# INLAND REVENUE BOARD OF MALAYSIA

# USER GUIDE FOR THE PREPARATION AND ENCRYPTION OF FILES FOR TRANSMISSION THROUGH THE HASIL INTERNATIONAL DATA EXCHANGE FACILITY (HiDEF)

(Revised Version – July 2019)

# Contents

# 1. DATA PREPARATION FOR CbCR XML REPORT

## 1.1 Overview

This section describes how to prepare a CBC data file. Before you begin, you must have a valid certificate from an IRBM approved certificate authority.

## 1.2 Prepare the CbCR XML File

These instructions may change with maintenance updates to the system. HiDEF will only accept files in .zip format. Each archive will contain three files and it will consists of the following files:

- MyCBCID_CBC_Metadata.xml
- MyCBCID_CBC_Payload
- MyCBCID_CBC_Key

| Steps | Process | File Naming Convention |
|---|---|---|
| --- | Obtain a digital certificate from an approved Certificate Authority (CA) | <MyCBCID>_CBC_Cert.crt |
| 1 | Prepare and validate the CbCR XML file Digitally sign the file | <MyCBCID>_CBC_Payload.xml |
| 2 | Compress the CbCR XML file with compatible zip utility | <MyCBCID>_CBC_Payload.zip |
| 3 | Encrypt the CbCR XML file with AES-256 key | <MyCBCID>_CBC_Payload |
| 4 | Encrypt AES key with IRBM public key | <MyCBCID>_CBC_Key |
| 5 | Create sender metadata | <MyCBCID>_CBC_Metadata.xml |
| 6 | Create the transmission file. | <MyCBCID>_<RPTYR>_CBC_<UTC>.zip |
| 7 | Transmit the data packet to HiDEF and receive delivery confirmation | N/A |

## 1.3 MyCBCID

A MyCBCID is created after registration. The ID is a unique 8 character-length number that identifies the transmission. This ID will be included in both HiDEF system alerts and notifications generated by the IRBM.

### 1.3.1 RPTYR

**RPTYR** represents the reporting tax year. It is a 4 character-length number in YYYY format.

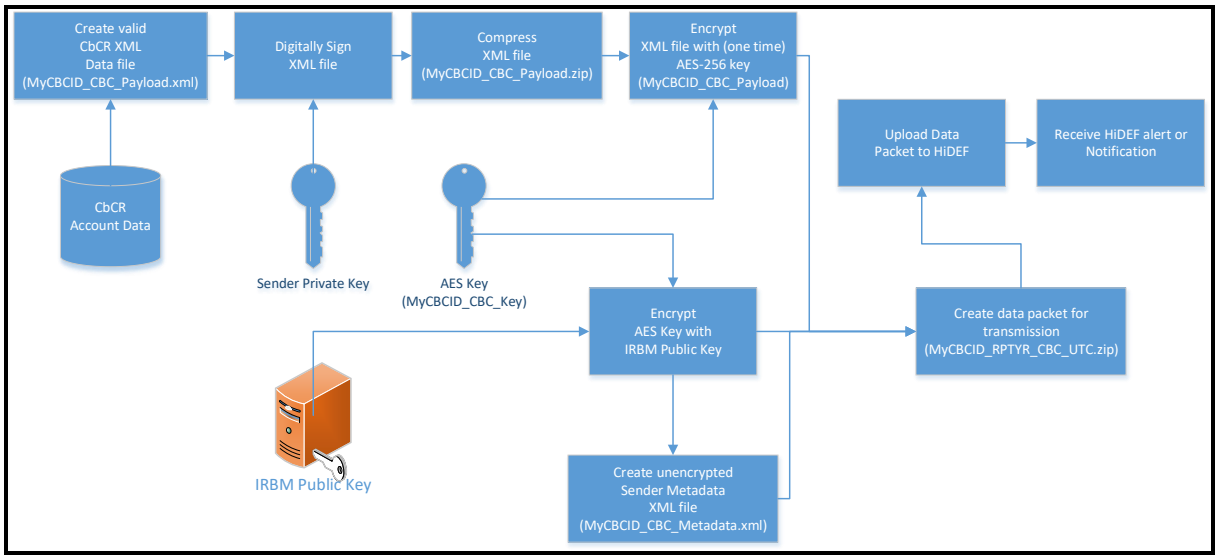
### 1.3.2 UTC

**UTC** represents a timestamp including milliseconds. The timestamp format of the UTC is YYYYMMDDTHHMMSSmsZ where:

- YYYY = 4-digit year
- MM = 2-digit month
- DD = 2-digit day
- HH = 24-hour
- MM = 2-digit minutes
- SS = 2-digit seconds
- ms = 3-digit milliseconds

For example, the timestamp for January 25, 2016 at 16:30:45.321 is represents by 20160125T163045321Z.

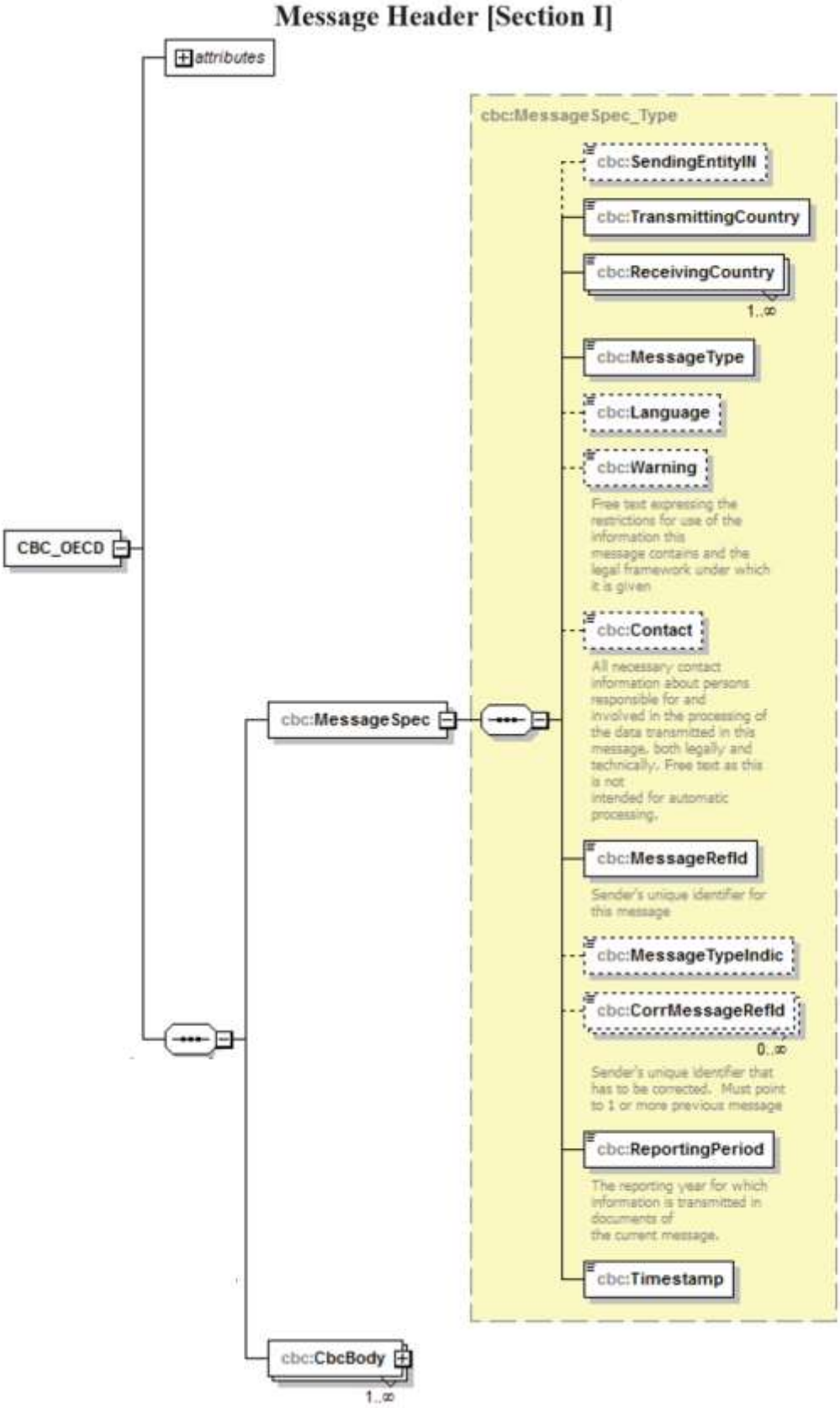
# 2. PROCESS TO PREPARE AND TRANSMIT XML FILE




## STEP 1 - CREATE AND VALIDATE THE CbCR XML FILE

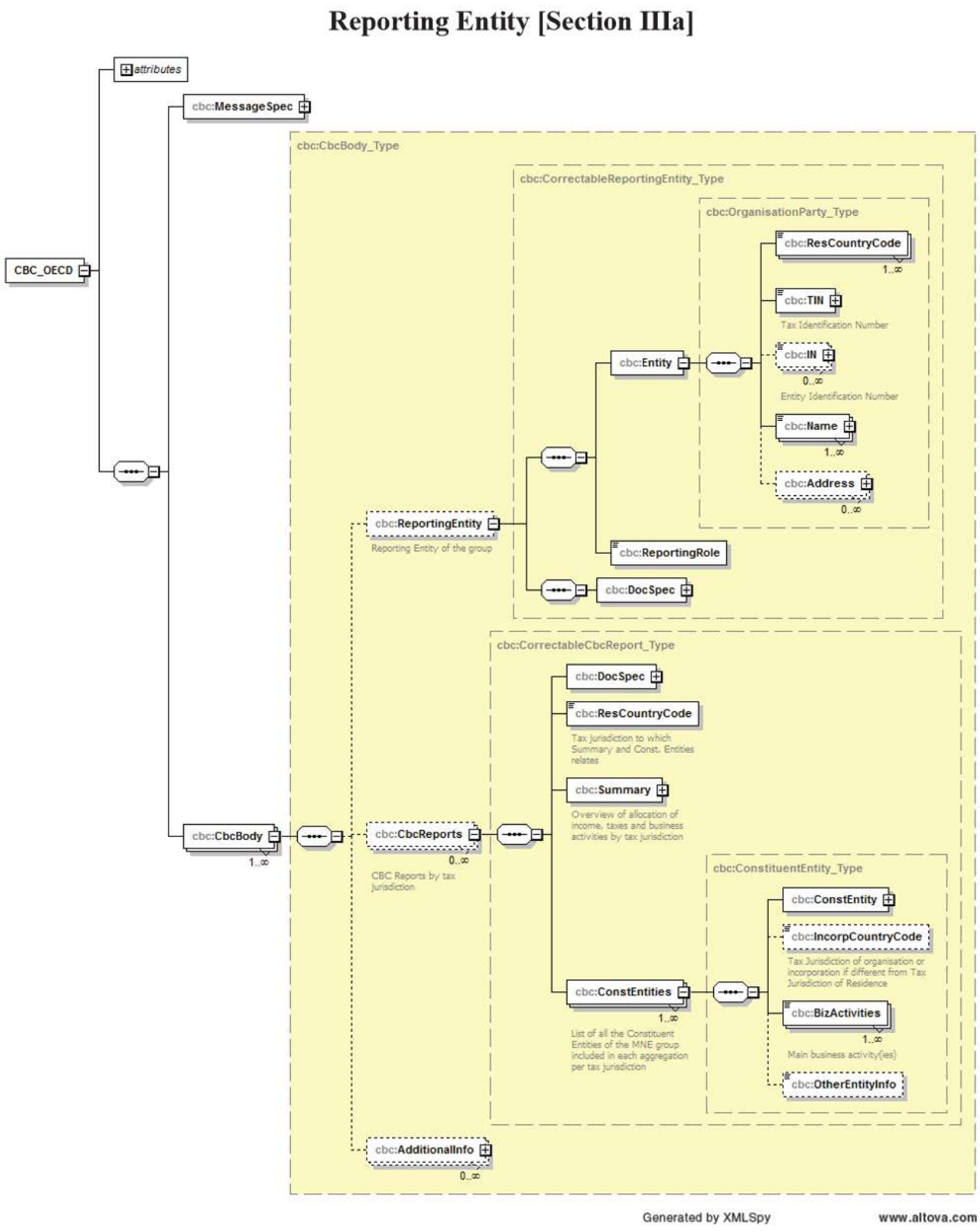
Step 1 explains on how to create a sender payload file. Each CbCR XML file contains information about the accounts required to be reported under CbCR. Ensure that all XML elements have prefixes, do not use default namespaces. For information on the CbCR XML and in relation to CbCR Report, see Country-by-Country Reporting XML Schema and User Guide, Country-by-Country Reporting Status Message XML Schema and User Guide and sample for CbCR Payload XML file, please click here.

## 1a. CbCR Message Header



| Field Name | Datatype | Status | Remarks |
|---|---|---|---|
| **MessageSpec : 1-1** | | | |
| SendingEntityIN | varchar(200) | Validation | **<MYCBCID>**<br><br>A MyCBCID is created after registration. The ID is a unique 8 character-length number. Example 20000001 |
| TransmittingCountry | Varchar(5) | Validation | MY |
| ReceivingCountry | Varchar(5) | Validation | All jurisdictions in which a Constituent Entity is found to be resident on the basis of the information provided by the Reporting Entity in the CbC Report should be entered in this field. |
| MessageType | varchar(20) | Validation | CBC |
| Warning | varchar(4000) | Optional | |
| Contact | varchar(200) | Optional | Authorize Person Name or Contact Person Name |
| MessageRefId | varchar(200) | Validation | Format:<br>**MY<YYYY>-<MYCBCID><YYYYMMDD><SSSS>**<br><br>A MyCBCID is created after registration. The ID is a unique 8 character-length number. Example 20000001<br><br>YYYY – Fiscal Year<br>YYYYMMDD - Date of file creation<br><br>The 4-digit incremental number is one that starts at '0001' and increases to '9999' when a file is produced **on the same day**.<br><br>Example MY2017-200000012018110800001 |
| MessageTypeIndic | varchar(10) | Optional | Possible values:<br>CBC401= The message contains new information<br>CBC402= The message contains corrections for previously sent information. |
| CorrMessageRefId<br>0..∞ | varchar(200) | Optional | This data element is **not used for CbC** reporting. |
| ReportingPeriod | Date | Validation | Format :<br>**<YYYY-MM-DD>**<br><br>The accounting period end date should be entered in this field.<br><br>Example: 2017-12-31 |
| Timestamp | Datetime | Validation | Format :<br>**<YYYY-MM-DD'T'hh:mm:ss>Z**<br><br>Example 2015-03-15T09:45:30Z |

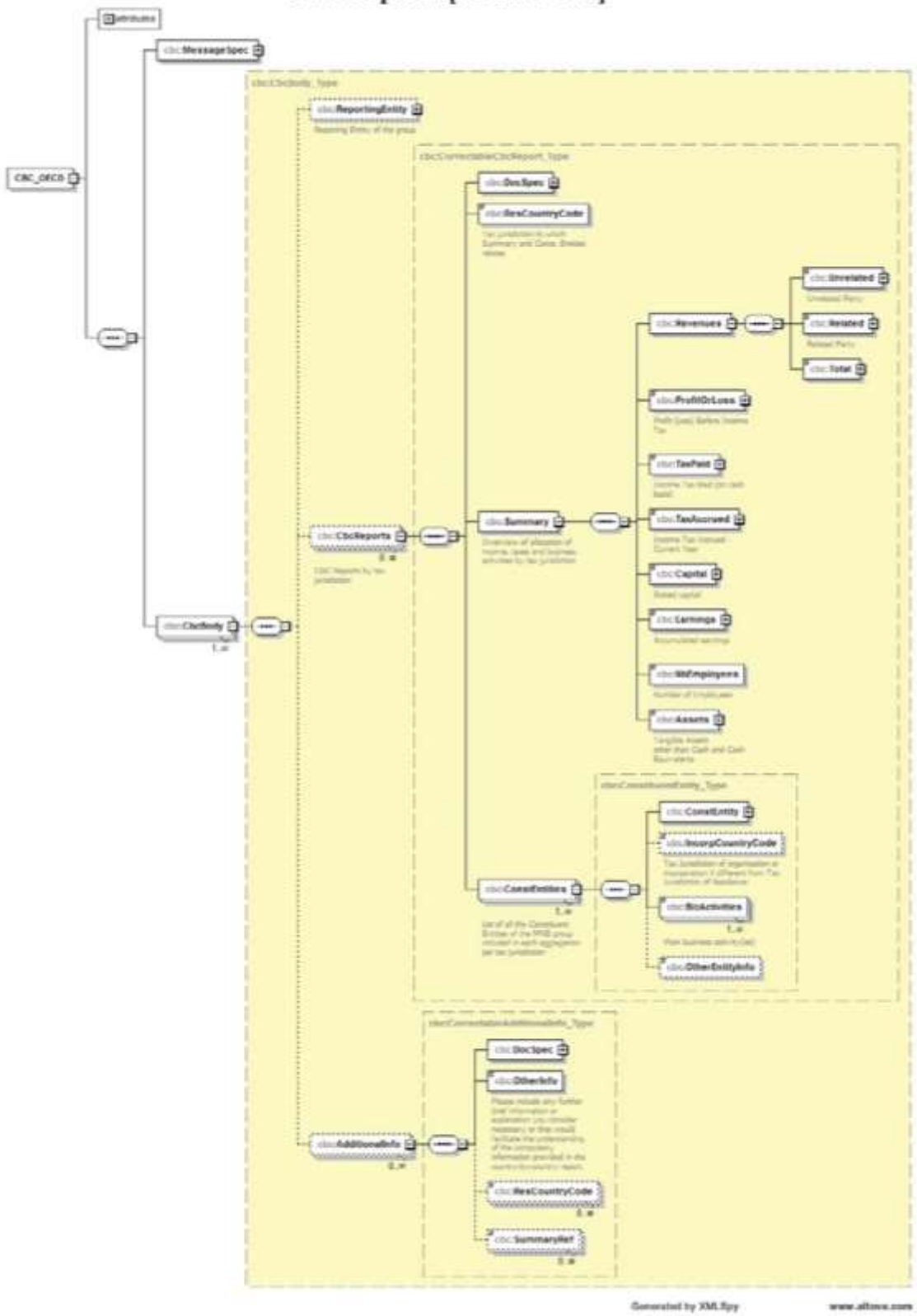
## 1b. CbCR Body : 1..∞



| Field Name | | Datatype | Status | Remarks |
|---|---|---|---|---|
| CbCR Body : 1..∞<br>**ReportingEntitiy : 1-1** | | | | |
| **Entity** | | | | |
| **ResCountryCode** | | 2-Character | | 0..∞<br>Mandatory<br>This data element should contain the country code(s) of the Tax Jurisdiction of the Constituent Entity (or, in case of a permanent establishment that is a Constituent Entity, the jurisdiction in which such permanent establishment is subject to tax). |
| **TIN** | | Min 1 char | | Tax Identification Number |
| | *IssuedBy* | 2-Character | | Country Code |
| **IN : 0..∞** | | Min 1 char | | This data element can be provided (and repeated) if there are other INs available, such as a company registration number or a Global Entity Identification Number (EIN). |
| | *IssuedBy* | 2-Character | | 0..∞<br>Validation<br>If an IN is provided, the issuing jurisdiction **must be provided**. |
| | *INType* | | | Optional<br>Identification Number Type<br>e.g. EIN, TIN |
| **Name : 1..∞** | NameOrganisation | Varchar | | 1..∞<br>Name of Organisation |
| **Address : 1..∞** | *legalAddressType* | 7-Character | | This is a datatype for an attribute to an address. It serves to indicate the legal character of that address (residential, business, etc.)<br>The possible values are:<br>OECD301=residentialOrBusiness<br>OECD302=residential<br>OECD303=business<br>OECD304=registeredOffice<br>OECD305=unspecified |
| | CountryCode | Varchar | | This data element provides the country code associated with the Constituent Entity. |
| | AddressFree | Varchar | | This data element allows input of address information in free text. If the data is entered in 'AddressFree', all available address details shall be presented as one string of bytes, with blanks, slashes or carriage returns being used as a delimiter between parts of the address. This |
| | AddressFix | Varchar | Street | |
| | | Varchar | BuildingIdentifier | |
| | | Varchar | SuiteIdentifier | |
| | | Varchar | FloorIdentifier | |
| | | Varchar | DistrictName | |

| | | | | |
|---|---|---|---|---|
| | | Varchar | POB | option should only be used if the data cannot be presented in the AddressFix format.<br>NOTE: If AddressFix is selected, there will be the option of inputting the full street address of a Constituent Entity in the AddressFree element rather than using the related fixed elements. In this case, the city, subentity, and postal code information should still be entered in the appropriate fixed elements. |
| | | Varchar | PostCode | |
| | | Varchar | City | |
| | | Varchar | CountrySubentity | |
| **ReportingRole** | | 6-Character | | The Reporting Role element specifies the role of the Reporting Entity with respect to the filing of the CbC Report. Possible values are:<br>CBC701 – Ultimate Parent Entity<br>CBC702 – Surrogate Parent Entity |
| **DocSpec : 1** | DocTypeIndic | 6-Character | | This element specifies the type of data being submitted. Allowable entries are:<br>OECD0 = Resent Data<br>OECD1 = New Data<br>OECD2 = Corrected Data<br>OECD3 = Deletion of Data OECD10 = Resent Test Data OECD11 = New Test Data<br>OECD12 = Corrected Test Data<br>OECD13 = Deletion of Test Data |
| | DocRefId | Varchar | | Refer here for more information.<br>Format :<br>**MY<fiscal year (YYYY)>-<MYCBCID><Date of creation (YYYYMMDD)><E for Entity or R for CbCR report (1 char)><4-digit Incremental number (0001)>**<br><br>The 4-digit incremental number is one that starts at '0001' and increases to '9999' when a file is produced **on the same day**.<br><br>e.g.<br>MY2017-20000001201811O8E0001<br>MY2017-20000001201811O8R0001<br><br>** need to do record level validation must unique across the board |
| | CorrDocRefId | Varchar | | Refer here for more information. |
| | CorrMessageRefId | Varchar | | |

## CbC Reports [Section IIIb]



| Field Name | Datatype | Status | Remarks |
|---|---|---|---|
| **CbCR Body : 1..∞**<br>**CbcReports : 0-∞** | | | |
| Refer here for more information. | | | |

# STEP 2 - DIGITALLY SIGN CbCR XML FILE

Digital signatures are used to assure data integrity, which means that the messages are not altered in transmission. HiDEF can verify that the received message is identical to the sent message. MNE uses its private key to digitally sign the message. Senders (MNE) and recipient (IRBM) of CbCR files will ensure that the file was not corrupted during compression, encryption, and decryption, or altered during transmission to or from HiDEF. For information on the Signed CbCR XML, see CbCR XML File Sample and download <MyCBCID>_CBC_Payload (Signed) zip file, please click here.

**Sign XML File:**

| Process | Description | File Naming Convention |
|---|---|---|
| Sign XML File | • Prepare the CbCR reporting data using XML element prefixes. Do not use the default namespaces.<br>• To generate the digital signature1, the XML file is processed by a “one-way hashing” algorithm to generate a fixed length message digest.<br>• Depending on the tool used to perform the digital signature, a different type of canonicalization method may be required. The following methods are acceptable:<br>o <Canonicalization Method Algorithm="http://www.w3.org/2001/10/xmlexc-c14n#"/><br>o <Canonicalization Method Algorithm="http://www.w3.org/TR/2001/RECxml-c14n-20010315"/><br>• IRBM requires that the payload file be signed by first creating a SHA2-256 hash. The Sender will then create an RSA digital signature using the 2048-bit private key that corresponds to the public key found in the Sender’s digital certificate on HiDEF.<br>• After validating the schema, digitally sign the CbCR XML file using W3C Recommendation XML Signature Syntax and Processing (Second Edition)3 “enveloping” signature.<br>• Use the digital signature “enveloping” type. The “enveloped and detached” types will cause the transmission to fail.<br>• The file name is “<MyCBCID>_CBC_Payload.xml”. The file is case sensitive and any variation in file name or format will cause the transmission to fail. | <MyCBCID>_CBC_Payload.xml |

# STEP 3 - COMPRESS THE XML FILE

The XML file "<MyCBCID>_CBC_Payload.xml" should be compressed using a compatible compression utility and the standard Deflate compression method.

**Recommended compression tools.**

| Tool | Version | Operating System |
|---|---|---|
| Winzip | 17.5 | Windows |
| 7-zip | 9.2 | Windows or Linux |
| Windows Built-in zip utility | N/A | Windows |
| Linux/Unix standard zip utility | N/A | Linux/Unix |
| Apple built-in archive utility | MAC OS X 10.3 and later | MAC |

**Compress XML File**

| Process | Description | File Naming Convention |
|---|---|---|
| Compress XML File | • The compressed file "zip" is the file extension used by the compression tool or library.<br>• Other tools may be used but the compression method must be recognized by one of the five tools or libraries for the file to be successfully processed. | <MyCBCID>_CBC_Payload.zip |
| Summary | • If the file is not recognized or processing fails, the file will be rejected. The sending partner will receive a notification that explains the reason for the transmission failure and how to modify and resubmit the file.<br>• The file name "<MyCBCID>_CBC_Payload.zip". The file is case sensitive and any variation in file name or format will cause the transmission to fail.<br>• Note: The current supported compression is ZIP compression using the standard Deflate compression method. | N/A |

# STEP 4 - ENCRYPT THE XML FILE WITH AES 256 KEY

AES is one of the most secure encryption algorithms and the preferred encryption standard for HiDEF. The file is encrypted to protect MNE and taxpayer sensitive information.

| Process | Description | File Naming Convention |
|---|---|---|
| Encrypt XML File | • After compression, encrypt the file "<MyCBCID>_CBC_Payload.zip" using the AES-256 cipher with a randomly generated "one-time use" AES key.<br><br>• While performing AES encryption, there are several settings and options depending on the tool used to perform encryption. IRBM recommended settings should be used to maintain compatibility:<br>○ Cipher Mode: CBC (Cipher Block Chaining).<br>○ Salt: No salt value<br>○ Initialization Vector: 16 bytes IV<br>○ Key Size: 256 bits / 32 bytes – Key size should be verified and moving the key across operating systems can affect the key size.<br>○ Encoding: There can be no special encoding. The file will contain only the raw encrypted bytes.<br>○ Padding: PKCS#7 or PKCS#5<br><br>• The AES encrypted file name is "<MyCBCID>_CBC_Payload". The file is case sensitive and any variation in file name or format will cause the transmission to fail. | <MyCBCID>_CBC_Payload |

# STEP 5 - ENCRYPT THE AES KEY WITH IRBM PUBLIC KEY

The next step is to encrypt the AES key with the IRBM public key. The file is encrypted to protect the AES key. All CbCR partners must validate the IRBM X.509 Digital Certificate to an approved CA. An X.509 Digital Certificate contains the public key for IRBM, and it can be retrieve from the IRBM website.

**Encrypt AES Key with Public Key:**

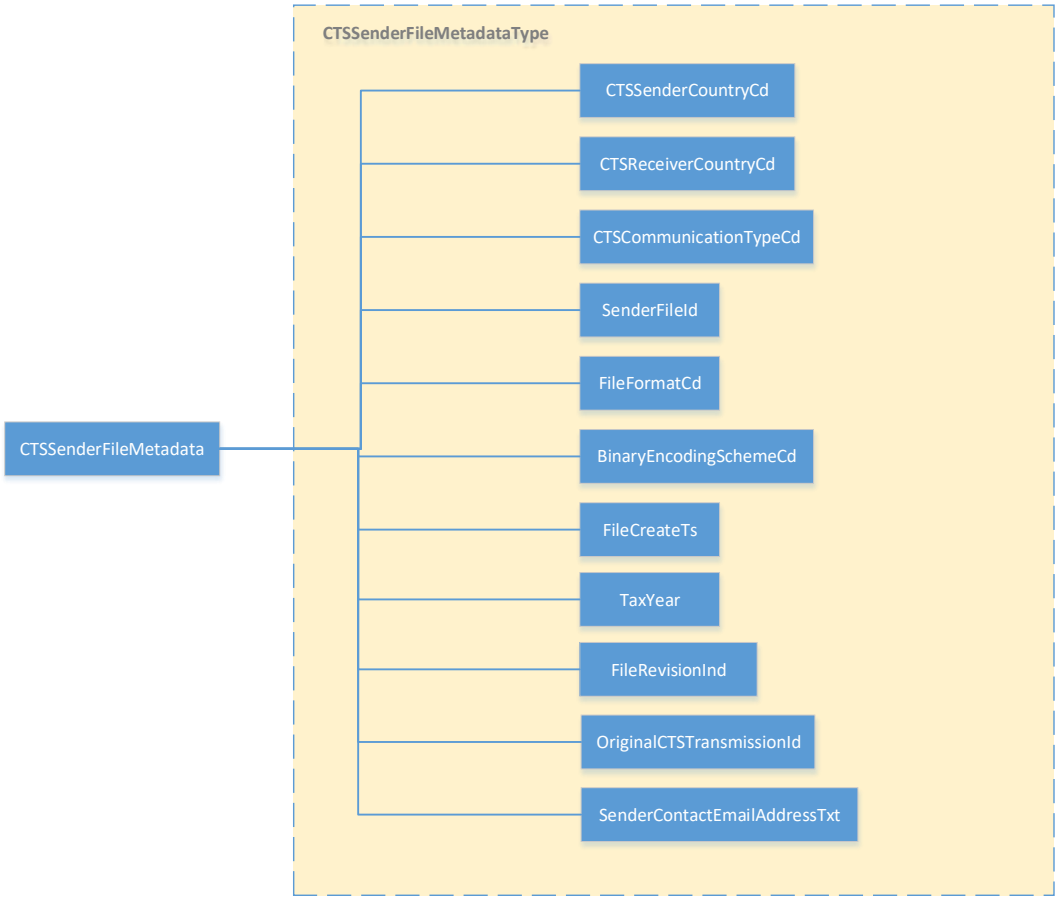
| Process | Description | File Naming Convention |
|---|---|---|
| Validate Certificate | • To validate the certificate:<br>1. Verify the certificate chain;<br>2. Check the revocation status of the certificate chain. There are two methods:<br>○ Retrieve a Certificate Revocation List (CRL) or<br>○ Send an Online Certificate Status Protocol (OCSP) query to a CA designated responder | N/A |
| Encrypt the AES Key | • After validating the certificate, use the public key from the IRBM certificate to encrypt the AES 256 key.<br><br>• The public key encryption uses the standard RSA algorithm. While performing AES encryption, there are | <MyCBCID>_CBC_Key |

| | | |
|---|---|---|
| | several settings and options depending on the tool used. IRBM recommended settings should be used to maintain compatibility:<br>○ Padding: PKCS#1 v1.5<br>○ Key Size: 2048 bits<br><br>• The encrypted file name is "<MyCBCID>_CBC_Key". "MyCBCID" is the 8-character HiDEF login id. | |
| Summary | • CbCR reporting with IRBM as a recipient will have two encrypted files. The files are case sensitive and any variation in file name or format will cause the transmission to fail:<br>1. Symmetric encryption - the AES 256 encrypted CbCR XML file name is "<MyCBCID>_CBC_Payload"<br>2. Asymmetric encryption - the public key encrypted AES 256 key file name is "<MyCBCID>_CBC_Key" | N/A |

## STEP 6 - CREATE SENDER METADATA FILE

Users can create a sender metadata file to ensure that IRBM accurately process CbCR XML files and notifications. Notifications are sent by the IRBM to an MNE and state whether the file is processed correctly or contained errors.

MNE must create a metadata file to attach to the payload before uploading to HiDEF. The CbCR Sender Metadata XML file name is "<MyCBCID>_CBC_Metadata.xml." All MNE must provide the values for the elements in the sender metadata file.



MNE must provide the values for the required elements in the Metadata. Please note that the Metadata file is validated by the HiDEF system and if the required information is missing, the uploading process will fail.

The content to be provided in the different elements of the CbCR Metadata Schema is as follows:

- The **CTSSenderCountryCd** element identifies the jurisdiction of the Sending Competent Authority. Only a value MY is currently allowed.

- The **CTSReceiverCountryCd** element indicates the jurisdiction of the Receiving Competent Authority. Only a value MY is currently allowed.

- The **CTSCommunicationTypeCd** element specifies the type of message transmitted. Only a value CBC is currently allowed.

- The **SenderFileID** element is a free text field to capture the file name or ID created by the Multinational Entities. The element helps both the Sending MNE and Receiving IRBM Authority to track and monitor a specific message. The agreed format is:

  CBC_ MY<Fiscal Year>-<MyCBCID><Date><SEQNO>

  - Fiscal Year format - YYYY
  - MyCBCID – will be given upon registration of MNE with HiDEF
  - Date format – YYYYMMDD
  - SEQNO – 4 digits character (0000 – 9999)

  For example, a sender with a MyCBCID of "20000001" that transmits a data packet on January 15, 2018 at 16:30:45 for fiscal year 2017 can create a SenderFileID as:

  CBC_MY2017-20000001201801150001

- The **FileFormatCd** element specifies the file format of message transmitted, the only allowable value being XML.

- The **BinaryEncodingSchemeCd** element identifies the type of encoding scheme for the transmission payload. If sending an XML file, the value should be 'NONE'.

- The **FileCreateTs** element identifies the timestamp for the transmission payload created by the Multinational Entities. The format for use is YYYY-MM-DD'T'hh:mm:ss'Z'. Fractions of seconds may be used. Example: 2018-02-15T14:37:40Z.

- The **TaxYear** element specifying the tax year to which the file relates.

- The **FileRevisionInd** element is a Boolean field to indicate if the file is a revised message. The only allowable values are "true" or "false".

- The **OrginialCTSTransmissionId** element is a free text field to reference the unique original HiDEF transmission ID. The identifier helps both the MNE and IRBM to track and monitor messages. HiDEF Transmission ID referencing an update to an earlier transmission
  - Optional – Use only after IRBM request

- The **SenderContactEmailAddressTxt** element is a free text field to identify the email address of the Multinational Entities.

# STEP 7 - CREATE A CBC DATA PACKET

A file that is transmitted through HiDEF is known as a CbCR transmission file or data packet. The data packet is an archive in .ZIP file format, and it should be created using one of the compatible data compression tools described in Step 3. HiDEF only supports data packets in a .ZIP file format with a .zip file extension. The files are case sensitive and any variation in the file name or format will cause the transmission to fail.

Files contained in a transmission archive or data packet :

- <MyCBCID>_CBC_Metadata.xml
- <MyCBCID>_CBC_Key
- <MyCBCID>_CBC_Payload

The file naming convention of data packet is composed of a Coordinated Universal Time (UTC) timestamp and the MyCBCID of the sender as:

| File Name | Description |
|---|---|
| **<MyCBCID>_<RPTYR>_CBC_<UTC>.zip** | Transmission file to be sent through the HiDEF |

The timestamp format of the UTC is YYYYMMDDTHHMMSSmsZ where:

- YYYY = 4-digit year
- MM = 2-digit month
- DD = 2-digit day
- HH = 24-hour
- MM = 2-digit minutes
- SS = 2-digit seconds
- ms = 3-digit milliseconds

For example, a sender with a MyCBCID of "20000001" that transmits a data packet on January 15, 2015 at 16:30:45.123 for reporting year 2014 can create a data packet named as:

- 20000001_2014_CBC_20150115T163045123Z.zip

# STEP 8 - TRANSMIT DATA PACKET USING HiDEF

After the archive is uploaded and transmitted, HiDEF sends an alert to the Multinational Entities via email. The message provides status information about the file upload. If the upload and HiDEF file checks are successful, HiDEF assigns a unique "TransmissionID" in the email. If there is an error, the HiDEF alert provides an appropriate error code in the email message.

# 3. HiDEF PUBLIC KEY INFRASTRUCTURE (PKI)

## 3.1 CURRENT LIST OF APPROVED CERTIFICATE AUTHORITIES

| Certificate Authority | Type of Certificate | External Website Links |
|---|---|---|
| Digicert® | SSL PlusTM (Single Name) | https://www.digicert.com/welcome/ssl-plus.htm |
| Entrust® | Standard SSL | http://www.entrust.net/ssl-certificates/standard.htm |
| GlobalSign® | Organization SSL | https://www.globalsign.com/ssl/organization-ssl/ |
| IdenTrust | TrustID Server (SSL) | https://www.identrust.com/certificates/buy_trustid_server.htmlhttp://identrust.com/irs/fatca/index.html |
| StartCom® | StartSSLTM EV | https://www.startssl.com/?app=30 |
| Symantec/Verisign | Secure Site SSL | http://www.symantec.com/ssl-certificates/secure-site/?inid=vrsn_symc_ssl_SS |
| Thawte® | SSL Web Server | http://www.thawte.com/ssl/web-server-ssl-certificates/index.html |

## 3.2 LOCAL RESELLER

| Certificate Authority | Type of Certificate | External Website Links |
|---|---|---|
| POS Digicert Sdn Bhd | SSL Certificates | https://www.posdigicert.com.my |
| TM Applied Business | SSL Certificates | http://www.tab.com.my |
| MSC Trustgate.com Sdn. Bhd. | SSL Certificates | https://www.msctrustgate.com |

## 3.3 CERTIFICATE FORMAT

Before you begin the CBC registration process, each Entities should obtain one valid digital certificate issued by an approved certificate authority (CA). Certificates in other formats, such as self-sign will be rejected. HiDEF will ONLY accept digital certificates issued by an approved CA.

Supported formats for the digital certificates are:

- Distinguished Encoding Rules (DER) binary X.509
- Privacy Enhanced eMail (PEM) ASCII (Base-64) encoded X.509

HiDEF will accept digital certificates (Public Key) in .crt format for storage and retrieval. If a digital certificate is not in CRT (Public Key) format, use openssl to convert it to .crt format.

## 3.4 UPLOAD A DIGITAL CERTIFICATE TO HiDEF

After a Multinational Entities (MNE) obtains a digital certificate, the MNE will provide the certificate to HiDEF. In order to do that, MNE is required to login to HiDEF using <MyCBCID> created after registration and upload the certificate. Upon upload, the certificate is validated by the Certificate Authority (CA) that issued the certificate. It is the responsibility of HiDEF users to verify that the certificate is valid before use. MNE must use different digital certificate for multiple entities.

(Refer to Current list of approved Certificate Authorities for the HiDEF and Local Reseller in para 3.1 and 3.2)

## 3.5 PUBLIC KEY CERTIFICATE

A public key certificate, also known as a digital certificate, is an electronic document used to prove ownership of a public key. The IRBM public key certificate can be downloaded from IRBM website.

# 4. SEQUENCE OF EXCHANGES THROUGH THE HIDEF

This section contains an example of CBC and CBC Status Message exchanges through the Hasil Data Exchange Facility (HiDEF).

Please note that file preparation for CBC XML Schema will need to be performed for all exchanges through the HiDEF.

As an example, for an exchange of CBC information between Malaysia Multinational Entities (MyMNE) and HiDEF, the following events occur:

1. MyMNE sends a CBC message with new data to HiDEF
   – HiDEF is not able to decrypt the file and sends a CBC Status Message

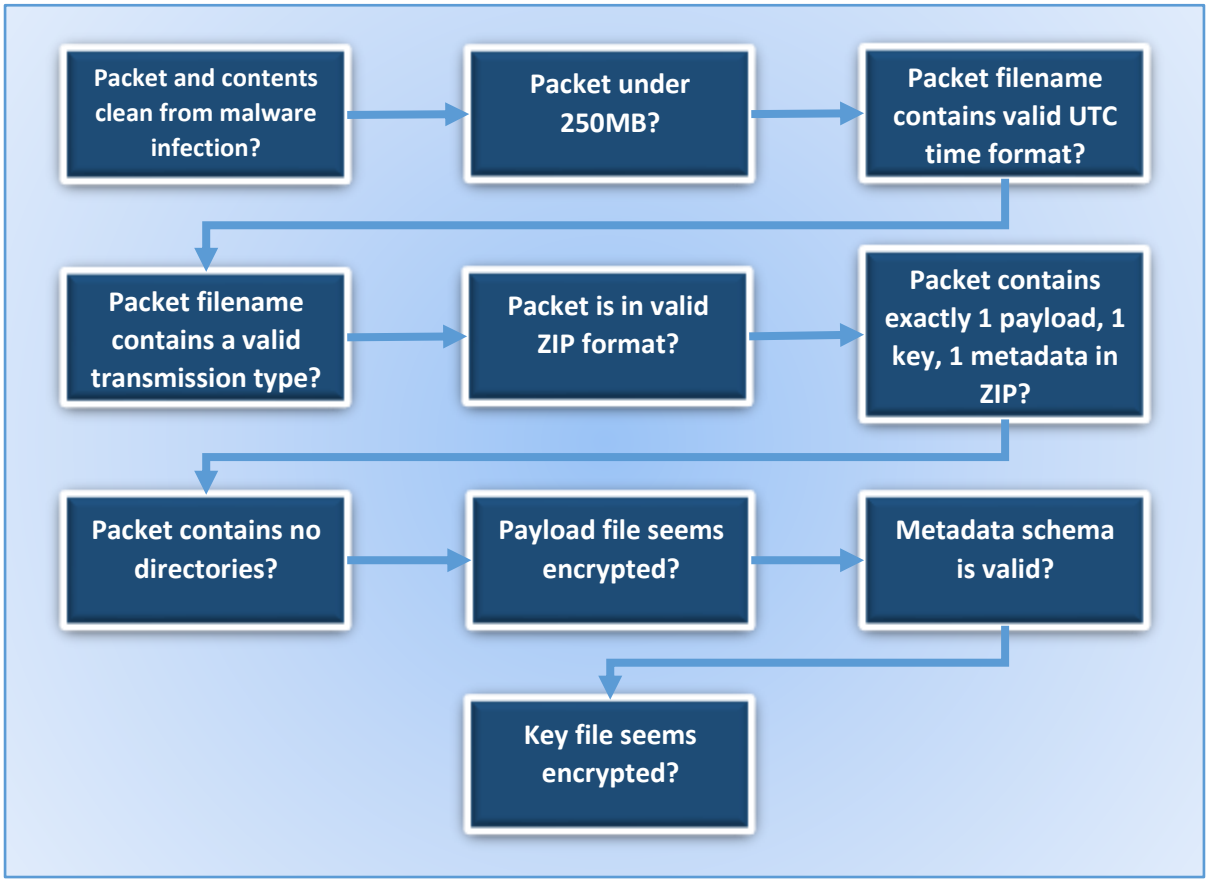
2. MyMNE corrects the file with proper encryption
   – HiDEF found XML validation errors and send a CBC Status Message

3. MyMNE corrects the XML validation issues and resubmits the file
   – HiDEF found no file error, but ten (minor) record errors. HiDEF accepts the file

4. MyMNE corrects the ten records errors (the file contains the ten corrected records)
   – HiDEF found no further errors. HiDEF accepts the file.


As shown in the diagram above, file preparation is required for all files sent, i.e. for CBC XML Schema files.\

# 5. TRANSMISSION PACKET VALIDATION




# 6. VALIDATION PROCESS

This section contains further guidance on the error codes to be used for indicating a file or record error within the XML Schema. Only such codes explicitly stated in this section should be provided in the CBC Status Message XML Schema. This message will be sent to Multinational Entities via email. *Please note that the File and Record Validations section may be subject to subsequent change.*

## FILE ERRORS

For file errors, only one Status Message should be sent for a specific MessageRefID (i.e. for a specific CBC Report file), but a different CTSTransmissionID should be provided. For example, the first time a file is sent, HiDEF could return the Failed Decryption error via the Status Message. In such case, XML validation and other subsequent validations have not been performed since the file could not be decrypted.

## RECORD ERRORS

For record errors, only one Status Message should be sent for a specific MessageRefID (i.e. for a specific CBC Report file).

## 6.1 FILE VALIDATIONS (50 000 – 59 999)

Please do not submit a request to correct or delete any of the records in this file until you receive a Status Message that this file has been received as valid (Status is Accepted). In case a file error is detected, the file should be resubmitted by the sender, using a new, unique MessageRefID.

| Failed Decryption (50002) | |
|---|---|
| **File error code** | 50002 |
| **File error description** | HiDEF could not decrypt the referenced file. |
| **Action Requested** | Please re-encrypt the file with a valid IRBM Public key and resubmit the file. |

| Failed Decompression (50003) | |
|---|---|
| **File error code** | 50003 |
| **File error description** | HiDEF could not decompress the referenced file. |
| **Action Requested** | Please compress the file (before encrypting) and resubmit the file with a new unique MessageRefID. |

| Failed Signature Check (50004) | |
|---|---|
| **File error code** | 50004 |
| **File error description** | HiDEF could not validate the digital signature on the referenced file. |
| **Action Requested** | Please re-sign the file with the owner's private key using procedures [as defined in the context of the HiDEF]. |

| Failed Threat Scan (50005) | |
|---|---|
| **File error code** | 50005 |
| **File error description** | HiDEF detected one or more potential security threats within the decrypted version of the referenced file. Such threats include but are not limited to hyperlinks, Java script, and executable files. |
| **Action Requested** | Please scan the file for known threats and viruses, remove all detected threats and viruses prior to encryption and re-encrypt and resubmit the file. |

| Failed Virus Scan (50006) | |
|---|---|
| **File error code** | 50006 |
| **File error description** | HiDEF detected one or more known viruses within the decrypted version of the referenced file. |
| **Action Requested** | Please scan the file for known threats and viruses, remove all detected threats and viruses prior to encryption, and re-encrypt and resubmit the file. |

| Failed Schema Validation (50007) | |
|---|---|
| **File error code** | 50007 |
| **File error description** | The referenced file failed validation against the relevant XML Schema. |
| **Action Requested** | Please re-validate the file against the relevant XML Schema, resolve any validation errors, and re- encrypt and resubmit the file. |

| Invalid MessageRefID format (50008) | |
|---|---|
| **File error code** | 50008 |
| **File error description** | The structure of the MessageRefID is not in the correct format, as set out in the relevant User Guide. |
| **Action Requested** | Please ensure the MessageRefID follows structure defined in the relevant User guide, and resubmit the file. |

| MessageRefID has already been used (50009) | |
|---|---|
| **File error code** | 50009 |
| **File error description** | The referenced file has a duplicate MessageRefID value that was received on a previous file. |
| **Action Requested** | Please replace the MessageRefID field value with a new unique value (not containing all blanks), and resubmit the file. |

| File Contains Test Data for Production Environment (50010) | |
|---|---|
| **File error code** | 50010 |
| **File error description** | The referenced file contains one or more records with a DocTypeIndic value in the range OECD10-OECD13, indicating test data. As a result, HiDEF cannot accept this file as a valid CBC file submission.<br><br>For more information on the DocTypeIndic data element, please consult the CBC User Guide. |
| **Action Requested** | If this file was intended to be submitted as a valid CBC file, please resubmit with DocTypeIndic values in the range OECD0-OECD3 (see CBC User guide). [If this file was intended as a test file, please submit to the CTS test environment during an agreed test window.] |

| File Contains Production Data for Test Environment (50011) | |
|---|---|
| **File error code** | 50011 |
| **File error description** | The referenced file was received in a test environment with one or more records having a DocTypeIndic value in the range OECD0-OECD3. These DocTypeIndic values indicate data in this file may have been intended as a valid file submission. Messages received in test environments are not accepted by HiDEF as a valid file submission. Submissions to the test environment should only include records with DocTypeIndic in the range OECD10-OECD13, indicating test files. |
| **Action Requested** | If this file was intended to be submitted as a valid file, please resubmit with DocTypeIndic values in the range OECD0-OECD3. [If this file was intended as a test file, please correct the DocTypeIndic for all records and resubmit to the CTS test link.] |

| An incorrect AES key size was detected by the receiving jurisdiction (50013) | |
|---|---|
| **File error code** | 50013 |
| **File error description** | The recipient has detected one or more of the following errors:<br>• Data packet transmitted with ECB cipher mode (or any cipher mode other than CBC)<br>• Data packet does not include IV in Key File<br>• Combined (IV and AES) data packet key size is not 48 bytes<br>• Data packet does not contain the concatenated key and IV. |
| **Action Requested** | The sending Competent Authority should resend the file (newly encrypted, with a new unique MessageRefID and with the correct AES key size). |

## 6.2 RECORD VALIDATION – FIELDS USED FOR THE CORRECTION PROCESS (80 000 – 89 999)

The record error codes indicate errors that have been detected in the context of the correction of previously sent records.

| Record Validation – Fields used for the correction process | | |
|---|---|---|
| **Record Error Code** | **Validation name** | **Validation description** |
| 80000 | DocRefID already used | The DocRefID is already used for another record. |
| 80001 | DocRefID format | The structure of the DocRefID is not in the correct format, as set out in the User Guide. |
| 80002 | CorrDocRefId unknown | The CorrDocRefId refers to an unknown record. |
| 80003 | CorrDocRefId no longer valid | The corrected record is no longer valid (invalidated or outdated by a previous correction message). As a consequence, no further information should have been received on this version of the record. |
| 80004 | CorrDocRefId for new data | The initial element specifies a CorrDocRefId. |
| 80005 | Missing CorrDocRefId | The corrected element does not specify any CorrDocRefId. |
| 80006 | DocSpec. CorrMessage RefID | The CorrMessageRefID is forbidden within the DocSpec_Type. |
| 80007 | MessageSpec. CorrMessage RefID | The CorrMessageRefID is forbidden within the Message Header. |
| 80010 | Message TypeIndic | A message can contain either new records (OECD1) or corrections/deletions (OECD2 and OECD3), but should not contain a mixture of both. |
| 80011 | CorrDocRefID twice in same message | The same DocRefID cannot be corrected or deleted twice in the same message. |

# 7. HiDEF FILE PREPARATION TOOL (sample codes)

The HiDEF Data Preparation OpenSSL project repository demonstrates a sample working application developed using DOS Script.

1) This application/script will create a randomly one time use AES key with 48bytes size (32 bytes AES Key + 16 bytes IV).
2) This AES key is then being used to encrypt the Payload zip file.
   Payload zip file contains CBC XML Payload file which has been validated against CBC XML Schema and this XML file has been signed by Multinational Entities using their Private Key. This Process is not included in this sample codes.
3) The AES key used to encrypt the payload zip file is then be encrypt using the IRBM Public key which can be download from IRBM website.
4) Multinational Entities need to create a transmission file by including the Metadata.xml file, Encrypted Payload file and Encrypted AES key by compressing these files into one file called data packet. Please refer to STEP 7 - CREATE CBC DATA PACKAGES for more information.

```
setlocal enabledelayedexpansion
openssl rand 48 > 48byterandomvalue.bin
hexdump -B 48byterandomvalue.bin > 48byterandomvalue.txt

set /a counter=0
for /f "tokens=* delims= " %%i in (48byterandomvalue.txt) do (
set /a counter=!counter!+1
set var=%%i
if "!counter!"=="1" (set aes1=%%i)
if "!counter!"=="2" (set aes2=%%i)
if "!counter!"=="3" (set iv=%%i)
)

set result1=%aes1:~0,50%
set result1=%result1: =%
set result2=%aes2:~0,50%
set result2=%result2: =%
set aeskey=%result1%%result2%
set initvector=%iv:~0,50%
set initvector=%initvector: =%

openssl aes-256-cbc -e -in <MyCBCID>_CBC_Payload.zip -out <MyCBCID>_CBC_Payload -K %aeskey% -
iv %initvector%

openssl rsautl -encrypt -certin -inkey %IRBMPublicKey% -in 48byterandomvalue.bin -out
<MyCBCID>_CBC_Key
```