Knowledge Document

Question: **Are you an Inbound or an Outbound CDF3 customer?**
Answer: The CDF3 layout is the default layout used both for files coming into the MasterCard Provider System and for files being shipped out of the MasterCard Provider System.

If you are a Processor or Issuer sending files TO MasterCard, then you are an ***INBOUND*** customer.

If you are a Corporation or a third-party host for corporations, receiving files FROM MasterCard, then you are an ***OUTBOUND*** customer.

Question: **Do you have the current CDF3 document?**
Answer : Please visit the SmartData Support website at http://smartdatasupport.mastercard.com/cdfinbound.html to ensure you are utilizing the most current specifications. You may enter an email address if you wish to be notified of changes to the specifications.

Question: **How does the CDF3 system work? When will the Smart Data files be distributed?**
Answer: The MasterCard Provider system runs 20 hours per day, 6 days per week (subject to change with limited notice). All inbound files are queued immediately for processing. Unless a file is received during an outage window, processing of the file will start no longer than 4 hours after receipt. In most cases, the outbound corporation files are distributed as soon as they are available. This means that the delivery of an outbound file is dependant almost entirely on the arrival of the source file at MasterCard, not on a MasterCard batch schedule.

Question: **Why is CDF3 an XML file?**
Answer: The decision to go with an XML format was based on the following factors:

- The fast-growing, broad acceptance of XML as a standard data format
- The flexibility for adding new data elements to an XML Schema, in a fast-changing business environment
- The removal of the need for Issuers/Processors to populate every field in the layout, whether it is relevant or not. While XML files are inherently larger than untagged formats, there is a great saving in the development and certification costs of an XML format file as well as the removal of the redundancy of shipping blank, zero, null and defaulted field values
- The "human readability" of XML will help expedite transmission trouble-shooting

At first glance, this can seem like a very intimidating document. 2000+ data elements is a significantly greater number of elements than is seen in most file formats in the credit card industry.

The second thing to do is to identify which data elements are applicable to the system which is sending data to MasterCard. If there are single data elements or whole groups of data elements that cannot be supplied then they can simply be ignored. Because the XML tags will not be sent for elements that cannot be supplied, there is no need to be concerned with the sending of default values in those fields.

Question: **What does the Data look like?**
Answer: Where applicable, ISO standards apply to fields in this file. Many acceptable values for fields are listed in the Schema Field definitions section of the layout document.

Question: **What does each attribute in the XML Schema Document (CDFFileTransmission.xsd) mean?**
Answer: The following example will explain each attribute:

```
<xs:element name="BillingType" type="BillingType" minOccurs="1"
maxOccurs="1"></xs:element>
```

- name= The name of the tag in the XML file. In this instance, the name of the tag is BillingType.
- type= The data type restrictions placed on this field. In other words, the acceptable types and values for this tag. In this instance the data type is BillingType.
- minOccurs= The minimum number of times a tag can occur.
- maxOccurs= The maximum number of times a tag can occur.

Question: **How do I determine if an XML tag (field) is required?**
Answer: Each tag in the XML Schema Document (CDFFileTransmission.xsd) contains minOccurs and maxOccurs attributes. The minOccurs tells you the minimum number of tags that must be contained in the file. The maxOccurs tell you the maximum number of tags that can be contained in the file.

Example 1:

```
<xs:element name="BillingType" type="BillingType" minOccurs="1"
maxOccurs="1"></xs:element>
```

This notation implies that the tag BillingType is required. There can be one and only one entry for this tag.

Example 2:

```
<xs:element name="ProcessingCorporateAccount" type="Id19Type"
minOccurs="0" maxOccurs="1"></xs:element>
```

This notation implies that the tag ProcessingCorporateAccount is not required. There can be zero to one entry for this tag.

Question: **How do I determine the data type and acceptable value for each tag (field)?**
Answer: Each tag in the XML Schema Document (CDFFileTransmission.xsd) contains a type= attribute. This attribute is used to detetermine not only the data type and size, but in some instances, acceptable values.

Example 1:
```
<xs:element name="AccountNumber" type="Id19Type" minOccurs="0"
maxOccurs="1"></xs:element>
```

The AccountNumber tag has a type of Id19Type. You will need to find the definition of this type in the CDFFileTransmission.xsd file.

```
<xs:simpleType name="Id19Type">
      <xs:restriction base="xs:token">
            <xs:minLength value="1"></xs:minLength>
            <xs:maxLength value="19"></xs:maxLength>
      </xs:restriction>
</xs:simpleType>
```

This Id19Type is a simple data type. In this example, the AccountNumber can have a length between 1 and 19 characters inclusive.

Valid Example: <AccountNumber>5499000000000000</AccountNumber>

Example 2:
```
<xs:element name="TransactionDate" type="MasterCardDateType"
minOccurs="1" maxOccurs="1"></xs:element>
```

The TransactionDate tag has a type of MasterCardDateType. You will need to find the definition of this type in the CDFFileTransmission.xsd file.

```
<xs:simpleType name="MasterCardDateType">
      <xs:restriction base="xs:date"></xs:restriction>
</xs:simpleType>
```

This MasterCardDateType is a simple data type. In this example, the TransactionDate can be a valid date in the CCYY-MM-DD format.

Valid Example: <TransactionDate>2003-01-31</TransactionDate>

Example 3:
```
<xs:element name="DebitOrCreditIndicator" type="SignCodeType"
minOccurs="1" maxOccurs="1"></xs:element>
```

The DebitOrCreditIndicator tag has a type of SignCodeType. You will need to find the definition of this type in the CDFFileTransmission.xsd file.

```
      <xs:simpleType name="SignCodeType">
            <xs:restriction base="xs:token">
                  <xs:enumeration value="C"></xs:enumeration>
                  <xs:enumeration value="D"></xs:enumeration>
            </xs:restriction>
      </xs:simpleType>
```

This SignCodeType is a simple data type with a restriction (list of acceptable values). In this example, the DebitOrCreditIndicator tag can only contain either a C or a D. Any other values will be rejected.

Valid Example: <DebitOrCreditIndicator>D</DebitOrCreditIndicator>

Example 4:

```
<xs:element name="AmountInOriginalCurrency" type="CurrencyAmountType"
minOccurs="1" maxOccurs="1"></xs:element
```

The AmountInOriginalCurrency tag has a type of CurrencyAmountType. You will need to find the definition of this type in the CDFFileTransmission.xsd file.

```
<xs:complexType name="CurrencyAmountType">
            <xs:simpleContent>
                  <xs:extension base="CurrencyType">
                        <xs:attribute name="CurrencyCode"
                        type="CurrencyCodeType"
                        use="required"></xs:attribute>
                        <xs:attribute name="CurrencyExponent"
                        type="CurrencyExponentType"
                        use="required"></xs:attribute>
                        <xs:attribute name="CurrencySign"
                        type="SignCodeType"
                        use="required"></xs:attribute>
                  </xs:extension>
            </xs:simpleContent>
      </xs:complexType>
```

This CurrencyAmountType is a complex data type that consists of many tags. This field consists of a CurrencyCode, CurrencyExponent and CurrencySign. Each one of these tags has their own type= that you must look up to determine the appropriate values.

Valid Example: <AmountInOriginalCurrency CurrencySign=”D” CurrencyExponent=”2” CurrencyCode=”840”>10938</AmountInOriginalCurrency>

The above example represents a debit amount in the 840 currency of 109.38.

Question: **Do I put in the decimal (“.”) character in amount fields?**
Answer: No, the CDF v3.0 XML Schema Document (CDFFileTransmission.xsd) states that no decimals are to be placed in amount fields. The concept of an exponent is used in the CDF v3.0 document.

Example 1: Incorrect way to represent 1,234.56

<AmountInOriginalCurrency CurrencySign=”D” CurrencyExponent=”2” CurrencyCode=”840”>1,234.56</AmountInOriginalCurrency>

Example 2: Correct way to represent 1,234.56

<AmountInOriginalCurrency CurrencySign=”D” CurrencyExponent=”2” CurrencyCode=”840”>123456</AmountInOriginalCurrency>

Question: **Is the CDF v3.0 file case sensitive?**
Answer: Yes, the tags (fields) must match exactly to the XML Schema Document (CDFFileTransmission.xsd).

Example:

If the schema states the following: `<xs:element name="TotalNumOfAccounts" type="Numeric10Type" minOccurs="0" maxOccurs="1"></xs:element>`

The following would be rejected due to the items highlighted in blue:

<TotalnumofAccounts>10</TotalnumofAccounts>

Question: **What is meant by the file will be rejected?**
Answer: Each and every CDF v3.0 file that is sent to MasterCard International is validated against the schema before it is loaded in to the Provider System. Validating against the schema (CDFFileTransmission.xsd) means that MasterCard International will take your input file and determine if any of the rules have been violated.

Examples of items that will not pass validation are: a) missing required fields b) incorrect data length c) incorrect data type d) incorrect data format (invalid date format) e) tags that are misspelled or have incorrect character case f) etc…

MasterCard International strongly recommends all CDF v3.0 files be validated against the schema before they are sent to MasterCard International. This will make it much easier on the data provider to fix data errors before they are sent to MasterCard International.

Question: **XML does not allow certain characters. What do I do?**
Answer: XML control characters need to be sent as indicated below. For example, if you have a merchant named “Steve & Jose Bookstore” you would need to enter this in the XML file as Steve &amp; Jose Bookstore”.

Character	***Reference***
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	&apos;

Question: **ProcessorNumber is an 11 digit number. Our processor number contains less positions. What exact data should we send?**
Answer: The ID11Type field is a minimum length of 1 and a maximum length of 11. If your Processor Number is 1111, you may send 1111 or you may send 00000001111, so long as you meet the minimum and maximum limitations.

Question: **May we change the XML version or encoding format in the CDF 3 file?**
Answer: No, the first two lines of the CDF 3 file must be send as identified in the Sample File.

```
  <?xml version="1.0" encoding="utf-8" ?>
- <CDFTransmissionFile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

Question: **The sequence numbering in the sample file does not look sequential. Does it need to be?**
Answer: Sequencing is used for error reporting and each instance should be unique throughout the file.The sequence number of the hierarchyrecordheader may be a continuation of the Sequence Number in the Financial Record Header.

Question: Is **the data in DebitOrCreditIndicator disregarded in a reversal or adjustment? Is a Retail Sale and Retail Sales Reversal both debit, or is the first one debit and the other one is credit?**
Answer: If the transaction increases the amount the cardholder will owe, it is a debit. If the transaction decreases the amount the cardholder will owe, it is a credit, regardless of the transaction type.

Question: **May we use the AcquirerReferenceData as the ProcessorTransactionID?**

Answer: No, it is not recommended. AcquirerReferenceData is assigned by the Acquirer when the transaction is processed by the Merchant's bank, and is not guaranteed to be unique among all Acquirer.

Question: In the TelephonyBilingDetail_5121, the tags CallToCountryCode and CallFromCountryCode are defined as 3 character ISO Country Codes, but IPM message PDS 0638s3 and 0644s3 are both 40 position fields. Also, CallToStateProvince and CallFromStateProvince also do not match IPM mapping. Is there a conversion table or should it be consistent with the IPM definitions?
Answer: No, there is no conversion chart. As these are not mandatory tags, if the data (or a way to convert it) is not available, do not provide the fields.

Question: How do I reverse a transaction?
Answer: Edit the reversal flag in the 5000 financial to an "R". Change the debit/credit indicator to the reverse of the original transaction. Also change the TransactionAmount debit/credit indicators to the reverse. Resubmit transaction.

Question: How do I handle fields, such as the Account Number, that expanded from a 16 position numeric field to a 19 positions alphanumeric field?
Answer: If the field was established in GDR prior to CDFv3, the field was decimal aligned and zero filled. The field is now defined as alphanumeric, meaning that it can handle both numeric and alpha characters, so, in order to match the field in the database, the first 16 position must be numeric and the remaining three characters must be spaces. If the field is new to the database, the field can be configured anyway you wish. But, consistency is the key ingredient.

Default Values:
The following fields have default values in the Schema, and may be sent as empty tags as indicated below:

<AcceptanceBrandIdCode></AcceptanceBrandIdCode>	= MCC
<CorporateProduct></CorporateProduct>	= MCO
<AdjustmentReasonCode></AdjustmentReasonCode>	= - -