specified in provided JSON
System	E_SystemError	Message describing system error

Revert manifest

Description

The Revert Manifest service deletes the manifest version in the “UnderCorrection” status by provided Manifest Tracking Number.

Service Type

Emanifest Data Service.

Parameters

- Security Token: Security Token received in Authentication. Will be passed in the Header of the request.
- Manifest Tracking Number

Service Invocation Example

PreProduction/Test Environment:
https://rcrainfopreprod.epa.gov/rcrainfo/rest/api/v1/emanifest/manifest/revert/100001380ELC
Production:
https://rcrainfo.epa.gov/rcrainfoprod/rest/api/v1/emanifest/manifest/revert/100001380ELC

Sequence of steps

1. The System will validate the Security Token
   1.1. If the Web Security Token is invalid the system stops the submission and generates the following error:
   - E_SecurityApiTokenInvalid: Invalid Security Token

   1.2. If the Web Security Token expired the system stops the submission and generates the following error:
   - E_SecurityApiTokenExpired: Security Token is Expired

   1.3. If Account was Inactivated after the token was issued the system stops the submission and generates the following error:
   - E_SecurityApiInvalidStatus: This API Id is no longer active
2. The system will perform User Authorization.
   2.1. System will check if the User has Site Manager (Industry) permissions for any Site.
   2.2. If the User does not have permission the System will stop the processing and generates the following error:
   - E_IndustryPermissions: The user does not have industry permissions for any Site
3. The system will process the request
   3.1. The system will check if provided Manifest Tracking Number is valid and in the system.
   3.1.1. Manifest Tracking Number shall be compliant with following rules:
   3.1.1.1. Nine numeric characters + valid three character Manifest Tracking Suffix
   3.1.1.2. If the provided Manifest Tracking Number does not have valid format the processing will be stopped and system generates the following error:
   - E_InvalidManifestTrackingNumber: Provided Manifest Tracking Number has invalid format

   3.1.1.3. If the provided Manifest Tracking Number does not have a valid suffix the processing will be stopped and system generates the following error:
   - E_InvalidManifestTrackingNumberSuffix: Provided Manifest Tracking Number has invalid

   3.1.2. If the Manifest Tracking Number is not in the system the processing will stop and the system generates the following error:
   - E_ ManifestTrackingNumberNotFound: Provided Manifest Tracking Number was not found

   3.2. The system will check if the user has Industry Permissions for the TSDF Site or for the Generator site or for the Alternate TSDF site or any Transporter site provided on the manifest. If the User does not have permission, the system will stop the processing and generates the following error:

- E_SitePermissions: The user does not have Industry Permissions for either the TSDF Site or Generator Site or Alternate TSDF site or any Transporter site specified in provided JSON

3.3. The system will check if the manifest can be reverted

3.3.1. If there is no “UnderCorrection” version for the provided Manifest Tracking Number then the service will stop processing and generates the following error:

- E_NoVersionForUnderCorrectionStatus: Manifest does not contain the Version in 'UnderCorrection' status

3.4. The System Reverts the manifest (Deletes Manifest version in “UnderCorrection” status)

4. On success the system returns JSON containing
    - Manifest Tracking Number
    - Operation Status: “Reverted”
    - Current Version Number: current version number
    - Date: Date/Time of the operation
5. If any system errors were encountered during processing, the system will return:
    - error: containing error code, error message, error id and error date

Response JSON Schema for successful return

crud-emanifest-success.json

Completed Response Example

```
{
 "manifestTrackingNumber" : "100001380ELC",
 "operationStatus": "Reverted",
"currentVersionNumber": 2,
  "date": "2017-07-27T19:13:28.628+0000"
}
```

Errors

Type	code	Description
N/A	E_SecurityApiTokenInvalid	Invalid Security Token
N/A	E_SecurityApiTokenExpired	Security Token is Expired
N/A	E_SecurityApiInvalidStatus	This API Id is no longer active
Authorization	E_IndustryPermissions	The user does not have Industry permissions for any Site
Authorization	E_SitePermissions	The user does not have Industry Permissions for either the TSDF Site

		or Generator Site or Alternate TSDF site or any Transporter site specified in provided JSON
Validation	E_InvalidManifestTrackingNumber	Provided Manifest Tracking Number has invalid format
Validation	E_InvalidManifestTrackingNumberSuffix	Provided Manifest Tracking Number has invalid suffix
Validation	E_ManifestTrackingNumberNotFound	Provided Manifest Tracking Number was not found
Validation	E_NoVersionForUnderCorrectionStatus	Manifest does not contain the Version in 'UnderCorrection' status
System	E_SystemError	Message describing system error

Validation Error Return Example

```
{

      "code": "E_ManifestTrackingNumberNotFound ",
      "message": "Provided Manifest Tracking Number was not Found"
      "errorId" : "9v7a9808-s35a5-4874-928f-12fc3def4b56",
      "date" : "2017-06-23T23:15:45.095+0000"
}
```

Manifest entities and fields validation for Save service

1. If current Emanifest.originType == “Mail” then the service generates the following error:

```
{
     "message": "Manifests submitted via Mail cannot be saved via Save service",
     "field": "Emanifest.originType",
    “value”: “originType value”
}
```

1.

2. Emanifest.manifestTrackingNumber

   2.1. If submissionType is “FullElectronic” or “Hybrid” and Emanifest.manifestTrackingNumber is provided, the service generates the following warning:

```
 {
      "message": "Provided Manifest Tracking Number will be ignored",
      "field": "Emanifest.manifestTrackingNumber",
        “value”: “manifestTrackingNumber value”
```

```
}
```

2.2. If submissionType is “DataImage5Copy” or “Image” the following applies

2.2.1. If Emanifest.manifestTrackingNumber is not provided, the service generates the following error:

```
{
      “message”: “Mandatory Field is not Provided”,
      “field”: “Emanifest.manifestTrackingNumber”
}
```

2.2.2. If Emanifest.manifestTrackingNumber is provided, the service will check if manifestTrackingNumber has a valid format. If manifestTrackingNumber has invalid format, the service generates the following error:

```
{
     "message": "Invalid Field Format. 9 digits followed by 3 upper case letters is expected",
     "field": "Emanifest.manifestTrackingNumber",
      “value”: “manifestTrackingNumber value”
}
```

2.2.3. If manifestTrackingNumber has a valid format, the service checks if the provided manifestTrackingNumber suffix is valid. All printed suffixes except ELC are valid If the suffix is invalid the service generates the following error:

```
{
     "message": "Invalid Manifest Tracking Number Suffix is Provided",
     "field": "Emanifest.manifestTrackingNumber",
      “value”: “manifestTrackingNumber value”
}
```

2.2.4. If manifestTrackingNumber has a valid format, the service will check if a manifest with the provided manifestTrackingNumber is already stored in the e-Manifest database. If the manifest with the provided manifestTrackingNumber is already stored in the e-Manifest database, the system generates the following error:

```
{
     “message”: “Manifest with provided Manifest Tracking Number is already stored in the e-
Manifest database”,
     “field”: “Emanifest.manifestTrackingNumber”,
      “value”: “manifestTrackingNumber value”
}
```

3. Emanifest.status

3.1. If the submission type is “DataImage5Copy” or “Image” and provided status is not “ReadyForSignature”, the service generates the following warning:

```
{
      “message”: “Provided Status will be ignored. e-Manifest will be assigned
ReadyForSignature status”,
      “field”: “Emanifest.status”,
      “value”: “status value”
}
```

3.2. If the submission type is “FullElectronic” or “Hybrid” following applies

3.2.1. If status is not provided, the service generates the following warning:

```
{
    “message”: “Field is not Provided. Pending status will be assigned”,
    “field”: “Emanifest.status”
}
```

3.2.2. If provided status is not “Pending” or “Scheduled” the service generates the following warning:

```
{
    “message”: “Invalid Value is Provided. Pending status will be assigned”,
    “field”: “Emanifest.status”,
    “value”: “status value”
}
```

4. Emanifest.submissionType

4.1. If the submissionType is not: “FullElectronic”, “DataImage5Copy”, “Image”, or “Hybrid” the service generates the following error:

```
{
    “message”: “Invalid Field Format. One of the following values “FullElectronic”,
“DataImage5Copy”, “Image”, or “Hybrid” is expected”,
    “field”: “Emanifest.submissionType”,
    “value”: “status value”
}
```

4.2. If submissionType is not provided, the service generates the following error:

```
{
    “message”: “Mandatory field is not provided”,
    “field”: “Emanifest.submissionType”
}
```

5. Validation of the State Waste Codes applies only for the “FullElectronic”, “DataImage5Copy” or “Hybrid” submission types.

5.1. It is possible that if the Waste is not hazardous no State Codes are provided (Federal Waste codes shall not be provided for non-hazardous waste. This is covered in section Hazardous Waste Validation). Manifest is not valid without a Waste Code. At least one State Waste Code shall be provided for the Manifest.

5.2. If submissionType is “FullElectronic”, “DataImage5Copy” or “Hybrid” and no State Waste Codes are provided, the service generates the following error:

```
{
    “message”: “Manifest does not have any Waste Codes. Valid Manifest requires at least
one Waste Code.”
“field”: “Emanifest.Waste.hazardousWaste”
}
```

6. Waste(s) Line Numbers Validation applies only for the “FullElectronic”, “DataImage5Copy” or “Hybrid” submission types

    6.1. If applies only for the “FullElectronic”, “DataImage5Copy” or “Hybrid” submission types the following applies

        6.1.1. Emanifest.waste.lineNumber for each Waste must be provided. If lineNumber(s) is/are not provided, the service generates the following error:

```
{
    “message”: “Mandatory Field is not Provided.”,
    “field”: “Emanifest.waste.lineNumber”
}
```

        6.1.2. If lineNumber is not a valid format, the service generates the following error:

```
{
    “message”: "Instance type (string) does not match any allowed primitive type (allowed:
[\"integer\",\"number\"])",
    “field”: “Emanifest.wastes.[i].lineNumber”,
    “value”: “lineNumber value”
}
```

        6.1.3. If one waste is provided and the lineNumber is not equal to 1, the service generates a warning:

```
{
 “message”: “Manifest expected to start with line 1”,
 “field”: “Emanifest.waste.lineNumber”,
 “value”: “order value”
}
```

        6.1.4. If multiple wastes are provided and lineNumbers are not sequential, the service generates the following error:

```
{
 “message”: “Provided Value is not Valid. Sequential waste line numbers are expected”
 “field”: “Emanifest.wastes.lineNumber”,
 “value”: “lineNumber 1”
}
{
 “message”: “Provided Value is not Valid. Sequential waste line numbers are expected”
 “field”: “Emanifest.wastes.lineNumber”,
 “value”: “lineNumber 2”
}
```

7. Transporters applies only for the “FullElectronic”, “DataImage5Copy” or “Hybrid” submission types

    7.1. If the transporters entity is not provided, the service generates the following error:

```
{
    "message": "Mandatory Field is not Provided",
    "field": "Emanifest.transporters"
}
```

8. Transporter(s) order(s) validation applies only for the "FullElectronic", "DataImage5Copy" or "Hybrid" submission types

   8.1. If Transporter.order is not provided, the service generates the following error:

   ```
   {
   "message": "Value is not provided"
   "field": "Emanifest.transporter.order"
   }
   ```

   8.2. If Transporter.order is an invalid format, the service generates the following error:

   ```
    {
            "message": "Invalid Field format. Numeric value expected",
            "field": "Emanifest.transporter.order",
            "value": "order value"
    }
   ```

   8.3. Validate consecutiveness of the Transporters order.

   8.3.1. If one transporter is provided and the order is not equal to 1, the service generates the following error:

   ```
   {
    "message": "Invalid Value",
    "field": "Emanifest.transporter.order",
    "value": "order value"
   }
   ```

   8.3.2. If multiple transporters are provided and orders are not consecutive, the service generates the following error:

   ```
   {
    "message": "Invalid value(s). Consecutive Transporter Orders are expected"
    "field": "Emanifest.transporter.order",
    "value": "order value 1"
   }
   {
    "message": "Invalid value(s). Consecutive Transporter Orders are expected"
    "field": "Emanifest.transporter.order",
    "value": "order value 2"
   }
   ```

9. If submissionType is "Image" Validate Facility Information. See details in section Facility Information Validation for "Image" submission type.

10. If submissionType is "FullElectronic" or "DataImage5Copy" or "Hybrid" Validate Import Information. See details in Import Information Validation section.

11. If submissionType is "FullElectronic" or "DataImage5Copy" or "Hybrid" Validate Rejection Information. See details in Rejection Information Validation section.

12. If submissionType is "FullElectronic" or "DataImage5Copy" or "Hybrid" Validate New Manifest Information. See details in New Manifest Information Validation section.

13. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Discrepancy Information. See details in Discrepancy fields validation section.
14. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Residue Information. See details in Residue Fields Validation section.
15. Validate Generator Information. See details in Generator Site Id and Site Information Validation section.
16. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Transporters Information. See details in Transporter Site ID and Site Information Validation section.
17. Validate Designated Facility Information. See details in Designated Facility (TSDF)/Generator Site Id and Site Information Validation section.
18. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Wastes Information. See details in Waste Validation section.
19. If submissionType is “DataImage5Copy” Validate Paper Signature Info. See details in Paper Signature Info Validation section.
20. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Manifest Additional Information. See details Manifest Additional Information Validation section.
21. If submissionType is DataImage5Copy or “Image” validate Attachment Information. See details in Attachment Validation section.

Manifest entities and fields validation for Update service

2. If current Emanifest.originType == “Mail” then the service generates the following error:

```
{
    "message": "Manifests submitted via Mail cannot be updated via Update service",
    "field": "Emanifest.originType",
   “value”: “originType value”
}
```

3. Emanifest.manifestTrackingNumber

   1.1. If Emanifest.manifestTrackingNumber is not provided, the service generates the following error:

```
{
    "message": "Mandatory Field is not Provided",
    "field": "Emanifest.manifestTrackingNumber"
}
```

   1.2. Emanifest.manifestTrackingNumber is provided and it has an invalid format, the service generates an error:

```
{
    "message": "Invalid Field Format. 9 digits followed by 3 upper case letters is expected",
    "field": "Emanifest.manifestTrackingNumber",
   “value”: “manifestTrackingNumber value”
}
```

1.3. If the manifestTrackingNumber has a valid format, the service checks if the provided manifestTrackingNumber suffix is valid. If the suffix is invalid, the service generates the following error:

```
 {
        "message": "Invalid Manifest Tracking Number Suffix is Provided",
        "field": "Emanifest.manifestTrackingNumber",
          “value”: “manifestTrackingNumber value”
 }
```

1.4. If the manifestTrackingNumber has a valid format, the service checks if a Manifest with the provided manifestTrackingNumber is already stored in the e-Manifest database. If the Manifest was not found in the e-Manifest database, the service generates the following error:

```
    {
          "message": "Manifest with provided Manifest Tracking Number was not found.
    Manifest cannot be updated",
          "field": "Emanifest.manifestTrackingNumber",
        “value”: “manifestTrackingNumber value”
    }
```

4. Emanifest.status.

2.1. If submissionType is “FullElectronic” or “Hybrid”, then the status can be changed from “Pending” to “Scheduled” via update service. Statuses after “Scheduled” cannot be changed via the update service. Status transitions after “Scheduled” will be done by e-Manifest and are triggered by the following events:

2.3.1. If there is one transporter, the following status transition is possible:

- Scheduled->ReadyForSignature: The status transition is triggered when the Transporter signs.
- ReadyForSignature->Signed: The status transition is triggered when the TSDF signs.

2.3.2. If there is more than one transporter, the following status transition is possible:

- Scheduled->InTransit: The status transition is triggered when the 1st Transporter signs.
- InTransit -> ReadyForSignature: Status transition is triggered when last Transporter signs.
- ReadyForSignature->Signed: The status transition is triggered when TSDF signs.

2.3.3. If the current status is “Pending” and provided status is not “Pending” or “Scheduled”, the service generates the following error:

```
{
      “message”: “For the FullElectronic and Hybrid submission type Manifest status
cannot be updated to statuses after “Scheduled” via Update Manifest service”,
      “field”: “Emanifest.status”,
      “value”: “status value”
}
```

2.3.4. If the current status is “Scheduled” and provided status is not “Scheduled”, the service generates a warning:

```
{
    “message”: “Provided Value will be ignored. For the FullElectronic and Hybrid submission type Manifest status cannot be updated to statuses before or after “Scheduled” via Update Manifest service”,
    “field”: “Emanifest.status”,
    “value”: “status value”
}
```

2.3.5. If the current status is “InTransit” and provided status is not “InTransit”, the service generates a warning:

```
{
    “message”: “Provided Value will be ignored. For the FullElectronic and Hybrid submission type Manifest status cannot be updated to statuses before or after “Scheduled” via Update Manifest service”,
    “field”: “Emanifest.status”,
    “value”: “status value”
}
```

2.3.6. If the current status is “ReadyForSignature” and provided status is not “ReadyForSignature”, the service generates a warning:

```
{
    “message”: “Provided Value will be ignored. For the FullElectronic and Hybrid submission type Manifest status cannot be updated to statuses before or after “Scheduled” via Update Manifest service”,
    “field”: “Emanifest.status”,
    “value”: “status value”
}
```

2.3.7. If the current status is “Signed” and provided status is not “Signed”, the service generates a warning:

```
{
    “message”: “Provided Value will be ignored. For the FullElectronic and Hybrid submission type Manifest status cannot be updated to statuses before or after “Scheduled” via Update Manifest service”,
    “field”: “Emanifest.status”,
    “value”: “status value”
}
```

2.3.8. If status is not provided, the service does not generate warning or error

2.4. If submissionType is “DataImage5Copy” or “Image” then status cannot be changed via the update service. The status transition will be done by e-Manifest and is triggered by the following event:

- ReadyForSignature -> Signed: Status transition is triggered when the TSDF electronically signs.

2.4.1. If Manifest status is not provided, no warnings or errors shall be generated.

2.4.2. The Manifest can be submitted only in the ReadyForSignature status. Any other status will be ignored with a warning:

```
{
        “message”: “Provided Manifest status will be ignored. For the DataImage5Copy submission type Manifest status is ReadyForSignature and cannot be updated to any other status via Update Manifest service”,
        “field”: “Emanifest.status”,
        “value”: “status value”
}
```

2.4.3. If the provided status is different than the current status, the service generates a warning:

```
{
        “message”: “Provided Manifest status will be ignored. For the DataImage5Copy submission type Manifest status cannot be updated to any other status via Update Manifest service”,
        “field”: “Emanifest.status”,
        “value”: “status value”
}
```

3. Emanifest.submissionType.

3.3. If the current status >= Scheduled and provided submissionType is not equal to the current submissionType then the service generates the following warning:

```
{
  “message”: “Provided Submission Type will be ignored. Submission Type cannot be updated”,
  “field”: “Emanifest. submissionType”,
  “value”: “submissionType value”
}
```

4. If submissionType is “Image” Validate Facility Information. See details in section Facility Information Validation for “Image” submission type.
5. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate the State Waste Codes. This section is identical to Validation of the State Waste Codes in Manifest entities and fields validation.
6. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Waste(s) Line Numbers. This section is identical to Waste(s) Line Numbers Validation in Manifest entities and fields validation.
7. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Transporter(s) order(s). This section is identical to Transporter(s) order(s) validation in Manifest entities and fields validation.
8. Validate Generator Information. See details in Generator Site Id and Site Information Validation section.

9. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Import Information. See details in Import Information Validation section.
10. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Rejection Information. See details in Rejection Information Validation section.
11. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate New Manifest Information. See details in New Manifest Information Validation section.
12. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Discrepancy Information. See details in Discrepancy fields validation section.
13. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Residue Information. See details in Residue Fields Validation section.
14. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Transporters Information. See details in Transporter Site ID and Site Information Validation section.
15. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Designated Facility Information. See detail in Designated Facility (TSDF)/Generator Site Id and Site Information Validation section.
16. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Wastes Information. See details in Waste Validation section.
17. If submissionType is “DataImage5Copy” validate Paper Signature Information. See details in Paper Signature Info Validation section.
18. If submissionType is “FullElectronic” or “DataImage5Copy” or “Hybrid” Validate Manifest Additional Information. See details section Manifest Additional Information Validation.
19. If submissionType is DataImage5Copy or “Image” validate Attachment Information. See details in Attachment Validation section.

Facility Information Validation for “Image” submission type

1. For the Image submission type one of the following facility EPA Site Ids must be provided: Designated Facility EPA Site Id, Generator EPA Site Id or Alternate Designated Facility Site Id
2. Only one facility EPA site id must be provided. e-Manifest will determine the “billable” facility by the provided EPA Site Id.
3. The following Facility EPA site Ids shall be provided for the following scenarios:
    - Original Manifest, no Full Rejection: Designated Facility EPA Site Id shall be provided.
    - Original Manifest, Full Rejection to Alternate Designated Facility: Alternate Designated Facility EPA Site Id shall be provided.
    - Original Manifest, Full Rejection to The Generator: Designated Facility EPA site Id shall be provided.
    - New Manifest shipping waste back to original Generator (New Manifest is the manifest created as a “result” of the Original manifest rejection or residue): Generator Facility EPA Site Id shall be provided. This EPA Site Id shall contain the Original Designated Facility Site Id.
    - New Manifest shipping waste back to another Designated Facility: Designated Facility EPA Site Id shall be provided.

4. If one of the following Ids: Emanifest.designatedFacility.epaSiteId and Emanifest.generator.epaSiteId and Emanifest.rejectionInfo.alternateDesignatedFacility.siteId is not provided then the service will return an authorization error.
5. If the provided EPA Site Id is not registered, then the service will return an authorization error.
6. If more than one of the following facility EPA Site Ids is provided, then the service generates the following errors (example is for the case if both Generator and Designated Facility EPA Site Ids are provided):

   ```
   {
   "message": “For Image submission type only one of the following facility EPA site id must be provided: Designated Facility EPA Site Id or Generator EPA Site Id or Alternate Designated Facility Site Id.”
   "field": "Emanifest.generator.epaSiteId”
   }
   {
   "message": “For Image submission type only one of the following facility EPA site id must be provided: Designated Facility EPA Site Id or Generator EPA Site Id or Alternate Designated Facility Site Id.”
   "field": "Emanifest.designatedFacility.epaSiteId”
   }
   ```

7. If Emanifest.designatedFacility.epaSiteId or Emanifest.generator.epaSiteId or Emanifest.rejectionInfo.alternateDesignatedFacility.siteId is provided, the following applies

   7.1. If any following Site information is provided then the service generates the following warning:

   - Site Name
   - Site Address
   - Mailing Address

   ```
   {
        "message": “Registered site was found. Provided site information will be ignored and replaced with registered information”,
        "field": "Emanifest.designatedFacility.name /siteAddress/mailingAddress",
        “value”: “name/siteAddress/mailingAddress”
   }
   ```

   7.2. Validation of the Contact information is identical to #2 in section Designated Facility (TSDF)/Generator Site Id and Site Information Validation

   7.3. If Emanifest.wastes are provided then the service generates the following warning:

   ```
   {
   "message": “Provided Value will be Ignored.”,
   "field": "Emanifest.wastes”
   }
   ```

   7.4. If Emanifest.residue == true then the service generates the following warning:

   ```
   {
   "message": “Provided Value will be Ignored. Emanifest.residue = false will be assigned”,
   ```

```
"field": "Emanifest.residue”,
"value": “residue value”
}
```

7.5. If Emanifest.import == true then the service generates the following warning:

```
{
"message": “Provided Value will be Ignored. Emanifest.import = false will be assigned”,
"field": "Emanifest.import”,
"value": “import value”
}
```

7.6. The service stores following information as a part of the manifest data:

- Emanifest.residue = false
- Emanifest.import = false

8. If Emanifest.designatedFacility.epaSiteId or Emanifest.generator.epaSiteId is provided then the following applies

8.1. If Emanifest.rejection == true, the service generates the following warning:

```
{
"message": “Provided Value will be Ignored. Emanifest.rejection = false will be assigned”,
"field": "Emanifest.rejection”,
"value": “ rejection value”
}
```

8.2. If any of the Emanifest.rejectionInfo fields are provided, the service generates the following warning:

```
{
"message": “Provided Value will be Ignored.”,
"field": "Emanifest.rejectionInfo.{field}”,
"value": “field value”
}
```

9. If Emanifest.designatedFacility.epaSiteId or Emanifest.rejectionInfo.alternateDesignatedFacility.siteId is provided, the following applies

9.1. If Emanifest.containsPreviousRejectOrResidue == true, the service generates the following warning:

```
{
"message": “Provided Value will be Ignored. Emanifest.containsPreviousRejectOrResidue = false will be assigned” ”,
"field": "Emanifest.containsPreviousRejectOrResidue”,
"value": “containsPreviousRejectOrResidue value”
}
```

10. If Emanifest.rejectionInfo.alternateDesignatedFacility.siteId is provided, the following applies

10.1. If Emanifest.rejection == false, the service generates the following warning:

```
{
"message": “Provided Value will be Ignored. Emanifest.rejection = true will be assigned”,
"field": "Emanifest.rejection”,
"value": “ rejection value”
}
```

10.2. If Emanifest.rejectionInfo.transporterOnSite ==false, the service generates the following warning:

```
{
"message": “Provided Value will be Ignored. Emanifest.rejectionInfo.transporterOnSite = true will be assigned”,
"field": "Emanifest.rejectionInfo.transporterOnSite”,
"value": “rejectionInfo.transporterOnSite value”
}
```

10.3. If Emanifest.rejectionInfo.rejectionType ==“PartiaReject”, the service generates the following warning:

```
{
"message": “Provided Value will be Ignored. Emanifest.rejectionInfo.rejectionType = “FullReject” will be assigned”,
"field": "Emanifest.rejectionInfo.rejectionType”,
"value": “rejectionInfo.rejectionType  value”
}
```

10.4. If Emanifest.rejectionInfo.alternateDesignatedFacilityType ==“Generator” the service generates the following warning:

```
{
"message": “Provided Value will be Ignored. Emanifest.rejectionInfo. alternateDesignatedFacilityType = “Tsdf” will be assigned”,
"field": "Emanifest.rejectionInfo. alternateDesignatedFacilityType”,
"value": “rejectionInfo. alternateDesignatedFacilityType value”
}
```

10.5. If rejectionInfo.newManifestTrackingNumbers are provided, the service generates the following warning:

```
{
 “message”: ”Provided Field will be Ignored. New Manifest(s) are not created if Transporter is On Site”,
 “field”: “Emanifest.RejectionInfo.newManifestTrackingNumbers”
}
```

10.6. The service stores following information as a part of the manifest data:

- Emanifest.rejection = true
- Emanifest.rejectionInfo.transporterOnSite = true
- Emanifest.rejectionInfo.rejectionType = “FullReject”
- Emanifest.rejectionInfo.alternateDesignatedFacilityType = “Tsdf”

Designated Facility (TSDF)/Generator Site Id and Site Information Validation

TSDF can act either as TSDF or as Generator. This will be determined during the authorization step. If TSDF acts as a TSDF the system requires the handler ID to be provided as the TSDF Site ID. If other information for the Site is passed, it will be ignored with warnings and only the registered information will be used. If the TSDF acts as a Generator the system requires the handler's ID to be placed in the Generator Site Id

1. If submissionType is "FullElectronic" then the following applies
   1.1. 1.2. If Emanifest.generator.epaSiteId is not provided then the service generates the following error:
   {
   "message": "Mandatory Field is not Provided",
   "field": "Emanifest. generator.epaSiteId",
   }
   1.2. If Emanifest. generator.epaSiteId has wrong format then the service generates the following error:
   {
   "message": "Invalid Field Format",
   "field": "Emanifest. generator.epaSiteId",
   "value": " epa site id value"
   }
   1.3. If Emanifest.generator.epaSiteId is not registered in RCRAInfo then the service generates the following error:
   {
   "message": "Provided Generator Facility Id is not registered in RCRAInfo",
   "field": "Emanifest. generator.epaSiteId",
   "value": " epa site id value"
   }

2. If submissionType is "FullElectronic", "DataImage5Copy" or "Hybrid", then the following applies
   2.1. If Emanifest.designatedFacility.epaSiteId is not provided then the service generates the following error:
   {
   "message": "Mandatory Field is not Provided",
   "field": "Emanifest.designatedFacility.epaSiteId",
   }
   2.2. If Emanifest.designatedFacility.epaSiteId has wrong format then service generates the following error:
   {
   "message": "Invalid Field Format",
   "field": "Emanifest.designatedFacility.epaSiteId",
   "value": " epa site id value"

```
}
```

2.3. If Emanifest.designatedFacility.epaSiteId is not registered in RCRAInfo then service generates the following error:

```
{
“message”: “Provided Designated Facility Id is not registered in RCRAInfo”,
“field”: “Emanifest.designatedFacility.epaSiteId”,
“value”: “ epa site id value”
}
```

2.4. If the Site Contact Phone number is not provided and registered in RCRAInfo Site does not contain the Site Contact Phone Number, the system generates the following error:

```
{
“message”: “Mandatory Field is not Provided”,
“field”: “Emanifest.designatedFacility.contact.phone.number”,
}
```

2.5. If the Site Contact Phone number is provided, the system will check the format.

2.5.1. If the provided phone number has a valid format the system will store site contact phone number into e-Manifest database

2.5.2. If the provided phone number has an incorrect format the system generates the following warning:

```
{
“message”: "String \"{provided phone value}\" is too long (length: {provided phone value length}, maximum allowed: 12)",
“field”: “Emanifest.designatedFacility.contact.phone.number “,
“value”: “number value”
}
```

2.6. If the Site Contact Phone Extension is provided, the system will check the format.

2.6.1. If provided phone extension has a valid format, the system will store the site contact phone extension into e-Manifest database

2.6.2. If the provided phone has an incorrect format, the system generates the following warning:

```
{
“message”: "String \"{provided phone extension value}\" is too long (length: {provided phone value length}, maximum allowed: 6)",
“field”: “Emanifest.designatedFacility.contact.phone.extension “,
“value”: “extension value”
}
```

2.7. If any of the following TSDF Site Information entities were provided:

- Site Name
- Site Address
- Mailing Address

The system generates the following warning:

```
{
    "message": “Registered site was found. Provided site information will be ignored and
    replaced with registered information”,
    "field": "Emanifest.designatedFacility.name /siteAddress",
    “value”: “name/siteAddress”
}
```

2.8. If the user is authorized to use the e-Manifest Save service for the provided Generator Site ID (It means that TSDF acts as Generator), then the provided Site ID was already validated during authorization and the system will perform following steps:

2.9. If the Site Contact Phone number is not provided in the JSON and site the is registered in RCRAInfo, the system checks if the site has a Contact Phone value registered for the site. If the Contact Phone is not registered in the system for that site, the system generates the following error:

```
{
"message": “Mandatory Field is not provided”
"field": "Emanifest.generator.contact.Phone.number”
}
```

2.10. If the Site Contact Phone number is provided and registered in RCRAinfo the system checks that the phone number is in the correct format.

2.10.1. If the provided phone number has a valid format, the system will store site contact phone number into the e-Manifest database

2.10.2. If provided phone number has an incorrect format, the system will store registered in RCRAInfo Contact Phone number into e-Manifest database and generate the following warning:

```
{
“message”: "String \"{provided phone value}\" is too long (length: { provided phone value
length}, maximum allowed: 12)",
“field”: “Emanifest.generator.contact.phone.number “,
“value”: “number value”
}
```

2.11. If the Site Contact Phone number is provided and registered in RCRAinfo site does not contain Contact Phone.

2.11.1. If the provided phone number has valid format, the system will store site contact phone number into e-Manifest database

2.11.2. If provided phone number has incorrect format, the system generates the following error:

```
{
“message”: "String \"{provided phone value}\" is too long (length: { provided phone value
length}, maximum allowed: 12)",
“field”: “Emanifest.generator.contact.phone.number “,
“value”: “number value”
}
```

2.12. Validate emergency phone

2.12.1. If emergency phone number is not provided, the system generates the following error:

```
{
“message”: “Mandatory Field is not provided”
“field”: “Emanifest.generator.emergencyPhone.number”
}
```

2.12.2. If emergency phone number is provided and has an invalid format, the system generates the following error:

```
{
    “message”: "String \"{provided phone value}\" is too long (length: { provided phone
    value length}, maximum allowed: 12)",
    “field”: “Emanifest.generator.emergencyPhone.number “,
    “value”: “phone number value”
}
```

2.12.3. If emergency phone extension is provided and has an invalid format, the system generates the following warning:

```
{
    "message": "String \"{provided phone value}\" is too long (length: { provided phone
    value length}, maximum allowed: 6)",
    "field": "Emanifest.generator.emergencyPhone.extension ",
    “value”: “extension value”
}
```

2.13. If Site Name, Site Location, Site Mailing Address are provided, the system generates the following warning:

```
{
    "message": “Provided Values will be Ignored. RCRAInfo values will be used”,
    "field": "Emanifest.genrator.name/mailingAddress/siteAddress ",
    “value”: “name/mailingAddress/siteAdress value”
}
```

2.14. If the contact email was provided, the system will check the format

2.14.1. If provided contact email has a valid format the system will store site contact email into e-Manifest database

2.14.2. If the provided contact email has an incorrect format the system generates the following warning:

```
{
"message": “Invalid Field format. Valid email format is expected.”,
"field": "Emanifest.designatedFacility.contact.email ",
“value”: “email value”
}
```

3. Update validation.

3.1. Original epaSiteId submitted via Save e-Manifest service cannot be updated via Update e-Manifest service.

3.2. If the provided designatedFacility.epaSiteId does not match current designatedFacility.epaSiteId in the Update e-Manifest service the following warning will be generated:

```
{
    “message”: “Provided Field will be ignored. Originally Submitted Designated Facility EPA Site Id cannot be updated”,
    “field”: “Emanifest.designatedFacility.epaSiteId”,
    “value”: “epaSiteId value”
}
```

3.3. designatedFacility.Contact.phone number can be updated at any status

Generator Site Id and Site Information Validation

1. If submissionType is “FullElectronic” then the Generator is valid if the Generator Site ID is registered in RCRAInfo and the Generator has at least one User with the e-Manifest Certifier Role and this user has a Received ESA.
2. If submissionType is “FullElectronic” the following applies

2.1. If the Generator Site ID is not provided, the service generates the following error:

```
    {
        “message”: “Mandatory Field is not Provided. For FullElectronic submission type registered Generator Site Id must be provided”,
        “field”: “Emanifest.designatedFacility.epaSiteId”
}
```

2.2. If the Generator Site ID has an invalid format the service generates the following error:

```
    {
        “message”: “Invalid Field Format. For FullElectronic submission type registered Generator Site Id must be provided ”,
        “field”: “Emanifest.generator.epaSiteId”,
        “value”: “Site ID value”
}
```

2.3. If the Generator Site ID is not registered in RCRAInfo the service generates the following error:

```
    {
        “message”: “For FullElectronic submission type a registered Generator Site Id must be provided ”,
        “field”: “Emanifest.generator.epaSiteId”,
        “value”: “Site ID value”
}
```

2.4. If the Generator Site ID is registered in RCRAInfo and there are no users with the e-Manifest Certifier role for the provided Generator the service generates the following error:

```
    {
        “message”: “Site doesn't have any users with Certifier role or with ESA status
    'Received”,
        “field”: “Emanifest.designatedFacility.epaSiteId”,
        “value”: “Site ID value”
}
```

2.5. If the Generator with the provided Site ID is registered and there are no users with a received ESA, the service generates the following error:

```
    {
        “message”: “Site doesn't have any users with Certifier role or with ESA status
    'Received”,
        “field”: “Emanifest.designatedFacility.epaSiteId”,
        “value”: “Site ID value”
}
```

3. If submissionType is “DataImage5Copy” or “Hybrid”, then registered and non-registered Generators are valid and the following cases are possible for the Generator Site ID and Generator Information:

3.1. If the Generator is registered in RCRAInfo and service requester does not intend to provide different (than registered) Site Information, then only the epaSiteId shall be provided. All site information will be obtained from RCRAInfo.

3.2. If the Generator is registered in RCRAInfo and service requester intends to provide different (than registered) Site Information, then the following site information entities must be provided:

- Site ID
- Site Name
- Site Address (all fields)
- Mailing Address (all fields)
- Contact Phone

Provided Site Information will be validated and if valid will be stored in e-Manifest.

3.3. If Generator is not registered in RCRAInfo, then following site information must be provided:

- Site Name
- Site Address (all fields)
- Mailing Address (all fields)
- Contact phone

Requester also can provide Site ID which is an optional field for this case. If Site ID is provided, the system will check if there is a registered Generator for the provided Site ID

3.4. For all above cases Emergency Phone Number must be provided

3.5. If the manifest status is InTransit or thereafter then Generator information cannot be updated.

3.6. If different than currently stored Generator information is provided, the service generates the following warning:

```
        {
            "message": “Provided Generator information will be ignored. Generator Information
cannot be updated if the manifest status is InTransit or thereafter”
            "field": "Emanifest.generator”
        }
```

3.7. If the Site ID is not provided AND any of the mandatory Site information entities are not provided, the service generates the following error(s):

```
{
            "message": “Mandatory Field is not provided”
            "field": "Emanifest.generator.epaSiteId”
}
{
            "message": “Mandatory Field is not provided”
            "field": "Emanifest.generator.siteAddress”
}
{
            "message": “Mandatory Field is not provided”
            "field": "Emanifest.generator.mailingAddress”
}
```

3.8. If the Site ID is provided AND Site Address information is not provided, the system performs the following steps:

3.8.1. If Site ID is not valid the service generates the following error:

```
{
            "message": “Provided Value is not Valid. Does not match format of: Two Letter Activity
Location Code + Up to 10 alphanumeric characters”,
            "field": "Emanifest.generator.epaSiteId ",
            “value”: “EPA site Id value”
}
```

3.8.2. If Site ID is valid and found in RCRAInfo, the system will obtain the Site information from RCRAInfo and store it into the Manifest. System sets modified= false, registered = true

3.8.3. If the Site ID is not found, the service generates the following error:

```
{
            "message": “Provided Value is not Found”,
            "field": "Emanifest.generator.epaSiteId",
            “value”: “EPA Site ID value”
}
```

3.9. If the Site ID is provided and the Site Information entities are provided, the system performs the following steps:

3.9.1. If the site name is not provided, the service generates the following error:

```
{
      "message": “Mandatory Field is not Provided”,
    "field": "Emanifest.generator.name"
}
```

3.9.2. If the site name is not valid (exceeds maximum length), the service generates the following error:

```
{
    "message": “Provided Value is not Valid. Exceeded maximum length of 80”,
    "field": "Emanifest.generator.name",
    “value”: “name value”
}
```

3.9.3. Validate the mandatory location address fields:

- address1 (50 Character maximum)
- city (25 Character maximum)
- state.code (2 Character State abbreviation)
- zip (14 Character maximum)

3.9.3.1. If any of the mandatory location address fields are not provided or invalid, the service generates the following error:

```
{
    “message”: “Value is not Provided/Provided Value is not Valid”,
    “field”: “Emanifest.generator.siteAddress.address1/city/state/zip”,
    “value”: “address1/city/state/zip value”
}
```

3.9.4. Validate the mandatory mailing address fields:

- address1 (50 Character maximum)
- city (25 Character Maximum)
- state.code (2 Character State Abbreviation)
- zip (14 Character Maximum)

3.9.4.1. If any of the mandatory mailing address fields are not provided or invalid, the service generates the following error:

```
{
    "message": “Value is not Provided/Provided Value is not Valid”,
    "field": "Emanifest.generator.mailingAddress.address1/city/state/zip",
    “value”: “address1/city/state/zip provided value”
}
```

3.10. Validate provided Site ID

3.10.1. If Site ID is not valid, the service generates the following warning:

```
{
    "message": “Provided value is not valid. Two Letter Activity Location Code + Up
to 10 alphanumeric characters”,
    "field": "Emanifest.generator.epaSiteId”,
    “value”: “EPA Site ID value”
}
```

3.10.2. If the Site ID is valid the system checks if the site is registered

3.10.3. If the site is registered, the system validates if the provided site address state code matches Site ID activity location.

3.10.3.1. If the state code does not match the Site ID activity location, the service generates the following error:

```
{
        "message": “Location State Code is different than Site Id Activity
Location”,
        "field": "Emanifest.generator.siteAddress.state.code”,
        “value”: “site address state code value”
}
```

3.10.4. If the site is not registered, the service generates a warning:

```
{
    "message": “Site with provided EPA Site ID is not registered”,
    "field": "Emanifest.generator.epaSiteId”,
    “value”: “Site ID value”
}
```

3.10.4.1. Store provided Site Information (if no other manifest errors were found)

4. If the Site ID is not provided AND the Site Information is provided, the system performs the following steps:

4.1. Validate the provided Site Information.

4.1.1. Validate the site name

4.1.2. If the site name is not valid the service generates the following error:

```
{
    "message": “Provided Value is not Valid”,
    "field": "Emanifest.generator.name",
    “value”: “name value”
}
```

4.1.3. Validate the mandatory site location address fields:

- address1
- city
- state.code
- zip

4.1.4. If any of the mandatory location address fields are not provided or invalid, the service generates the following error:

```
{
    "message": “Value is not Provided/Provided Value is not Valid”,
    "field": "Emanifest.generator.siteAddress.address1/city/state/zip",
    “value”: “address1/city/state/zip value”
}
```

4.1.5. Validate the mandatory mailing address fields:

- address1
- city

- state.code
- zip

4.1.6. If any of the mandatory mailing address fields are not provided or invalid, the service generates the following error:

```
{
      "message": “Value is not Provided/Provided Value is not Valid”,
      "field": "Emanifest.generator.mailingAddress.address1/city/state/zip",
      “value”: “address1/city/state/zip provided value”
}
```

4.2. If the provided Site Information is valid, but the Site ID is not provided, the system performs the following steps:

4.2.1. The service generates the following warning:

```
{
      "message": “Value is not Provided”,
      "field": "Emanifest.generator.epaSiteId”
}
```

4.2.2. Store the provided Site Information (if no other manifest errors were found)

4.3. If the Site Contact Phone number is not provided and site is registered in RCRAInfo, the system checks if the site has a Contact Phone value registered for the site. If the Contact Phone is not registered in the system for that site, the system generates the following error:

```
{
"message": “Mandatory Field is not provided”
"field": "Emanifest.generator.contact.Phone.number”
}
```

4.4. If the Site Contact Phone number is provided and registered in RCRAinfo the system checks that the phone number is in the correct format.

4.4.1. If the provided phone number has valid format the, system will store site contact phone number into e-Manifest database

4.4.2. If the provided phone number has an incorrect format, the system will store registered in RCRAInfo Contact Phone number into e-Manifest database and generates the following warning:

```
{
"message": "String \"{provided phone value}\" is too long (length: { provided phone value length}, maximum allowed: 12)",
"field": "Emanifest.generator.contact.phone.number ",
“value”: “number value”
}
```

4.5. If the Site Contact Phone number is provided and registered in RCRAinfo the system checks that the phone number is in the correct format.

4.5.1. If the provided phone number has valid format, the system will store site contact phone number into the e-Manifest database

4.5.2. If the provided phone number has an incorrect format, the system generates the following error:

```
{
"message": "String \"{provided phone value}\" is too long (length: {provided phone value length}, maximum allowed: 12)",
"field": "Emanifest.generator.contact.phone.number ",
“value”: “number value”
}
```

4.6. If emergency phone number is not provided, the service generates the following error:

```
{
"message": “Mandatory Field is not provided”
"field": "Emanifest.generator.emergencyPhone.number”
}
```

4.7. If emergency phone number is provided and has an invalid format, the service generates the following error:

```
{
    "message": "String \"{provided phone value}\" is too long (length: {provided phone value length}, maximum allowed: 12)",
    "field": "Emanifest.generator.emergencyPhone.number ",
    “value”: “phone number value”
}
```

4.8. If emergency phone extension is provided and has invalid format, the system generates a warning:

```
{
    "message": "String \"{provided phone value}\" is too long (length: {provided phone value length}, maximum allowed: 6)",
    "field": "Emanifest.generator.emergencyPhone.extension ",
    “value”: “extension value”
}
```

4.9. If the provided Contact email has an invalid format, the service generates a warning:

```
{
    "message": “Provided Value will be ignored. Does not match email format”,
    "field": "Emanifest.generator.contact.email",
    “value”: “email value”
}
```

5. Update Validation

5.1. If the current status > Scheduled and provided status > Scheduled, Generator information cannot be updated

5.2. Service will check if any of provided following Generator fields are different than currently stored values:

- Site Id
- Site Name

- Site Address (all fields)
- Mailing Address (all fields)
- Contact (all fields)

5.3. If any of the Generator information fields are different than currently stored, the service generates the following warning:

```
{
    "message": “Provided Value will be ignored. For statuses InTransit, Received, and
ReadyForSignature the Generator information cannot be updated.”,
    "field": "Emanifest.generator.epaSiteId/name/siteAddress/mailingAddress/contact",
    “value”: “field value”
}
```

Transporter Site ID and Site Information Validation

1. If submissionType is “FullElectronic” or “Hybrid”, then the Transporter is valid if the Transporter is registered in RCRAInfo and the Transporter has at least one User with the e-Manifest Certifier Role and this user has a Received ESA.

1.1. If the Transporter Site ID is not provided, the service generates schema validation error

1.2. If the Transporter Site ID has an invalid format the service generates schema validation error

1.3. If the Transporter Site ID is not registered the service generates the following error:

```
{
    “message”: “For FullElectronic submission type registered Transporter Site Id must be
provided ”,
    “field”: “Emanifest.transporter.epaSiteId”,
    “value”: “Site ID value”
}
```

1.4. If the Transporter Site ID is registered and there are no users with the e-Manifest Certifier role for the provided Transporter, the service generates the following error:

```
{
    “message”: “Site doesn't have any users with Certifier role or with ESA status
'Received”,
    “field”: “Emanifest.transporter.epaSiteId”,
    “value”: “Site ID value”
}
```

1.5. If the Transporter Site ID is registered and there are no users with a recieved ESA,the service generates the following error:

```
{
    “message”: “Site doesn't have any users with Certifier role or with ESA status
'Received”,
    “field”: “Emanifest.transporter.epaSiteId”,
    “value”: “Site ID value”
}
```

2. If submissionType is “DataImage5Copy”, then registered and non-registered Transporters are valid

2.1. If the Transporter is registered in RCRAInfo as a Transporter, then only the epaSiteId must be provided. All the site information will be obtained from RCRAInfo. If other site information is provided it will be ignored with warnings.

2.2. If the Transporter is not registered in RCRAInfo, then the following site information must be provided:

- Site Address
- Site Name
- Contact phone

2.3. If Site ID AND Site Information are not provided, the system generates errors:

```
{
        "message": “Value is not provided”
        "field": "Emanifest.transporter.epaSiteId”
}
{
        "message": “Value is not provided”
        "field": "Emanifest.transporter.siteAddress”
}
{
        "message": “Mandatory field is not provided”
        "field": "Emanifest.transporter.name”
}
{
        "message": “Value is not provided”
        "field": "Emanifest.transporter.contact.phone.number”
}
```

2.4. If Site ID is provided AND Site Information is not provided, the system performs the following steps:

2.4.1. Validate Site ID format

2.4.2. If Site ID is not valid, the service generates the following error:

```
{
"message": “Provided Value is not Valid. Does not match format of: Two Letter Activity Location Code + Up to 10 alphanumeric characters”,
"field": "Emanifest.transporter.epaSiteId "
}
```

2.4.3. If the Site ID is valid, the system will search in RCRAInfo by Site ID.

2.4.4. If the site is found, the system obtains Site Information from RCRAInfo and stores it into the Manifest.

2.4.5. If site is not found, the service generates the following error:

```
{
    "message": “Site with the provided EPA Site ID is not registered”,
    "field": "Emanifest.transporter.epaSiteId",
}
```

2.5. If the Site ID is provided AND Site Information is provided, the site information will be ignored with warnings and the system will perform the following steps:

2.5.1. Validate provided Site ID

2.5.2. If Site ID is not valid, the service generates error:

```
{
    "message": “Provided value is not valid. Does not match format of: Two Letter Activity Location Code + Up to 10 alphanumeric characters”,
    "field": "Emanifest.transporter.epaSiteId”
    “value”: “EPA Site ID value”
}
```

2.5.3. If Site ID is valid, check if site is registered

2.5.4. If the site is not registered, the service generates a warning:

```
{
    "message": “Site with provided epa site id is not registered”,
    "field": "Emanifest.transporter.epaSiteId”,
    “value”: “Site ID value”
}
```

2.6. If Site ID is not provided AND Site Information is provided, the system will perform the following steps:

2.6.1. Validate the provided Site Information

2.6.2. Validate the Site Name

2.6.3. If the Site Name is not valid, the service generates the following error:

```
{
    "message": “Provided Value is not Valid”,
    "field": "Emanifest.transporter.name",
    “value”: “site name value”
}
```

2.6.4. Validate the mandatory site address fields:

- address1 (50 Character maximum)
- city (25 Character maximum)
- state.code (2 Character State Abbreviation)
- zip (14 Character maximum)

2.6.5. If any of the mandatory location address fields are not provided or invalid, the service generates the following error:

```
{
    "message": “Value is not Provided/Provided Value is not Valid”,
    "field": "Emanifest.transporter.siteAddress.address1/city/state/zip",
    “value”: “address1/city/state/zip value”
}
```

3. If Emanifest.submissionType is “Hybrid” or “FullElectronic” and Emanifest.status > Scheduled and provided transporter information is different than currently stored the service generates the following warning:
   {
   "message": “Provided Transporter information will be ignored. Transporter Information cannot be updated if the manifest status is InTransit or thereafter”
   "field": "Emanifest.transporter.epaSiteId"
   }

Paper Signature Info Validation

1. If submissionType is “FullElectronic” or “Image” and any (Generator, Transporter(s), designatedFacility) of the paperSignature Information (printedName or signatureDate) is provided, the service generates a warning:
   {
   "message": “Provided Field will be ignored. Paper Signature Information is not applicable for FullElectronic/Image submission type”,
   "field": "Emanifest.generator/transporter/designatedFacility.paperSignatureInfo",
   “value”: “printedName or signatureDate value”
   }
2. If submissionType is “Hybrid” and Emanifest.status > Scheduled and generator.PaperSignature is provided, the service generates the following warning:
   {
   “message”: “Provided value will be ignored. Generator paper signature Date/Printed Name shall not be provided after Scheduled status.”,
   “field”: “emanifest.generatorPaperSignature.signatureDate/printedName”,
   “value”: “ value”
   }

3. If submissionType is “DataImage5Copy”, the following applies:
   3.1. If Generator/Transporter/designatedFacility Printed Names are not provided, the service generates a warning:
   {
   “message”: “Field is not Provided”,
   “field”: “Emanifest.generator/transporter/designatedFacility.paperSignature.printedName”
   }
   3.2. If the Generator/Transporter/designatedFacility Printed Names are provided, validate the format. If an invalid format is provided, the service generates the following error:
   {
   “message”: "String \"{provided printed name value}\" is too long (length: { provided printed name length}, maximum allowed: 255)",

```
      “field”:
“Emanifest.generator/transporter/designatedFacility.paperSignature.printedName”,
      “value”: “value”
}
```

3.3. If the Generator/Transporter/designatedFacility Signature Dates are not provided, the service generates the following error

```
{
      “message”: “Mandatory Field is not Provided”,
      “field”:
“Emanifest.generator/transporter/designatedFacility.paperSignatureInfo.signatureDate”,
}
```

3.4. If Generator/Transporter/TSDF Signature Dates are provided, validate format. If format is invalid the service generates the following error (for this example: TSDF is used)

```
{
      “message”: "String \"{provided signature date}\" is invalid against requested date
format(s) [yyyy-MM-dd'T'HH:mm:ssZ, yyyy-MM-dd'T'HH:mm:ss.SSSZ]",
“field”: “Emanifest.generator/transporter/designatedFacility.paperSignatureInfo.
signatureDate”,
      “value”: “value”
}
```

4. The system will perform the following validation for consecutiveness of Signature Dates for DataImage5Copy:

4.1. If the signature dates for Generator, all Transporters, and designated Facility are provided and have a valid format, and more than one transporter is provided, the system performs the following steps:

4.1.1. Check if Generator signatureDate <= First Transporter signature date (Transporter.order will be used to determine consecutiveness of transporters). If not, the service generates the following error:

```
{
“message”: “Transporter Signature date must be on or after Generator Signature
date”,
“field”: ”Emanifest.generator.paperSignatureInfo.signatureDate”,
“value”: “Generator Signature Date value”
}
{
“message”: “Generator Signature Date must be earlier or the same date as the first
Transporter Signature Date”,
“field”: ”Emanifest.transporter.paperSignatureInfo.signatureDate”,
“value”: “Transporter Signature Date value”
}
```

4.1.2. Check the order of the Transporters signature dates. If there is more than one Transporter provided, check if signature dates of transporters are in order or the same. For example, if there are two transporters and the signature date of transporter which has order = 2 is earlier than signature date of transporter with order = 1, the service generates the following error:

```
{
    "message": "Signature Date of previous transporter must be before this transporter's signature date",
    "field": "Emanifest.transporters[1].paperSignatureInfo.signatureDate",
    "value": " Second Transporter Signature Date value "
}
```

4.1.3. Check if the Designated Facility signature date >= Last Transporter signature date. If not, the service generates the following error:

```
{
    "message": "Signature of designated facility must be on or after the last transporter's signature date",
    "field": "Emanifest.designatedFacility.paperSignatureInfo.signatureDate",
    "value": "Designated Facility Signature Date value "
  }
```

4.1.4. If any of the Signature Dates > current date the service generates the following error:

```
{
“message”: “Generator/Transporter/Designated Facility Signature Date cannot be in the future”,
“field”: ”Emanifest.generator/transporter/designatedFacility.paperSignatureInfo.signatureDate”,
“value”: “Signature Date value”
}
```

4.2. If Emanifest.rejectionInfo.rejectionType = “FullReject” and Emanifest.rejectionInfo.transporterOnsite == true the following applies

4.2.1. If Emanifest.rejectionInfo.alternateDesignatedFacilityType == “Tsdf” the following applies

4.2.1.1. If Emanifest.rejectionInfo.alternateDesignatedFacility.paperSignatureInfo.signatureDate is not provided, the service generates the following error:

```
{
        “message”: “Mandatory field is not provided”,
        “field”: “Emanifest.rejectionInfo.alternateDesignatedFacility.paperSignatureInfo.signatureDate”
}
```

4.2.1.2. If Emanifest.rejectionInfo.alternateDesignatedFacility.paperSignatureInfo.printedName is not provided, the service generates the following error:

```
{
        “message”: “Mandatory field is not provided”,
        “field”:
    “Emanifest.rejectionInfo.alternateDesignatedFacility.paperSignatureInfo.
    printedName”
}
```

4.2.1.3. If Emanifest.rejectionInfo.alternateDesignatedFacility.paperSignatureInfo.signatureDate is invalid the service generates the following error:

```
{
        “message”: "String \"{provided signature date}\" is invalid against
requested date format(s) [yyyy-MM-dd'T'HH:mm:ssZ, yyyy-MM-
dd'T'HH:mm:ss.SSSZ]",
        “field”:
        “Emanifest.rejectionInfo.alternateDesignatedFacility.paperSignatureInfo
        .signatureDate”,
        “value”: “value”
}
```

4.2.1.4. If Emanifest.rejectionInfo.alternateDesignatedFacility.paperSignatureInfo.signatureDate < Emanifest.designatedFacility.paperSignatureInfo.signatureDate then the service generates the following error:

```
{
   "message": "Signature of alternate designated facility must be on or after the
designated facility’s signature date",
   "field": "Emanifest.designatedFacility.paperSignatureInfo.signatureDate",
   "value": "Alternate Designated Facility Signature Date value "
  }
```

4.2.2. If Emanifest.rejectionInfo.alternateDesignatedFacilityType == “Generator” the following applies

4.2.2.1. If Emanifest.rejectionInfo.generator.paperSignatureInfo.signatureDate is not provided, the service generates the following error:

```
{
        “message”: “Mandatory field is not provided”,
        “field”:
    “Emanifest.rejectionInfo.generator.PaperSignatureInfo.signatureDate”
```

}

4.2.2.2. If Emanifest.rejectionInfo.generator.paperSignatureInfo.printedName is not provided, the service generates the following error:

```
{
        “message”: “Mandatory field is not provided”,
        “field”: “Emanifest.rejectionInfo. generator.paperSignatureInfo.
    printedName”
}
```

4.2.2.3. If Emanifest.rejectionInfo.generator.paperSignatureInfo.signatureDate is invalid the service generates the following error:

```
{
        “message”: "String \"{provided signature date}\" is invalid against
requested date format(s) [yyyy-MM-dd'T'HH:mm:ssZ, yyyy-MM-
dd'T'HH:mm:ss.SSSZ]",
        “field”:
        “Emanifest.rejectionInfo.generator.paperSignatureInfo.signatureDate”,
        “value”: “value”
}
```

4.2.2.4. If Emanifest.rejectionInfo.generator.paperSignatureInfo.signatureDate < Emanifest.designatedFacility.paperSignatureInfo.signatureDate then the service generates the following error:

```
{
“message”: “Invalid Value. Generator Signature Date must be the same or later than
Designated Facility Signature Date”,
“field”: ” Emanifest.rejectionInfo. generator.paperSignatureInfo.signatureDate”,
“value”: “value”
}
{
“message”: “Invalid Value. Generator Signature Date must be the same or later than
Designated Facility Signature Date”,
“field”: ”Emanifest.designatedFacility.paperSignatureInfo.signatureDate”,
“value”: “Designated Facility Signature Date value”
}
```

Import Information Validation

1. If submissionType is “FullElectronic” or “Hybrid” and provided Emanifest.status > “Scheduled” the following applies

   1.1. If current Emanifest.import == true and provided Emanifest.import == false then the service generates the following warning:

   {

“message”: “Provided field will be ignored. Emanifest.import cannot be changed after Scheduled status for “FullElectronic” or “Hybrid” submission type”,
“field”: “Emanifest.import”,
“value”: “import value”
}

1.2. If current Emanifest.import == false and provided Emanifest.import == true then the service generates the following warning:
{
“message”: “Provided field will be ignored. Emanifest.import cannot be changed after Scheduled status for “FullElectronic” or “Hybrid” submission type”,
“field”: “Emanifest.import”,
“value”: “import value”
}

1.3. If current Emanifest.import == falseand Emanifest.importInfo.importGenerator fields are provided then the service generates the following warning:
{
“message”: “Provided field will be ignored. Emanifest.importInfo.importGenerator fields cannot be changed after Scheduled status”,
“field”: “Emanifest.importInfo.importGenerator.{specific field}”,
“value”: “field value”
}

2. If Emanifest.import is not provided, the service generates the following error:
{
“message”: “Mandatory field is not provided”,
“field”: “Emanifest.import”
}

3. If Emanifest.import is true the following applies

2.1. If any of the following ImportInfo fields are not provided:

- ImportGenerator.name
- ImportGenerator.city
- ImportGenerator.address
- ImportGenerator.country.code
- ImportPortInfo.city
- ImportPortInfo.state.code

then the service generates the following error:
{
“message”: “Mandatory Field is not Provided.”,
“field”:
“Emanifest.importInfo.ImportGenerator.name/city/address/country/ImportPortInfo.city/ImportPortInfo.state”
}

2.2. If any of the following ImportInfo fields are not provided:

- ImportGenerator.postalCode
- ImportGenerator.province

then the service generates the following warning:

```
{
    “message”: “Field is not Provided.”,
    “field”: “Emanifest.importInfo.ImportGenerator.postalCode/province”
}
```

2.3. If for any Waste the Waste.additionalInfo.consentNumber is not provided then the service generates the following warning:

```
{
“message”: “Field is not Provided.”,
“field”: “Emanifest.Waste.additionalInfo.consentNumber”
}
```

2.4. If Waste.additionalInfo.consentNumber is invalid then the service generates the following warning:

```
{
“message”: “Invalid Value Provided. Expected format 12 digits. 6 digits "I" and then 5 digits.”,
“field”: “Emanifest.additionalInfo.consentNumber”
“value”: “consentNumber value”
}
```

2.5. If ImportGenerator.name is invalid then the service generates the following error:

```
{
“message”: “Invalid Value Provided. Expected length not to exceed 80 characters”,
“field”: “Emanifest.importInfo.ImportGenerator.name”
“value”: “name value”
}
```

2.6. If ImportGenerator.address is invalid then the service generates the following error:

```
{
“message”: “Invalid Value Provided. Expected length not to exceed 50 characters”,
“field”: “Emanifest.importInfo.ImportGenerator.address”
“value”: “address value”
}
```

2.7. If ImportGenerator.city is invalid then the service generates the following error:

```
{
“message”: “Invalid Value Provided. Expected length not to exceed 25 characters”,
“field”: “Emanifest.importInfo.ImportGenerator.city”
“value”: “city value”
}
```

2.8. If ImportGenerator.province is invalid then the service generates the following warning:

```
{
“message”: “Invalid Value Provided. Expected length not to exceed 50 characters”,
```

“field”: “Emanifest.importInfo.ImportGenerator.province”
“value”: “province value”
}

2.9. If ImportGenerator.postalCode is invalid then the service generates the following warning:

{
“message”: “Invalid Value Provided. Expected length not to exceed 50 characters”,
“field”: “Emanifest.importInfo.ImportGenerator.postalCode”
“value”: “province value”
}

2.10. If ImportGenerator.country.Code is invalid then the service generates the following warning:

{
“message”: “Provided Value was not Found”,
“field”: “Emanifest.importInfo.ImportGenerator.country.code”
“value”: “code value”
}

2.11. If portOfEntry.cityPort is invalid then the service generates the following warning:

{
“message”: “Invalid Value Provided. Expected length not to exceed 25 characters”,
“field”: “Emanifest.importInfo. portOfEntry.cityPort”
“value”: “portOfEntry.cityPort value”
}

2.12. If portOfEntry.state.code is invalid then the service generates the following warning:

{
“message”: “Provided Value was not Found”,
“field”: “Emanifest.importInfo. portOfEntry.state.code”
“value”: “code value”
}

3. If Emanifest.import == false and any of the following fields are provided:

- ImportGenerator.name
- ImportGenerator.city
- ImportGenerator.address
- ImportGenerator.country.code
- ImportGenerator.postalCode
- ImportGenerator.province
- portOfEntry.cityPort
- portOfEntry.state.code
- Waste.additinonalInfo.consentNumber

Then service generates the following warning:

{
“message”: “Provided Value will be ignored. The field is not applicable if Manifest is not imported”

```
“field”: “{provided field}”,
“value”: “field value"
}
```

New Manifest Information Validation

1. If the New Manifest is created for shipping the waste to another TSDF or back to the Generator as a “result” of the rejection or residue of the Original Manifest then the following applies

1.1. If Emanifest.status >= “ReadyForSignature” and Emanifest.containsPreviousRejectOrResidue is not provided then the service generates the following error:

```
{
    “message”: ”Mandatory Field is not Provided.”,
    “field”: “Emanifest.containsPreviousRejectOrResidue”
}
```

1.2. If Emanifest.containsPreviousRejectOrResidue == true and Emanifest.additionalInfo.newManifestDestination is not provided then the service generates the following error:

```
{
    “message”: ”Mandatory Field is not Provided.”,
    “field”: “Emanifest.additionalInfo.newManifestDestination”
}
```

1.3. If Emanifest.containsPreviousRejectOrResidue == true and Emanifest.additionalInfo.originalManifestTrackingNumbers are not valid the service generates the following warning:

```
{
    “message”: ”Invalid value provided. Provided Field will be ignored.”,
    “field”: “Emanifest.additionalInfo.originalManifestTrackingNumbers”,
    “value”: “provided value”
}
```

1.4. If Emanifest.additionalInfo.newManifestDestination == “OriginalGenerator” and Emanifest.rejection == true the service generates the following errors:

```
{
“message”: “New Manifest cannot be rejected if shipped back to the original Generator”,
“field”: “Emanifest.additionalInfo.newManifestDestination”,
“value”: “newManifestDestination value”
}
{
“message”: “New Manifest cannot be rejected if shipped back to the original Generator”,
“field”: “Emanifest.rejection”,
“value”: “rejection value”
}
```

Rejection Information Validation

1. If an Original Manifest is rejected then Emanifest.rejection shall be specified as “true”
2. Emanifest.rejection and RejectionInfo shall be provided at the “ReadyForSignature” status. If provided for earlier statuses these fields will be ignored.

   2.1. If Emanifest.status < “ReadyForSignature” and either Emanifest.rejection or RejectionInfo fields are provided then the service generates the following warning:

```
{
    “message”: ”Provided Field will be ignored. Rejection information shall be provided at
    ReadyForSignature status”,
    “field”: “Emanifest.rejection/Emanifest.rejectionInfo”,
    “value”: “provided value”
}
```

3. If submissionStatus >= “ReadyForSignature” and Emanifest.rejection is not provided then the service generates the following error:

```
{
    “message”: ”Mandatory Field is not Provided.”,
    “field”: “Emanifest.rejection”
}
```

4. Emanifest.containsPreviousRejectOrResidue is not provided then the service generates the following error:

```
{
    “message”: ”Mandatory Field is not Provided.”,
    “field”: “Emanifest.containsPreviousRejectOrResidue”
}
```

5. If Emanifest.rejection == true then the following applies

   5.1. If RejectionInfo.rejectionComments is not provided then the service generates the following error:

```
{
    “message”: ”Mandatory Field is not Provided.”,
    “field”: “Emanifest.RejectionInfo. rejectionComments”
}
```

   5.2. If RejectionInfo.rejectionComments are not valid then the service generates the following error:

```
{
    “message”: "String \"{provided rejection comments value}\" is too long (length: { provided
    rejection comments length}, maximum allowed: 255)",
    “field”: “Emanifest.RejectionInfo.rejectionComments”
    “vale”: “rejectionComments value”
}
```

   5.3. If RejectionInfo.transporterOnSite is not provided then the service generates the following error:

```
{
    “message”: ”Mandatory Field is not Provided.”,
    “field”: “Emanifest.RejectionInfo.transporterOnSite”
}
```

5.4. If Emanifest.submissionType == “DataImage5Copy” and RejectionInfo.transporterOnSite == true the following applies

5.4.1. If RejectionInfo.rejectionType == “PartialReject” then the service sets RejectionInfo.rejectionType = “FullReject” and generates the following warning:

```
{
  “message”: ”Provided value will be Ignored. If Transporter is On Site rejection type must be “FullReject”,
  “field”: “Emanifest.RejectionInfo.rejectionType”,
  “value”: “rejectionType value”
}
```

5.4.2. e-Manifest sets rejectionInfo.rejectionType as “FullReject” in the e-Manifest database

5.4.3. If RejectionInfo.newManifestTrackingNumbers is provided then then the service generates the following warning:

```
{
  “message”: ”Provided Field will be Ignored. New Manifest(s) are not created if Transporter is On Site”,
  “field”: “Emanifest.RejectionInfo.newManifestTrackingNumbers”
}
```

5.4.4. If RejectionInfo.alternateDesignatedFacilityType is not provided then then the service generates the following error:

```
{
  “message”: ”Mandatory Field is not Provided.”,
  “field”: “Emanifest.RejectionInfo.alternateDesignatedFacilityType”
}
```

5.4.5. If RejectionInfo.alternateDesignatedFacilityType == “Generator then the following applies

5.4.5.1. If generatorPaperSignatureInfo.printedName is not provided then the service generates the following warning:

```
{
"message": "Mandatory Field is not Provided.",
"field": "Emanifest.RejectionInfo.generatorPaperSignature.printedName "
}
```

5.4.5.2. If generatorPaperSignature.printedName has invalid format then the service generates the following warning:

```
{
```

```
"message": "String \"{provided printed name value}\" is too long (length: { provided printed name length}, maximum allowed: 255)",
"field": "Emanifest.RejectionInfo.generatorPaperSignature.printedName ",
"value": "printedName value"
}
```

5.4.5.3. If generatorPaperSignature.signatureDate is not provided then then the service generates the following error:

```
{
"message": "Mandatory Field is not Provided.",
"field": "Emanifest.RejectionInfo.generatorPaperSignature.signatureDate "
}
```

5.4.5.4. If generatorPaperSignature.signatureDate < Emanifest.designatedFacility.paperSignature.signatureDate then the service generates the following error:

```
{
"message": "Invalid Value Provided. Generator signature date (signing as alternate Facility) shall be on or later than TSDF signature",
"field": "Emanifest.RejectionInfo.generatorPaperSignature.signatureDate ",
"value": "signatureDate value"
}
```

5.4.5.5. If generatorPaperSignature.signatureDate > current Date then the service generates the following error:

```
{
"message": "Invalid Value Provided. Generator signature date (signing as alternate Facility) cannot be in future",
"field": "Emanifest.RejectionInfo.generatorPaperSignature.signatureDate ",
"value": "signatureDate value"
}
```

5.4.6. If RejectionInfo.alternateDesignatedFacilityType == “Tsdf” the following applies

5.4.6.1. If RejectionInfo.alternateDesignatedFacility.epaSiteId is not provided then the service generates the following error:

```
{
"message": "Mandatory Field is not Provided.",
"field": "Emanifest.RejectionInfo. alternateDesignatedFacility.epaSiteId"
}
```

5.4.6.2. If RejectionInfo.alternateDesignatedFacility.epaSiteId has invalid format then the service generates the following error:

```
{
"message": "Provided Value is not Valid. Does not match format of: Two Letter Activity Location Code + Up to 10 alphanumeric characters “
"field": "Emanifest.RejectionInfo.alternateDesignatedFacility.epaSiteId",
“value”: “epaSiteId value”
```

}

5.4.6.3. If epaSiteId is valid, the system will search RCRAInfo by Site ID. If the Site ID is found the system will obtain the Site information from RCRAInfo and store it into the Manifest.

5.4.6.4. If the epaSiteId is not found, the service generates the following error:

```
{
"message": “Provided Value is not Found”,
"field"Emanifest.RejectionInfo.alternateDesignatedFacility.epaSiteId",
“value”: “epaSiteId value”
}
```

5.5. If Emanifest.submissionType == “FullElectronic” or “Hybrid” and provided RejectionInfo.transporterOnSite == true then the service generates the following warning:

```
{
  “message”: “Provided Value will be ignored”,
  “field”: “Emanifest.rejectionInfo.transporterOnSite”,
  “value”: “transporterOnSite value”
  }
```

The service will assign rejectionInfo.transporterOnSite = false

5.6. If RejectionInfo.transporterOnSite == false the following applies

5.6.1. If RejectionInfo.rejectionType is not provided then then the service generates the following error:

```
{
  “message”: ”Mandatory Field is not Provided.”,
  “field”: “Emanifest.RejectionInfo.rejectionType”
}
```

5.6.2. If RejectionInfo.newManifestTrackingNumber(s) is provided and not valid the service generates the following error:

```
{
        "message": "Invalid Field Format. 9 digits followed by 3 upper case letters is
expected ",
        "field": "Emanifest.RejectionInfo.newManifestTrackingNumbers",
          “value”: “newManifestTrackingNumber value”
}
```

5.6.3. If manifestTrackingNumber has a valid format, the service checks if the provided manifestTrackingNumber suffix is valid. If the suffix is invalid the service generates the following error:

```
{
        "message": "Invalid Manifest Tracking Number Suffix is Provided",
        "field": "Emanifest.RejectionInfo.newManifestTrackingNumber ",
      “value”: “manifestTrackingNumber value”
}
```

5.6.4. If any of the following fields are provided:

- RejectionInfo.alternateDesignatedFacilityType
- RejectionInfo.generatorPaperSignature
- RejectionInfo.alternateDesignatedFacility

the service generates the following warning:

{
“message”: ”Provided Field will be Ignored. alternateDesignatedFacilityType/ generatorPaperSignature/alternateDesignatedFacility not applicable if Transporter is Off Site”,
“field”: “Emanifest.RejectionInfo. alternateDesignatedFacilityType/ generatorPaperSignature/ alternateDesignatedFacility”,
“value”: “alternateDesignatedFacilityType/generatorPaperSignature/ alternateDesignatedFacility value”
}

6. If Emanifest.rejection == false and If any of the following RejectionInfo fields are provided:
   - RejectionInfo.rejectionType
   - RejectionInfo.transporterOnSite
   - RejectionInfo.alternateDesignatedFacilityType
   - RejectionInfo.generatorPaperSignature
   - RejectionInfo.alternateDesignatedFacility

   the service generates the following warning:

   {
   “message”: ”Provided Field will be Ignored. {Field} not applicable if Manifest is not Rejected”,
   “field”: “Emanifest.RejectionInfo. rejectionType /transporterOnSite/ alternateDesignatedFacilityType/generatorPaperSignature/alternateDesignatedFacility”,
   “value”: “rejectionType/transporterOnSite/alternateDesignatedFacilityType/ generatorPaperSignature/alternateDesignatedFacility value”
   }

Discrepancy fields validation

1. If status >= ReadyForSignature and DiscrepancyResidueInfo.wasteQuantity is not provided then the service generates the following error:

   {
   “message”: ”Mandatory Field is not Provided.”,
   “field”: “Emanifest.discrepancyResidueInfo.wasteQuantity”
   }
2. If status >= ReadyForSignature and DiscrepancyResidueInfo.wasteType is not provided then the service generates the following error:

   {
   “message”: ”Mandatory Field is not Provided.”,
   “field”: “Emanifest.discrepancyResidueInfo.wasteType”
   }

3. If DiscrepancyResidueInfo.wasteQuantity == true or DiscrepancyResidueInfo.wasteType == true and DiscrepancyResidueInfo.discrepancyComments is not provided then the service generates the following error:
   ```
   {
       “message”: ”Mandatory Field is not Provided.”,
       “field”: “Emanifest.discrepancyResidueInfo.discrepancyComments”
   }
   ```
4. If DiscrepancyResidueInfo.discrepancyComments has invalid format then the service generates the following error:
   ```
   {
       “message”:  "String \"{provided discrepancyComments value}\" is too long (length: {
       provided discrepancyComments value length}, maximum allowed: 255)",
       “field”: “Emanifest.discrepancyResidueInfo.discrepancyComments”
       “value”: “discrepancyComments value”

   }
   ```
5. If DiscrepancyResidueInfo.wasteQuantity == false and DiscrepancyResidueInfo.wasteType == false and DiscrepancyResidueInfo.discrepancyComments is provided then the service generates the following warning:
   ```
   {
       “message”: ”Provided Field will be Ignored.”,
       “field”: “Emanifest.discrepancyResidueInfo.discrepancyComments”
       “value”: “discrepancyComments value”
   }
   ```
6. If DiscrepancyResidueInfo.wasteQuantity == true or DiscrepancyResidueInfo.wasteType == true then the service sets Emanifest.discrepancy = true
7. If DiscrepancyResidueInfo.wasteQuantity == false and DiscrepancyResidueInfo.wasteType == false then the service sets Emanifest.discrepancy = false

Residue Fields Validation

1. If status >= ReadyForSignature and DiscrepancyResidueInfo.residue is not provided then the service generates the following error:
   ```
   {
       “message”: ”Mandatory Field is not Provided.”,
       “field”: “Emanifest.discrepancyResidueInfo.residue”
   }
   ```
2. If DiscrepancyResidueInfo.residue == true and DiscrepancyResidueInfo.residueComments is provided, the service validates if DiscrepancyResidueInfo.residueComments has valid format. If DiscrepancyResidueInfo.residueComments has invalid format the service generates the following error:
   ```
   {
   ```

```
    “message”: "String \"{provided residueComments value}\" is too long (length: {provided
    residueComments value length}, maximum allowed: 255)",
    “field”: “Emanifest.discrepancyResidueInfo.residueComments”
    “value”: “residueComments value”
}
```

3. If DiscrepancyResidueInfo.residue == false and DiscrepancyResidueInfo.residueComments is provided then the service generates the following warning:

```
{
    “message”: ”Provided Field will be Ignored.”,
    “field”: “Emanifest.discrepancyResidueInfo.residueComments”
    “value”: “residueComments value”
}
```

4. If DiscrepancyResidueInfo.residue == true then the service sets Emanifest.residue = true
5. If DiscrepancyResidueInfo.residue == true then the Emanifest. residueNewManifestTrackingNumber(s) can be provided.
    5.1. If no Emanifest.residueNewManifestTrackingNumber(s) is provided, the service generates the following warning:

```
{
“message”: ”One of the Waste has residue. Manifest Tracking Number(s) for shipping waste
to another Facility(ies) is not provided”,
“field”: “Emanifest.residueNewManifestTrackingNumber”
“value”: “residueNewManifestTrackingNumber value”
}
```

    5.2. If Emanifest.residueNewManifestTrackingNumber(s) are provided then the service validates the format of residueNewManifestTrackingNumber(s). If it has an invalid format, the service generates the folllowing warning:

```
{
      "message": "Field will be ignored. Invalid Field Format. 9 digits followed by 3 upper
case letters is expected",
      "field": "Emanifest.residueNewManifestTrackingNumbers ",
    “value”: “residueNewManifestTrackingNumbers value”
}
```

    5.3. If the residueNewManifestTrackingNumber has a valid format, the service checks if the provided residueNewManifestTrackingNumber suffix is valid. If the suffix is invalid, the service generates the following warning:

```
{
    "message": "Field will be ignored. Invalid Manifest Tracking Number Suffix is Provided",
    "field": "Emanifest. residueNewManifestTrackingNumber",
      “value”: “residueNewManifestTrackingNumber value”
}
```

6. If DiscrepancyResidueInfo.residue == false then the following applies
    6.1. The service sets Emanifest.residue = false

6.2. If at least one Emanifest.residueNewManifestTrackingNumber is provided, the service generates the following warning:

```
{
      "message": "Field will be ignored. Manifest has no Residue. New Manifest Tracking Number(s) shall not be provided ",
      "field": "Emanifest.residueNewManifestTrackingNumbers ",
    “value”: “residueNewManifestTrackingNumbers value”
}
```

Manifest Additional Information Validation

1. handlingInstructions Validation.

1.1. If handlingInstructions are invalid, the service generates the following warning:

```
{
“message”: “Invalid Value. Text can be no longer than 4000 characters”,
“field”: ”Emanifest.additionalInfo.handlingInstructions”,
“value”: “handlingInstructions value”
}
```

2. originalManifestTrackingNumber Validation

2.1. If Original Manifest Tracking Number is invalid, the service generates the following warning:

```
{
“message”: “Invalid Format. Tracking Number must be 9 digits followed by 3 upper case letters”,
“field”: ”Emanifest.additionalInfo.originalManifestTrackingNumber”,
“value”: “Original Manifest Tracking Number value”
}
```

3. comments Validation

3.1. handlerId validation

3.1.1. If the handler Id is not provided, then the comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. When comment is provided all fields must have a value”,
“field”: ”Emanifest.additionalInfo.comments.handlerId”
}
```

3.1.2. If the handlerId has an invalid format provided, then the comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored, due to an invalid Site ID format being submitted”,
“field”: ”Emanifest.additionalInfo.comments.handlerId”
“value”: “Handler ID value”
}
```

3.1.3. If the provided handlerId does not match the Generator or TSDF or any of Transporter Site IDs in the manifest, the comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. Provided handlerId does not match Generator or TSDF or any of Transporter Site IDs”,
“field”: ”Emanifest.additionalInfo.comments.handlerId”
“value”: “Handler ID value”
}
```

3.2. label validation

3.2.1. If the label element is not provided, then comment will be ignored with and generate the following warning:

```
{
“message”: “Comment will be ignored. Field is not Provided”,
“field”: ”Emanifest.additionalInfo.comments.label”
}
```

3.2.2. If the label element has an invalid format provided, then comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. Invalid Value. Text can be no longer than 250 characters.”,
“field”: ”Emanifest.additionalInfo.comments.label”
“value”: “Label value”
}
```

3.3. description validation

3.3.1. If the description element is not provided, then the comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. Field is not Provided”,
“field”: ”Emanifest.additionalInfo.comments.description”
}
```

3.3.2. If the description element has an invalid format, the provided comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. Invalid Value. Text can be no longer than 250 characters.”,
“field”: ”Emanifest.additionalInfo.comments.description”
“value”: “Description value”
}
```

4. newManifestDestination validation.

4.1. If Emanifest.containsPreviousRejectOrResidue == true and newManifestDestination is not provided then the service generates the following error:

```
{
“message”: “Mandatory Field is not Provided. If the Manifest contains previous Rejection or Residue then newManifestDestination shall be provided”,
“field”: ”Emanifest.additionalInfo. newManifestDestination”
}
```

4.2. If Emanifest.containsPreviousRejectOrResidue == false and newManifestDestination is provided then the service generates the following warning:

```
{
“message”: “newManifestDestination will be ignored. If the Manifest does not contain previous Rejection or Residue then newManifestDestination shall not be provided”,
“field”: ”Emanifest.additionalInfo. newManifestDestination”,
“value” : “newManifestDestination value”
}
```

5. If consentNumber is provided then the service generates the following warning:

```
    {
    “message”: “consentNumber will be ignored. It shall be provided only for the Waste additionalInfo”,
    “field”: ”Emanifest.additionalInfo.consentNumber”,
    “value” : “consentNumber value”
}
```

Waste Validation

1. If the waste element is not provided for any status >= Scheduled, then the service generates the following error:

```
{
    “message”: “Mandatory Field is not Provided.”,
    “field”: “Emanifest.waste”,
}
```

2. If the waste element is provided, the system will perform the following steps:

2.1. If waste.dotHazardous is not provided, the service generates the following error:

```
{
    “message”: "Object has missing required properties ([\"dotHazardous\"])",
    “field”: “Emanifest.wastes.{i}.dotHazardous”
}
```

2.2. If waste.epaWaste is not provided, the service generates the following error:

```
{
    “message”: "Object has missing required properties ([\"epaWaste \"])",
    “field”: “Emanifest.waste. {i}.epaWaste”
}
```

2.3. If waste.pcb is not provided, the service generates the following error:

```
{
      “message”: "Object has missing required properties ([\"pcb \"])",
      “field”: “Emanifest.waste.{i}.pcb”
}
```

2.4. If waste.pcb is true then the system will validate pcbInfos. See details in PCB Information Validation section

2.5. If waste.br is not provided, the service generates the following error:

```
{
      “message”: "Object has missing required properties ([\"br \"])",
      “field”: “Emanifest.waste.{i}.br”
}
```

2.6. If the waste.br is true, the system will validate brInfo. See details in Biennial Report Information Validation section

2.7. Validate dotInformation/wasteDescription.

2.7.1. If the element dotHazardous is true, the waste must contain dotInformation

2.7.2. If the element dotHazardous is false, the waste must contain the wasteDescription.

2.7.3. If the element dotHazardous is true, and dotInformation is not provided, the service generates the following error:

```
{
      “message”: “Mandatory Field is not Provided.”,
      “field”: “Emanifest.waste.dotInformation”
}
```

2.7.4. If the element dotHazardous is true and dotInformation is provided, the service will validate the dotInformation fields. See section DOT Information fields validation for details

2.7.5. If the element dotHazardous is false and dotInformation is provided, the service generates the following warning:

```
{
      “message”: “For non hazardous Waste Dot Information will be ignored. ”,
      “field”: “Emanifest.waste.dotIformation”
}
```

2.7.6. If the element dotHazardous is false and the element wasteDescription is not provided, then the service generates the following error:

```
{
      “message”: “Mandatory Field is not Provided.”,
      “field”: “Emanifest.waste.wasteDescription”
}
```

2.7.7. If the element dotHazardous is false and the element wasteDescription is provided, then the service validates wasteDescription. If wasteDescription is not valid the service generates the following error:

```
{
        “message”: “Invalid Field format.Value of no longer than 500 characters is
    expected”,
        “field”: “Emanifest.waste.wasteDescription”,
        “value”: “wasteDescription value”
    }
```

2.7.8. If the element dotHazardous is true and the element wasteDescription is provided, the service generates the following warning:

```
{
    “message”: “For hazardous Waste, wasteDescription will be ignored.”,
    “field”: “Emanifest.waste.wasteDescription”
}
```

2.8. Validate Management Method Code. See section Management Method Code Validation for details.

2.9. Validate Waste.consentNumber.

2.9.1. If Emanifest.import is true, Waste.consentNumber can be provided at any status. See details of Waste.consentNumber in section Import Information Validation step 2.3

2.9.2. If Emanifest.import is false, Waste.consentNumber shall not be provided. See details of Waste.consentNumber in section Import Information Validation step 3.0

2.10. Validate DOT Information. See section DOT Information fields validation for details.

2.11. Validate Containers and Quantity. See section Containers and quantity fields validation for details

2.12. Validate Hazardous Waste Codes. See section Hazardous Waste Validation for details

2.13. Validate Special Handling Instructions and Additional Information. See section Special Handling Instructions and Additional Information Validation for details

2.14. Validate Biennial Report Information. See section Biennial Report Information Validation

Management Method Code Validation

1. If submissionType is “DataImage5Copy” the following applies

1.1. If epaWaste = true and Waste.managementMethod.code is not provided, the service generates the following error:

```
{
        "message": “Field is Not Provided”,
        "field": "Emanifest.wastes.managementMethod.code "
}
```

1.2. If the Waste.managementMethod.code is provided validate if code in the lookup. If the provided code is not found service generates the following error:

```
{
        “message”: “Provided Value not Found.”,
        “field”: “Emanifest.wastes.managementMethod.code”,
        “value”: “code value”
}
```

2. If submissionType is “FullElectronic” or “Hybrid” the following applies

    2.1. If Emanifest.status == “Scheduled” or “InTransit” or “ReadyForSignature” the following applies

        2.1.1. If the Waste.managementMethod.code is provided validate if code in the lookup. If the provided code is not found, the service generates a warning:

```
{
        “message”: “Provided Value not Found.”,
        “field”: “Emanifest.wastes.managementMethod.code”,
        “value”: “code value”
}
```

    2.2. If Emanifest.status == “Signed” the following applies

        2.2.1. If the Waste.managementMethod.code are provided then the service updates Emanifest.validationStatus = true

        2.2.2. If the Waste.managementMethod.code is not provided, the service generates the following error:

```
{
        "message": “Field is Not Provided”,
        "field": "Emanifest.wastes.managementMethod.code "
}
```

        2.2.3. If the Waste.managementMethod.code is provided validate if code in the lookup. If provided code is not found service generates the following error:

```
{
        “message”: “Provided Value not Found.”,
        “field”: “Emanifest.wastes.managementMethod.code”,
        “value”: “code value”
}
```

DOT Information Fields Validation

The system will perform the following steps on the DOT Information fields:

1. Validate ID Number

    1.1. There are Proper Shipping Names which do not have corresponding Id Numbers in the DOT specifications. For these Proper Shipping Names the Emanifest requires Id Number value of ”NOID”

    1.2. If dotInformation.IdNumber is not provided, the service generates the following error:

```
{
   "message": “Mandatory Field is Not Provided”,
   "field": "Emanifest.wastes.dotInformation.idNumber ",
}
```

    1.3. If dotInformation.IdNumber is provided and not found in Id Number Lookup, the service generates the following error:

```
{
        "message": “Provided Value is not Found”,
        "field": "Emanifest.wastes.dotInformation.IdNumber"
        “value": "value of Id Number"
    }
```

2. Validate Printed Dot Information.
   2.1. If printedDotInformation is not provided, the service generates the following error:

```
{
    "message": "Mandatory Field is not Provided",
    "field": "Emanifest.wastes.dotInformation.printedDotInformation "
}
```

   2.2. If printedDotInformation exceeds the 500 character limit, the service generates the following error:

```
{
    "message": "Invalid Field Format. Printed Dot Information exceeds the 500 character length",
    "field": "Emanifest.wastes.dotInformation.printedDotInformation ",
    "value": "value of printedDotInformation"
}
```

Containers and quantity fields validation

1. Containers and Quantity fields are mandatory
2. If the quantity entity is not provided, the system generates the following error:

```
{
"message": "Mandatory Field is not Provided",
"field": "Emanifest.waste.quantity"
}
```

3. If any of the Containers and Quantity fields are not provided, the service generates the following error:

```
{
"message": "Mandatory Field is not Provided",
"field":
"Emanifest.waste.quantity.containerNumber/containerType/quantity/quantityUnitOfMeasurement"
}
```

4. If the element containerNumber is provided, the service validates containerNumber format. If format is not valid the service generates the following error:

```
{
“message”: "Numeric instance is greater than the required maximum (maximum: 9999, found:
{provided container number})",
“field”: “Emanifest.wastes.quantity.containerNumber”,
“value”: “value of container number”
}
```

5. If the element containerType code is provided, the service validates the containerType code. If containerType code is not valid the service generates the following error:

   ```
   {
   "message": "Instance value (\"AB\") not found in enum (possible values:
   [\"BA\",\"DT\",\"CF\",\"DW\",\"CM\",\"HG\",\"CW\",\"TC\",\"CY\",\"TP\",\"DF\",\"TT\",\"DM\"
   ])",
   "field": "Emanifest.wastes.quantity.containerType.code",
   "value": "value of container type code "
   }
   ```

6. If the element quantity is provided, the service validates the quantity format. If the format is not valid the service generates the following error:

   ```
   {
   "message": " Numeric instance is greater than the required maximum (maximum: 99999999999,
   found: {provided quantity})”,
   "field": "Emanifest.wastes.quantity.quantity ",
   "value": "value of quantity "
   }
   ```

7. If the quantityUnitOfMeasurement code is provided, the service validates the quantityUnitOfMeasurement code. If the quantityUnitOfMeasurement code is not valid, the service generates the following error:

   ```
   {
   "message": "message": "Instance value (\"{provided code}\") not found in enum (possible
   values: [\"P\",\"T\",\"K\",\"M\",\"G\",\"L\",\"Y\",\"N\"])",
   "field": "Emanifest.wastes.quantity.quantityUnitOfMeasurement.code ",
   "value": "value of container type"
   }
   ```

Hazardous Waste Validation

1. If for every waste in the manifest waste.epaWaste == false and no waste codes are provided then the service does not generate any errors or warnings
2. If the waste.dotHazardous is true and waste.epaWaste is true the following applies:

   2.1. If no federal waste codes, TSDF/Generator State waste codes, or TX waste codes (if applicable) are provided, the service generates the following error:

   ```
   {
   "message": "No Federal waste codes or TSDF or Generator state waste codes are provided. At
   least one Federal or State Waste Code shall be provided for the Waste",
   "field": "Emanifest.wastes.hazardousWaste.federalWasteCodes ",
   }
   ```

   2.2. If all Federal waste codes are invalid, the service generates the following error:

   ```
   {
   ```

"message": "Invalid Federal waste code is provided. Waste codes have a 4 character length, first character a letter F, K, P, or U, last three characters expected to be numeric. At least one valid Federal or State waste code shall be provided for the Waste",
"field": "Emanifest.wastes.hazardousWaste.federalWasteCodes.code ",
“value”: “ federalWasteCode value”
}

2.3. If Generator.siteAddress.state.code is not “TX,” then hazardousWaste.generatorStateWasteCodes can be provided and hazardousWaste.txStateWasteCodes shall be ignored.

2.4. If DesignatedFacility.siteAddress.state.code is not “TX,” then hazardousWaste.tsdfStateWasteCodes can be provided and hazardousWaste.txStateWasteCodes shall be ignored.

2.5. If Generator.siteAddress.state.code is “TX” then hazardousWaste.txStateWasteCodes shall be provided and hazardousWasteCode.generatorStateWasteCodes shall be ignored

2.6. If DesignatedFacility.siteAddress.state.code is “TX” hazardousWaste.txStateWasteCodes shall be provided and hazardousWaste.tsdfStateWasteCodes shall be ignored.

2.7. If TSDF/Generator state waste code(s) are provided and TSDF/Generator.siteAddress.state.code are not “TX” the following applies:

2.7.1. If TSDF/Generator state waste codes are invalid, the service generates the following error(s):
{
"message": "Invalid TSDF/Generator waste code provided. Six-character length, alphanumeric characters expected. At least one Federal or State waste code shall be provided for the Waste",
"field": "Emanifest.wastes.hazardousWaste.tsdf/generatorWasteCode.code",
“value”: “Waste Code value”
}

2.7.2. If TSDF/Generator state waste codes are valid then the service checks if provided state waste code is defined for the state. If the waste code is not defined for the state, the service generates the following error(s):
{
"message": " Invalid TSDF/Generator waste code provided. At least one Federal or State waste code shall be provided for the Waste",
"field": "Emanifest.wastes.hazardousWaste.tsdf/generatorWasteCodes ",
}

2.8. If TSDF/Generator state waste code(s) are provided and either TSDF or Generator.siteAddress.state.code is “TX” and state waste code(s) are invalid then the service generates the following error:
{
"message": "Invalid TSDF/Generator waste code provided. Eight-character length, alphanumeric characters expected for TX. At least one Federal or State waste code shall be provided for the Waste",

```
"field": "Emanifest.wastes.hazardousWaste.tsdf/generatorStateWasteCodes.code",
“value”: “Waste Code value”
}
```

3. If waste.dotHazardous == true and waste.epaWaste == false, at least one state waste code is valid and other state waste codes are invalid or not defined for the state, then the service generates warnings for all invalid state waste codes:

   ```
   {
   "message": "Provided TSDF/Generator Waste Codes will be ignored.",
   "field":
   "Emanifest.wastes.hazardousWaste.tsdfStateWasteCodes/generatorStateWasteCodes.code",
   “value”: “waste code value”
   }
   ```

4. If waste.dotHazardous == false and Federal waste code(s) are provided, the service generates the following warning:

   ```
   {
   "message": "Provided Federal Waste Codes will be ignored.",
   "field": "Emanifest.wastes.hazardousWaste.federalWasteCodes.code",
   “value”: “federal waste code value”
   }
   ```

5. If waste.dotHazardous == false and the waste.epaWaste == true the service generates the following warning:

   ```
   {
   "message": "Provided EPA Waste value will be ignored. If the Waste.dotHazardous is false the waste.epaWaste cannot be true ",
   "field": "Emanifest.wastes.epaWaste ",
   “value”: “epaWaste value”
   }
   ```

6. If Waste.dotHazardous == true and Waste.epaWaste == false and Federal waste code(s) are provided, the service generates the following warning:

   ```
   {
   "message": "Provided Federal Waste Codes will be ignored.",
   "field": "Emanifest.wastes.hazardousWaste.federalWasteCodes.code",
   “value”: “federal waste code value”
   }
   ```

7. If Emanifest.generator.siteAddress.state.code == Emanifest.designatedFacility.siteAddress.state.code and waste.hazardousWaste.tsdfStateWasteCodes and waste.hazardousWaste.generatorStateWasteCodes are provided, the following applies

   7.1. The service will store waste.hazardousWaste.tsdfStateWasteCodes and waste.hazardousWaste.generatorStateWasteCodes under waste.hazardousWaste.generatorStateWasteCodes

   7.2. The service generates the following warning:

```
{
"message": "Provided TSDF Waste Codes will be stored with Generator Waste Codes. If Generator and TSDF are located in the same state then Generator and TSDF waste codes shall be provided under Generator waste codes",
"field": "Emanifest.wastes.hazardousWaste.tsdfWasteCodes"
}
```

Biennial Report Information Validation

1. If the Waste.br is true, then the Waste.brInfo fields are mandatory. If the Waste.brInfo is not provided or empty, then the system generates the following error:
   ```
   {
       "message": "Mandatory Field is not Provided",
       "field": "Emanifest.waste.brInfo"
   }
   ```
2. Validate Source Code
   2.1. If the Source Code is not provided, the service generates the following error:
   ```
   {
   "message": "Mandatory Field is not Provided",
   "field": "Emanifest.waste.brInfo.sourceCode.code"
   }
   ```
   2.2. If Source Code is not found, the service generates the following error
   ```
   {
   "message": "Provided Value not Found",
   "field": "Emanifest.waste.br.sourceCode.code",
   “value”: “code value”
   }
   ```
3. Validate Form Code
   3.1. If Form Code is not provided, the service generates the following error:
   ```
   {
   "message": "Mandatory Field is not Provided",
   "field": "Emanifest.waste.brInfo.formCode.code"
   }
   ```
   3.2. If Form Code is not found the service generates the following error
   ```
   {
   "message": "Provided Value not Found",
   "field": "Emanifest.waste.brInfo.formCode.code",
   “value”: “code value”
   }
   ```
4. Validate Waste Minimization Code
   4.1. If Waste Minimization is not provided, the service generates the following error:
   ```
   {
   ```

```
"message": "Mandatory Field is not Provided",
"field": "Emanifest.waste.brInfo.wasteMinimization.code"
}
```

4.2. If the Waste Minimization is not found the service generates the following error

```
{
"message": "Provided Value not Found",
"field": "Emanifest.waste.brInfo.wasteMinimization.code",
“value”: “code value”
}
```

5. Validate Density and Density Units Of Measurement

5.1. If Waste.UnitOfMeasurement.code is not G or L or N (volumes) then Density and Density Units Of Measurement will be ignored with the following warning:

```
{
"message": "Provided field will be ignored. Density shall be provided only if the quantity units of measurement is the volume (G, L, N)",
"field": "Emanifest.waste.brInfo.density",
“value”: “density value”
}
```

5.2. If Waste.UnitOfMeasurement.code is G or L or N (volumes) and density is not provided, the service generates the following error:

```
{
"message": "Mandatory Field is not Provided",
"field": "Emanifest.waste.brInfo.density"
}
```

5.3. If Waste.UnitOfMeasurement.code is G or L or N (volumes) and density is provided, the service validates the Density format. If Density format is invalid, the service generates the following error

```
{
"message": "Invalid Field Format. Numeric value containing no more than 3 whole digit(s) and 2 decimal digit(s) expected",
"field": "Emanifest.waste.brInfo.density",
“value”: “density value”
}
```

5.4. If Waste.UnitOfMeasurement.code is G or L or N (volumes) and Density Units Of Measurement is not provided, the service generates the following error:

```
{
"message": "Mandatory Field is not Provided",
"field": "Emanifest.waste.brInfo.densityUnitOfMeasurement.code "
}
```

5.5. If Density Units Of Measurement is provided, the service checks if the Density Units Of Measurement is in the lookup format. If the Density Units Of Measurement is not found, the service generates the following error:

```
{
"message": "Provided Value not Found",
"field": "Emanifest.waste.brInfo.densityUnitOfMeasurement.code ",
“value”: “code value”
}
```

6. If br == false and brInfo fields are provided, the following warning will be generated:

```
{
"message": "Provided field will be ignored",
"field": "Emanifest.waste.brInfo.",
“value”: “value”
}
```

PCB Information Validation

1. If pcb == false and any of the pcbInfos entities or fields are provided, the service generates the following warning:

```
{
   “message”: “Provided value will be ignored. PcbInfos provided but pcb flag equals false”
   “field”: “Emanifest.waste.pcbInfos”
   “value”: “ pcbsInfo value”
}
```

2. If pcb == true and pcbInfos are not provided, the service generates the following error:

```
      {
      “message”: “Mandatory Field is not Provided”,
      “field”: “Emanifest.waste.pcbInfos”
}
```

3. If at least one PcbInfos element is provided then service validates pcbInfo fields

3.1. Validate PcbInfos.loadType.code

3.1.1. If PcbInfos.loadType is not provided, the service generates the following error:

```
{
“message”: “Mandatory Field is not Provided”,
“field”: “Emanifest.wastes.pcbInfos.loadType.code”
}
```

3.1.2. If pcbInfos.loadType.code is provided, the service validates it against LoadType lookup. If the field is not valid the service generates the following error:

```
{
“message”: “Invalid Field Format. One of the following values is expected “Container”,
“ArticleInContainer”, “ArticleNotInContainer”, “BulkWaste””,
“field”: “Emanifest.waste.pcbInfos.loadType.code”
“value”: “loadType.code value”
}
```

3.2. Validate pcbInfos.articleContainerId

3.2.1. If pcbInfos.loadType.code is “Container”or “ArticleInContainer” or “ArticleNotInContainer” and PcbInfo.articleContainerId is not provided, the service generates the following error:

```
{
“message”: ”Mandatory Field is not Provided. Fields must be provided if PCB load type is “Container”or “ArticleInContainer” or “ArticleNotInContainer””,
“field”: “Emanifest.waste.pcbInfos. articleContainerId”
}
```

3.2.2. If pcbInfos.articleContainerId is provided, the service checks the field length. If the field is invalid the service generates the following error:

```
{
“message”: “Invalid Field Format. Text no longer than 255 characters is expected”,
“field”: “Emanifest.waste.pcbInfos.articleContainerId”
“value”: “articleContainerId value”
}
```

3.2.3. If pcbInfos.loadType.code is “BulkWaste” and articleContainerId is provided then service generates the following warning:

```
{
“message”: “Field will be ignored”
“field”: “Emanifest.waste.pcbInfos.articleContainerId”
“value”: “articleContainerId value”
}
```

3.3. Validate PcbInfo.dateOfRemoval

3.3.1. If pcbInfos.dateOfRemoval is not provided, the service generates the following error:

```
{
“message”: ”Mandatory Field is not Provided.”,
“field”: “Emanifest.waste.pcbInfos.dateOfRemoval”
}
```

3.3.2. If PcbInfos.dateOfRemoval is provided, the service validates field format. If the field format is invalid the service generates the following error:

```
{
“message”: “Invalid Field Format.”,
“field”: “Emanifest.waste.pcbInfos.dateOfRemoval”
“value”: “dateOfRemoval value”
}
```

3.4. Validate pcbInfos.weight

3.4.1. If pcbInfos.weight is not provided, the service generates the following error:

```
{
```

“message”: ”Mandatory Field is not Provided.”,
“field”: “Emanifest.waste.pcbInfos.weight”
}

3.4.2. If PcbInfos.weight is provided, the service validates field format. If the field format is invalid the service generates the following error:
{
“message”: “Invalid Field Format. Value containing no more than 11 whole digit(s) and 6 decimal digit(s) is expected”,
“field”: “Emanifest.waste.pcbInfo.weight”
“value”: “weight value”
}

3.5. Validate PcbInfos.wasteType

3.5.1. If pcbInfos.loadType is “Container” or “ArticleInContainer” and wasteType is not provided, the service generates the following error:
{
“message”: ”Mandatory Field is not Provided.”,
“field”: “Emanifest.waste.pcbInfos.wasteType”
}

3.5.2. If PcbInfos.loadType is “BulkWaste” and wasteType is provided, the service generates the following warning:
{
“message”: “Field will be ignored”
“field”: “Emanifest.waste.pcbInfos.wasteType”
“value”: “conainerType value”
}

3.5.3. If PcbInfos.wasteType is provided, the service checks the field length. If the field is invalid the service generates the following error:
{
“message”: “Invalid Field Format. Text no longer than 255 characters is expected”,
“field”: “Emanifest.waste.pcbInfos.wasteType”
“value”: “containerType value”
}

3.6. Validate PcbInfos.bulkIdentity

3.6.1. If PcbInfos.loadType is “BulkWaste” and bulkIdentity is not provided, the service generates the following error:
{
“message”: ”Mandatory Field is not Provided.”,
“field”: “Emanifest.waste.pcbInfos.bulkIdentity”
}

3.6.2. If PcbInfos.loadType is “Container” or “ArticleInContainer”and bulkIdentity is provided, the service generates the following warning:
{

```
“message”: “Field will be ignored”
“field”: “Emanifest.waste.pcbInfos.bulkIdentity”
“value”: “bulkIdentity value”
}
```

3.6.3. If PcbInfos.bulkIdentity is provided, the service checks the field length. If the field is invalid the service generates the following error:

```
{
“message”: “Invalid Field Format. Text no longer than 255 characters is expected”,
“field”: “Emanifest.waste.pcbInfos. bulkIdentity”
“value”: “bulkIdentity value”
}
```

Waste Additional Information Validation

1. All Additional Information fields are optional. If any of the fields are invalid, they will be ignored with a warning.
2. handlingInstructions Validation.

2.1. If handlingInstructions is invalid the service generates the following warning:

```
{
“message”: “Invalid Value. Text can be no longer than 4000 characters”,
“field”: ”Emanifest.wastes[i].additionalInfo.handlingInstructions”,
“value”: “handlingInstructions value”
}
```

3. comments Validation

3.1. handlerId validation

3.1.1. If the handler Id is not provided, then the comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. Field is not Provided”,
“field”: ”Emanifest.wastes[i].additionalInfo.comments.handlerId”
}
```

3.1.2. If the handlerId has an invalid format provided, then the comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored, due to an invalid Site ID format being submitted”,
“field”: ”Emanifest.wastes[i].additionalInfo.comments.handlerId”
“value”: “Handler ID value”
}
```

3.1.3. If the provided handlerId does not match the Generator or TSDF or any of Transporter Site IDs in the manifest, the comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. Provided handlerId does not match Generator or TSDF or any of Transporter Site IDs”,
“field”: ”Emanifest.wastes[i].additionalInfo.comments.handlerId”
“value”: “Handler ID value”
}
```

3.2. label validation

3.2.1. If the label element is not provided, then comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. Field is not Provided”,
“field”: ”Emanifest. wastes[i].additionalInfo.comments.label”
}
```

3.2.2. If the label element has an invalid format provided, then comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. Invalid Value. Text can be no longer than 250 characters.”,
“field”: ”Emanifest. wastes[i].additionalInfo.comments.label”
“value”: “Label value”
}
```

3.3. description validation

3.3.1. If the description element is not provided, then the comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. Field is not Provided”,
“field”: ”Emanifest.wastes[i].additionalInfo.comments.description”
}
```

3.3.2. If the description element has an invalid format, the provided comment will be ignored and generate the following warning:

```
{
“message”: “Comment will be ignored. Invalid Value. Text can be no longer than 250 characters.”,
“field”: ”Emanifest. wastes[i].additionalInfo.comments.description”
“value”: “Description value”
}
```

3.4. If originalManifestTrackingNumbers are provided then the service generates the following warning:

```
{
```

“message”: “originalManifestTrackingNumbers will be ignored. This field shall not be provided for the Waste.additionalInfo”,
“field”: ”Emanifest.wastes[i].additionalInfo.originalManifestTrackingNumbers”,
“value”: “originalManifestTrackingNumbers value”
}

3.5. If newManifestDestination is provided then the service generates the following warning:

{
“message”: “newManifestDestination will be ignored. This field shall not be provided for the Waste.additionalInfo”,
“field”: ”Emanifest.wastes[i].additionalInfo. newManifestDestination”,
“value”: “newManifestDestination value”

}

Attachment Validation

1. For the “DataImage5Copy” submission type an attachment contains the scanned copy of the printed manifest form 8700-22 and accompanying 8700-22a’s signed by the Generator, Transporter(s), Designated Facility, and Alternate Designated Facility (if applicable).. This document will be obtained via Get Manifest service and will contain all Manifest data in the format of the 8700-22.
2. For the “Image” submission type an attachment contains the scanned copy of the paper manifest form 8700-22 and accompanying 8700-22a’s signed by the Generator, Transporter(s), Designated Facility, and Alternate Designated Facility (if applicable).
3. The scanned document must be zipped and submitted as a part of the Save and Update Manifest service. The zipped manifest attachment must be passed as a multipart binary content. The attachment may be submitted at the ReadyForSignature status. If the manifest contains an attachment, then attachment metadata elements shall be provided.
4. If the document is already stored as a part of the manifest and document is not provided, the service generates no errors or warnings.
5. If the document is not stored and the status is ReadyForSignature, the document is mandatory.
6. If an attachment is provided, the service checks if the attachment is a zip file (The services only accept one PDF file within the zip file for the printedDocument element). If a non-zipped attachment is provided, the service generates the following error:
{
"message": "Attached document is not compressed(zipped). Service accepts compressed (zip) attachments only",
"field": "Emanifest.printedDocument.name",
“value”: “name of the attached document”
}

7. If a zipped attachment is provided, the service will unzip the document and perform the following steps:
   7.1. If the zip file contains more than one document the service generates the following error
   ```
   {
     "message": "Zip contains more than one document",
     "field": "Emanifest.printedDocument"
   }
   ```
   7.2. If the attachment contained in zip file is not a PDF, the service generates the following error:
   ```
   {
     "message": "Attachment Document is not pdf",
     "field": "Emanifest.printedDocument"
   }
   ```
8. If an attachment is not provided and Emanifest.printedDocument metadata is provided, the service generates a warning:
   ```
   {
       "message": "No attachment provided",
       "field": "Emanifest.printedDocument.name"
   }
   ```
9. If the following attachment metadata is not provided:
   ```
   "printedDocument" : {
      "name" : "name provided by the user.pdf",
      "size" : {pdf document size in bytes},
      "mimeType" : "APPLICATION_PDF"
    }
   ```
   The service generates the following warning:
   ```
   {
     "message": "Attachment Document metadata is not provided",
     "field": "Emanifest.printedDocument"
   }
   ```
10. If one or more metadata fields is not provided, the service generates the following warning:
    ```
    {
      "message": "Attachment Document metadata is not provided",
      "field": "Emanifest.printedDocument.name/size/mimeType"
    }
    ```
11. If metadata is provided, but some of the fields values do not match the zipped content, the service generates the following warning(s):
    ```
    {
      "message": "Attachment Document name/size does not match actual file name/size ",
      "field": "Emanifest.printedDocument.name/size",
     “value”: “Emanifest.printedDocument.name/size value ”
    }
    ```
12. If an incorrect mimeType is provided, the service generates the following warning:

```
{
     "message": "Instance value (\"{provided mimeType}\") not found in enum (possible
values: [\"APPLICATION_PDF\",\"TEXT_HTML\"])",
     "field": "Emanifest.printedDocument.mimeType",
    “value”: “Emanifest.printedDocument.mimeType value”
}
```

Manifest entities and fields validation for Create Correction service

1. If Emanifest.submissionType == “Image” following applies

   1.1. If Emanifest.originType == “Web” or “Service” following applies

      1.1.1. If Emanifest.ppcStatus != “PendingDataEntry” and Emanifest.ppcStatus != “DataQaCompleted” the system will generate following error:

```
{
   “message”: “Manifest version creation is Locked. Manifest is being processed by
   Paper Processing Center”,
   “field”: “Emanifest.ppcStatus”,
   “value”: “ppcStatus value”

}
```

      1.1.2. If Emanifest.originType == “Mail” following applies

         1.1.2.1. If Emanifest.ppcStatus == “PendingDataEntry” the system will generate following error:

```
{
   “message”: “Manifest version creation is Locked. Manifest has not been processed by
   Paper Processing Center”,
   “field”: “Emanifest.ppcStatus”,
   “value”: “ppcStatus value”

}
```

         1.1.2.2. If Emanifest.ppcStatus == “DataEntryInProgress”or “PendingDataQc” or “PendingDataQa” the system will generate following error:

```
{
   “message”: “Manifest version creation is Locked. Manifest is being processed by
   Paper Processing Center”,
   “field”: “Emanifest.ppcStatus”,
   “value”: “ppcStatus value”
}
```

2. Emanifest.manifestTrackingNumber validation is identical to the validation performed in section Manifest entities and fields validation for Update service

2.1. If Emanifest.status is provided, the service generates the following warning:

```
{
```

“message”: “Provided Value will be ignored. Created or updated manifest correction version will be assigned “UnderCorrection” status.”,
“field”: “Emanifest.status”,
“value”: “status value”
}

3. Emanifest.submissionType

3.1. If Emanifest.submissionType is provided and different than the original value the service generates the following warning:
{
“message”: “Provided Submission Type will be ignored. Submission Type cannot be updated.”,
“field”: “Emanifest.submissionType”,
“value”: “submissionType value”
}

3.2. If the service creates a new manifest version, then Emanifest.submissionType will be copied over from the latest version of the manifest

3.3. If the service updates manifest version then the service will perform no actions

4. Emanifest.generator

4.1. If submissionType == “FullElectronic” or “Hybrid” the following applies

4.1.1. If generator is provided then the service generates the following warning:
{
“message”: “Provided Value will be ignored. Created or updated manifest correction version will be assigned current generator.”,
“field”: “Emanifest.generator”
}

4.1.2. If the service creates new manifest version then Emanifest.generator will be copied over from the latest version of the manifest

4.1.3. If the service updates manifest version then service will perform no actions

4.2. If submissionType == “DataImage5Copy” the following applies

4.2.1. If current Emanifest.rejection == true and current Emanifest.rejectionInfo. transporterOnSite == true following applies

4.2.1.1. If generator is provided then the service generates the following warning:
{
“message”: “Provided Value will be ignored. Created or updated manifest correction version will be assigned current generator.”,
“field”: “Emanifest.generator”
}

4.2.1.2. If the service creates new manifest version then Emanifest.generator will be copied over from the latest version of the manifest

4.2.1.3. If the service updates manifest version then service will perform no actions

4.2.2. If current Emanifest.rejection == true and current Emanifest.rejectionInfo. transporterOnSite == false or current Emanifest.rejection == false then generator validation is identical to the validation performed in section Generator Site Id and Site Information Validation

5. Emanifest.designatedFacility
   - 5.1. If submissionType == "FullElectronic" or "Hybrid" the following applies
     - 5.1.1. If designatedFacility is provided then the service generates the following warning:

       {
       "message": "Provided Value will be ignored. Created or updated manifest correction version will be assigned current designatedFacility.",
       "field": "Emanifest.designatedFacility"
       }
     - 5.1.2. If the service creates a new manifest version then Emanifest.designatedFacility will be copied over from the latest version of the manifest
     - 5.1.3. If the service updates the manifest version then service will perform no actions
   - 5.2. If submissionType == "DataImage5Copy" the following applies
     - 5.2.1. If current manifest Emanifest.rejectionInfo.alternateDesignatedFacility != null then designatedFacility validation is identical to the validation performed in section Designated Facility (TSDF)/Generator Site Id and Site Information Validation
   - 5.3. If current manifest Emanifest.rejectionInfo.alternateDesignatedFacility == null the following applies
     - 5.3.1. If designatedFacility is provided then the service generates the following warning:

       {
       "message": "Provided Value will be ignored. Created or updated manifest correction version will be assigned current designatedFacility.",
       "field": "Emanifest.designatedFacility"
       }
     - 5.3.2. If the service creates new manifest version then Emanifest.designatedFacility will be copied over from the latest version of the manifest
     - 5.3.3. If the service updates manifest version then service will perform no actions
6. Emanifest.transporters
   - 6.1. If submissionType == "FullElectronic" or "Hybrid" the following applies
     - 6.1.1. If transporters are provided then the service generates the following warning:

       {
       "message": "Provided Value will be ignored. Created or updated manifest correction version will be assigned current transporters.",
       "field": "Emanifest.transporters"
       }
     - 6.1.2. If the service creates a new manifest version then Emanifest.transporters will be copied over from the latest version of the manifest
     - 6.1.3. If the service updates the manifest version then the service will perform no actions
   - 6.2. If submissionType == "DataImage5Copy" transporters validation is identical to the validation performed in section Transporter Site ID and Site Information Validation
7. Emanifest.waste validation is identical to the validation performed in section Waste Validation

8. Emanifest.generator/designatedFacility/rejectionInfo.alternateDesignatedFacilityType.epaSiteId

8.1. If submissionType == “Image” and Emanifest.generator/ designatedFacility/ rejectionInfo.alternateDesignatedFacility.epaSiteId is different than the original epaSiteId the service generates the following warning:

{
“message”: “Provided Value will be ignored. Created or updated manifest correction version will be assigned current generator/designatedFacility/alternateTsdf.”,
“field”: “Emanifest.generator/ designatedFacility/ rejectionInfo. alternateDesignatedFacilityType.epaSiteId”
}

9. Emanifest.waste validation is identical to the validation performed in section Waste Validation

10. Emanifest.rejection and Emanifest.rejectionInfo

10.1. If submissionType == “FullElectronic” or “Hybrid” the following applies

10.1.1. If Emanifest.rejection is not provided then the following applies

10.1.1.1. If Emanifest.rejectionInfo is provided, the service generates the following warning:

{
“message”: “Provided Value will be ignored. Created or updated manifest correction version will be assigned current rejectionInfo.”,
“field”: “Emanifest.rejectionInfo”
}

10.1.1.2. Emanifest.rejection will be copied over from the latest version of the manifest

10.1.1.3. Emanifest.rejectionInfo will be copied over from the latest version of the manifest

10.1.2. If provided Emanifest.rejection == false and Emanifest.rejectionInfo is provided, the service generates the following warning:

{
“message”: “Provided Value will be ignored.”,
“field”: “Emanifest.rejectionInfo”
}

10.1.3. If provided Emanifest.rejection == true the following applies

10.1.3.1. If rejectionInfo.rejectionType is not provided then the service generates the following error:

{
“message”: “Value is not Provided”,
“field”: “Emanifest.rejectionInfo.rejectionType”
}

10.1.3.2. If provided rejectionInfo.transporterOnSite == true then the service generates the following warning:

{
“message”: “Provided Value will be ignored”,
“field”: “Emanifest.rejectionInfo.transporterOnSite”,
“value”: “transporterOnSite value”

}
The service will assign rejectionInfo.transporterOnSite = false

10.1.3.3. If rejectionInfo.newManifestTrackingNumbers are provided then their validation will be identical to the validation described in section Rejection Information Validation

10.1.3.4. rejectionInfo.rejectionComments validation is identical to the validation described in section Rejection Information Validation

10.1.3.5. If any of the following fields are provided:

- alternateDesignatedFacilityType
- generatorPaperSignature
- generatorElectronicSignature
- alternateDesignatedFacility

the service generates the following warning:

{
“message”: “Provided Value will be ignored”,
“field”: “Emanifest.rejectionInfo.alternateDesignatedFacilityType/ generatorPaperSignature/ generatorElectronicSignature/ alternateDesignatedFacility”,
“value”: “value”
}

10.2. If submissionType == “DataImage5Copy” the following applies

10.2.1. If current Emanifest.rejection == false or current Emanifest.rejection == true and current transporterOnSite == false then the following applies

10.2.1.1. If provided Emanifest.rejection == true and provided transporterOnSite == true then the service generates the following:

{
“message”: “Provided Value will be ignored”,
“field”: “Emanifest.rejectionInfo.transporterOnSite”,
“value”: “transporterOnSite value”
}
The service will assign rejectionInfo.transporterOnSite = false

10.2.1.2. rejectionInfo.rejectionType validation is identical to validations described in section Rejection Information Validation

10.2.1.3. rejectionInfo.newManifestTrackingNumbers validation identical to validations described in section Rejection Information Validation

10.2.1.4. rejectionInfo.rejectionComments validation identical to validations described in section Rejection Information Validation

10.2.1.5. If any of the following fields are provided:

- alternateDesignatedFacilityType
- generatorPaperSignature
- generatorElectronicSignature
- alternateDesignatedFacility

the service generate following warning:

```
{
 “message”: “Provided Value will be ignored”,
 “field”: “Emanifest.rejectionInfo.alternateDesignatedFacilityType/
 generatorPaperSignature/ generatorElectronicSignature/
 alternateDesignatedFacility”,
 “value”: “value”
 }
```

10.2.2. If current Emanifest.rejection == true and current rejectionInfo.transporterOnSite == true and provided Emanifest.rejection == false then following applies

10.2.2.1. The service generates the following warning:

```
{
 “message”: “Provided Value will be ignored”,
 “field”: “Emanifest.rejection”,
 “value”: “ value”
 }
```

10.2.2.2. The service will assign Emanifest.rejection = true

10.2.2.3. rejectionInfo.rejectionComments validation identical to validations described in section Rejection Information Validation

10.2.2.4. If any of the following fields are provided:

- transporterOnSite
- rejectionType
- alternateDesignatedFacilityType
- generatorPaperSignature
- alternateDesignatedFacility

the service generate following warning:

```
{
 “message”: “Provided Value will be ignored”,
 “field”: “Emanifest.rejectionInfo.alternateDesignatedFacilityType/
 generatorPaperSignature/ alternateDesignatedFacility”,
 “value”: “value”
 }
```

10.2.2.5. The following fields will be copied over if new correction is created or won’t be updated if correction is updated:

- transporterOnSite
- rejectionType
- alternateDesignatedFacilityType
- generatorPaperSignature
- alternateDesignatedFacility

11. Emanifest.import and Emanifest.importInfo

11.1. If Emanifest.submissionType == “DataImage5Copy” validation for the Emanifest.import and Emanifest.importInfo is identical to validation described in section Import Information Validation

11.2. If Emanifest.submissionType == “FullElectronic” or “Hybrid” and current Emanifest.import == false the following applies

11.2.1. If Emanifest.import is provided, the service generates the following warning:

{
“message”: “Provided Value will be ignored. Created or updated manifest correction version will be assigned current import.”,
“field”: “Emanifest.import”,
“value”: “import value”
}

11.2.2. If Emanifest.importInfo is provided, the service generates the following warning:

{
“message”: “Provided Value will be ignored. Created or updated manifest correction version will be assigned current importInfo.”
}

11.3. If Emanifest.submissionType == “FullElectronic” or “Hybrid” and current Emanifest.import == true (only portOfEntry can be changed) the following applies

11.3.1. If provided Emanifest.import == false the service generates the following warning:

{
“message”: “Provided Value will be ignored. Created or updated manifest correction version will be assigned current import.”,
“field”: “Emanifest.import”,
“value”: “import value”
}

11.3.2. If any Emanifest.importGenerator fields are provided, the service generates the following warning:

{
“message”: “Provided Value will be ignored. Created or updated manifest correction version will be assigned current importGenerator.<field>.”,
“field”: “Emanifest.importInfo.ImportGenerator.<field>”,
“value”: “field value”
}

11.3.3. Provided Emanifest.importInfo.portOfEntry fields validation is identical to validation in section Import Information Validation

12. Emanifest attachments fields validation is identical to validation in section Attachment Validation
13. Discrepancy fields validation is identical to validation described in section Discrepancy fields validation
14. Residue fields validation is identical to validation described in section Residue Fields Validation
15. New Manifest Information validation is identical to validation described in section New Manifest Information Validation
16. Additional Information validation is identical to validation described in section Manifest Additional Information Validation