MDM FILE FORMAT AND LOAD PROCESS

PREPARED BY:	AEMO Markets
DOCUMENT REF:	
VERSION:	2.0~~1.1~~
EFFECTIVE DATE:	[TBA] December 2020~~1 December 2017~~
STATUS:	Initial Draft~~FINAL~~

Approved for distribution and use by:

APPROVED BY:	Peter Geers
TITLE:	Executive General Manager, Markets
DATE:	[TBA] 2019~~1/12/2017~~

Australian Energy Market Operator Ltd ABN 94 072 010 327 www.aemo.com.au info@aemo.com.au
NE Australian Energy Market Operator Ltd ABN 94 072 010 327 www.aemo.com.au info@aemo.com.au
NEW SOUTH WALES QUEENSLAND SOUTH AUSTRALIA VICTORIA AUSTRALIAN CAPITAL TERRITORY TASMANIA

VERSION RELEASE HISTORY

Version	Effective Date	Summary of Changes
0.10	August 2009	Draft compilation of details from numerous sources and documents to form one complete detailed process document.
0.20	December 2009	Draft updated per first round responses.
1.0	February 2010	Update per draft determination responses, issued as Final.
1.1~~0~~	1 December 2017	Updated to incorporate changes due to Power of Choice reforms and corrections.
2.0	[#] December 2020	Updated to incorporate: National Electricity Amendment (Five Minute Settlement) Rule 2017 No.15 and National Electricity Amendment (Global settlement and market reconciliation) Rule 2018 No. 14

CONTENTS

TABLES

FIGURES

1. INTRODUCTION

1.1 Purpose and Scope

This document specifies the Meter Data Management (MDM) Format to be used by MDPs for the provision of *metering data* to AEMO.

It also details the process for uploading ~~the~~ MDMF and MDFF files and the validations that occur when a file is submitted.

1.2 Definitions and Interpretation

The Retail Electricity Market Procedures – Glossary and Framework:

(a) is incorporated into and forms part of this document; and

(b) should be read with this document.

1.3 Related Documents

Title	Location
aseXML Schema	https://www.aemo.com.au/Gas/IT-systems-and-change/aseXML_standards/aseXML-Schemas~~https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/IT-systems-and-change/aseXML_standards/aseXML-Schemas~~
Hints and Tips – CATS & NMI Discovery	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Market-Settlement-and-Transfer-Solutions
CATS Procedures	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Market-Settlement-and-Transfer-Solutions
MDM Procedures	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Market-Settlement-and-Transfer-Solutions
Metrology Procedure: Part A	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering
Metrology Procedure: Part B	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering
NMI Procedure	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Market-Settlement-and-Transfer-Solutions
Retail Electricity Market Procedures – Glossary and Framework	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering~~/Glossary-and-Framework~~
B2B SMP Technical Guide	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Business-to-business-procedures
B2M e-Hub Technical Guide	

Title	Location
B2B Procedure Meter Data Process	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Business-to-business-procedures
MDFF Specification NEM12 NEM13	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Metering-procedures-guidelines-and-processes
Guide to MSATS Web Portal	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Market-Settlement-and-Transfer-Solutions
Guide to MSATS B2B	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Market-Settlement-and-Transfer-Solutions
Introduction to MSATS	https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Market-Settlement-and-Transfer-Solutions

2. OVERVIEW

2.1 Outline of Metering Data Management (MDM)

MDM is the centralised database of *metering data* ~~within MSATS~~. This database receives and holds *metering data* and stores *profile* information generated internally by AEMO~~MSATS~~.

MDM allows for storage of configuration data required to generate *profiles* that conform to the *metrology procedure* and substitutes of missing data.

MDM is the source of data for *settlements*.

2.2 Inputs Accepted by MDM

MDM accepts and stores all *metering data* in the *NEM*, which is submitted in the form of a Meter Data Notification~~*metering data* notification transaction~~.

The *metering data* ~~notification transaction~~ is submitted to AEMO~~MSATS~~ by MDPs in the form of a comma separated values-wrapped aseXML under the MDMT or MTRD Meter Data Notification transaction type~~file~~. Details of the creation and submission of ~~this file~~these transaction types are in section 3. There are a number of validation requirements undertaken on the *metering data* notification transactions before the data file is accepted by AEMO~~MSATS~~.

Data is required for all Datastreams in MSATS for any period of time where the Datastream Status Code is set to 'A' (Active). MDM stores this data for every Datastream against a certain *connection point* for *settlements* purposes.

3. PROCESS OF LOADING METERING DATA

3.1 Outline

Once *NMI* and *connection point* information is set up in MSATS, MDPs can upload *metering data* into MDM~~MSATS~~ for *settlements* processing, which can be delivered to MDM~~MSATS~~ via the MSATS ~~b~~Browser, FTP ~~or the~~ batch interface or e-Hub API.

(a) MSATS Browser – MDPs can load MDMT Meter Data Notification ~~*metering data*~~ manually using the MSATS Browser, 'Participant Inbox' screen and the 'Upload' action. The process steps are described in the Upload File section of the Guide to MSATS B2B.~~When MDPs click on the 'Upload' action, they are effectively placing the batch file into their 'Inbox' directory on the file share on the AEMO network.~~

MDPs can load MTRD Meter Data Notification manually using the MSATS Browser, 'B2B Browser > 'Upload File' screen. The process steps are described in the Data Load Import section of the Guide to MSATS Web Portal.~~When using the Browser interface the File Upload Screen will be used to transfer a *metering data* file to the appropriate AEMO file directory (or 'Participant Inbox') for *settlements* processing.~~

(b) FTP Batch Interface – MDPs can place MDMT and MTRD Meter Data Notification ~~*metering data* files~~messages directly into their 'Participant Inbox' directory on the AEMO network. This is the preferred option if an MDP has a large number of files to process. The process steps are described in the Batch handler section of the Introduction to MSATS.

(c) E-Hub API – MDPs can load MDMT Meter Data Notification using the AEMO B2M e-Hub Asynchronous Push-Push or Push-Pull APIs. This is a preferred option if the MDP has a large number of files to process. The process steps are described in the B2M e-Hub Technical Guide.

MDPs can load MTRD Meter Data Notification using the AEMO B2B e-Hub Asynchronous Push-Push or Push-Pull APIs. This is a preferred option if the MDP has a large number of files to process. The process steps are described in B2B SMP Technical Guide.

3.2 Security

To upload *metering data* via the MSATS ~~b~~Browser interface, the following rules must be adhered to:

The UserID identified in the SecurityContext element of the header must belong to the Participant ID.

(a) The UserID identified in the SecurityContext element in the XML message must be permitted to perform the batch transactions contained in the XML message (e.g. have been allocated a right that allows submission of MDM *metering data*).

(b) The Participant User submitting the transaction must have a right that allows full access to the 'Participant Mailbox' entity.

(c) The Participant ID must match the logged-on Participant User's Participant ID.

3.3 aseXML Message Format

To import *metering data* into MDM~~MSATS~~, the MDP systems must generate an XML-wrapped CSV file containing all relevant Metering and Participant information that conforms to the applicable aseXML Schema.

This aseXML document ~~file~~ must then be zipped and uploaded into MDM~~MSATS~~ using the MSATS ~~b~~Browser interface,~~ or by~~ directly placing the file into the appropriate 'Participant Inbox' on the AEMO file share, or sent as an aseXML payload through AEMO's market facing e-Hub APIs.

The aseXML document~~file~~ itself will contain 3 main sections:

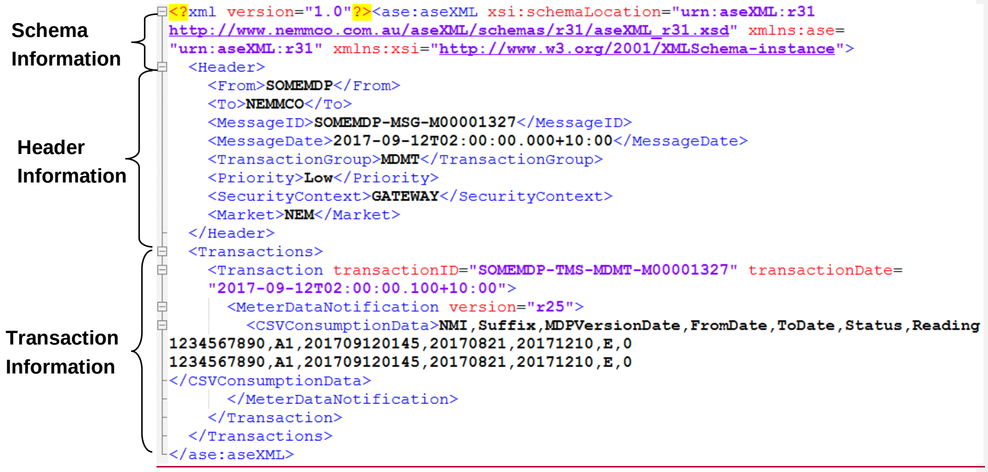
(a) Schema Information: This section details the MSATS schema version information and should not be modified unless AEMO releases an update.

(b) Header Information: This section contains information about the Participant submitting the file, its destination, and the type of transaction being submitted (MDMT or MTRD).

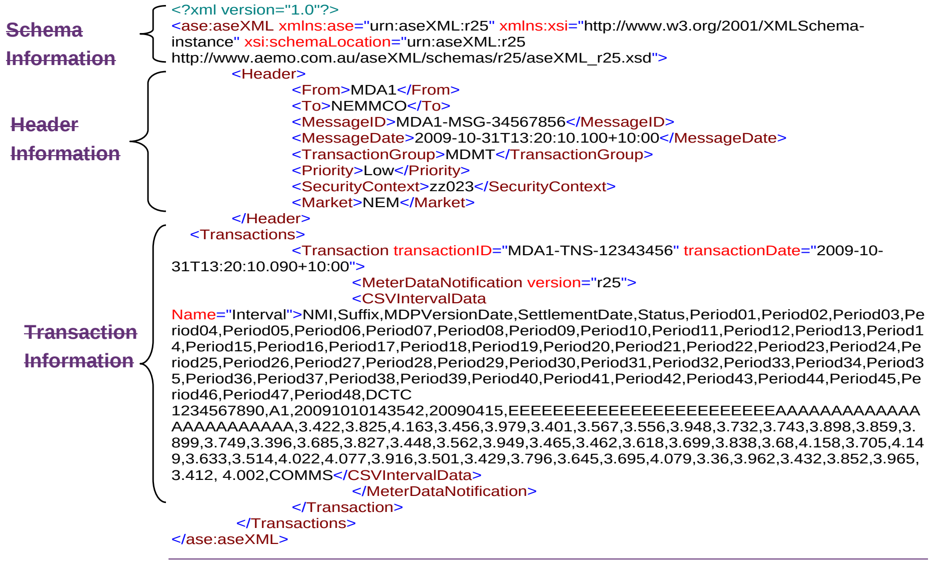
(c) Transaction Information: This section contains all of the transaction-specific information, i.e. the actual *metering data* to be loaded, depending on the transaction type, in either a CSV Meter Data Management Format (MDMF) or Meter Data File Format (MDFF).

This document has been developed using the current schema version. Whilst all endeavours will be made to keep this document up to date with the schema changes, the document needs to be read in conjunction with the latest schema information available on the AEMO website.

Figure 1 is an example of an aseXML file for the MDMT transaction type containing ~~the~~MDMF CSV payload data. The payload~~file~~ shows the structure of a transaction containing ~~interval~~*accumulated metering data*. Note the examples in this document refer to schema version r25, please refer to the latest version available from the AEMO website.



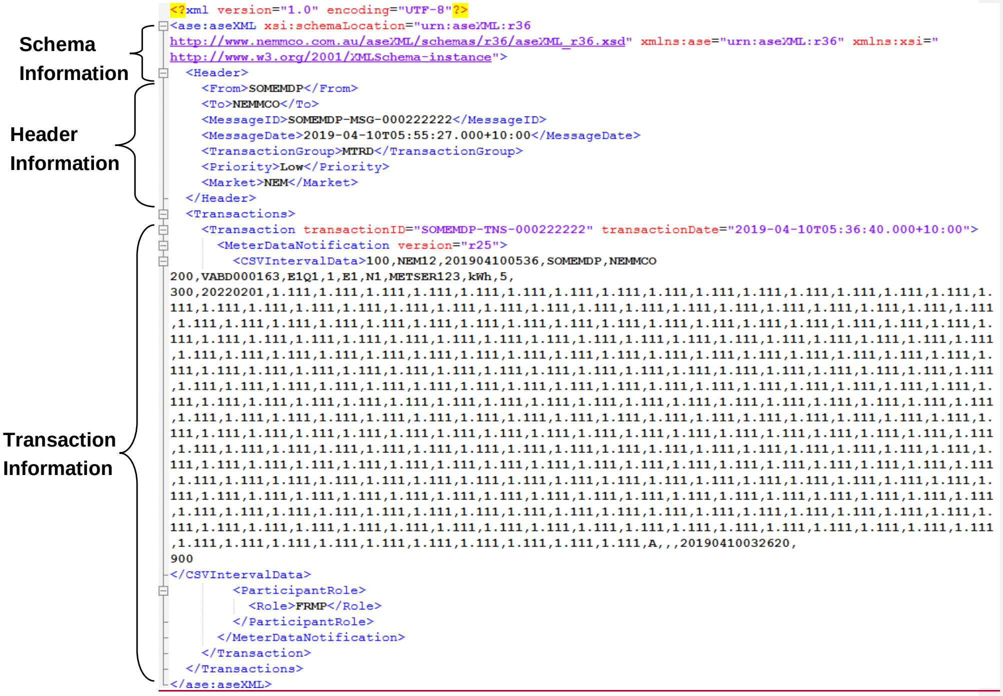
```
<?xml version="1.0"?><ase:aseXML xsi:schemaLocation="urn:aseXML:r31
http://www.nemmco.com.au/aseXML/schemas/r31/aseXML_r31.xsd" xmlns:ase=
"urn:aseXML:r31" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <From>SOMEMDP</From>
    <To>NEMMCO</To>
    <MessageID>SOMEMDP-MSG-M00001327</MessageID>
    <MessageDate>2017-09-12T02:00:00.000+10:00</MessageDate>
    <TransactionGroup>MDMT</TransactionGroup>
    <Priority>Low</Priority>
    <SecurityContext>GATEWAY</SecurityContext>
    <Market>NEM</Market>
  </Header>
  <Transactions>
    <Transaction transactionID="SOMEMDP-TMS-MDMT-M00001327" transactionDate=
    "2017-09-12T02:00:00.100+10:00">
      <MeterDataNotification version="r25">
        <CSVConsumptionData>NMI,Suffix,MDPVersionDate,FromDate,ToDate,Status,Reading
1234567890,A1,201709120145,20170821,20171210,E,0
1234567890,A1,201709120145,20170821,20171210,E,0
</CSVConsumptionData>
      </MeterDataNotification>
    </Transaction>
  </Transactions>
</ase:aseXML>
```



```
<?xml version="1.0"?>
<ase:aseXML xmlns:ase="urn:aseXML:r25" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:aseXML:r25
http://www.aemo.com.au/aseXML/schemas/r25/aseXML_r25.xsd">
        <Header>
                <From>MDA1</From>
                <To>NEMMCO</To>
                <MessageID>MDA1-MSG-34567856</MessageID>
                <MessageDate>2009-10-31T13:20:10.100+10:00</MessageDate>
                <TransactionGroup>MDMT</TransactionGroup>
                <Priority>Low</Priority>
                <SecurityContext>zz023</SecurityContext>
                <Market>NEM</Market>
        </Header>
   <Transactions>
                <Transaction transactionID="MDA1-TNS-12343456" transactionDate="2009-10-
31T13:20:10.090+10:00">
                        <MeterDataNotification version="r25">
                        <CSVIntervalData
Name="Interval">NMI,Suffix,MDPVersionDate,SettlementDate,Status,Period01,Period02,Period03,Pe
riod04,Period05,Period06,Period07,Period08,Period09,Period10,Period11,Period12,Period13,Period1
4,Period15,Period16,Period17,Period18,Period19,Period20,Period21,Period22,Period23,Period24,Pe
riod25,Period26,Period27,Period28,Period29,Period30,Period31,Period32,Period33,Period34,Period3
5,Period36,Period37,Period38,Period39,Period40,Period41,Period42,Period43,Period44,Period45,Pe
riod46,Period47,Period48,DCTC
1234567890,A1,20091010143542,20090415,EEEEEEEEEEEEEEEEEEEEEEEEAAAAAAAAAAAAA
AAAAAAAAAAA,3.422,3.825,4.163,3.456,3.979,3.401,3.567,3.556,3.948,3.732,3.743,3.898,3.859,3.
899,3.749,3.396,3.685,3.827,3.448,3.562,3.949,3.465,3.462,3.618,3.699,3.838,3.68,4.158,3.705,4.14
9,3.633,3.514,4.022,4.077,3.916,3.501,3.429,3.796,3.645,3.695,4.079,3.36,3.962,3.432,3.852,3.965,
3.412, 4.002,COMMS</CSVIntervalData>
                        </MeterDataNotification>
                </Transaction>
         </Transactions>
</ase:aseXML>
```

Figure 1 MDMT aseXML Message Format

Figure 2 is an example of an aseXML file for the MTRD transaction type containing MDFF CSV payload data. The payload shows the structure of a transaction containing *interval metering data* in the "NEM12" file format. Note the examples in this document refer to schema version r25, please refer to the latest version available from the AEMO website.



```
<?xml version="1.0" encoding="UTF-8"?>
<ase:aseXML xsi:schemaLocation="urn:aseXML:r36
http://www.nemmco.com.au/aseXML/schemas/r36/aseXML_r36.xsd" xmlns:ase="urn:aseXML:r36" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <From>SOMEMDP</From>
    <To>NEMMCO</To>
    <MessageID>SOMEMDP-MSG-000222222</MessageID>
    <MessageDate>2019-04-10T05:55:27.000+10:00</MessageDate>
    <TransactionGroup>MTRD</TransactionGroup>
    <Priority>Low</Priority>
    <Market>NEM</Market>
  </Header>
  <Transactions>
    <Transaction transactionID="SOMEMDP-TNS-000222222" transactionDate="2019-04-10T05:36:40.000+10:00">
      <MeterDataNotification version="r25">
        <CSVIntervalData>100,NEM12,201904100536,SOMEMDP,NEMMCO
200,VABD000163,E1Q1,1,E1,N1,METSER123,kWh,5,
300,20220201,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.
111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111
,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.
111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111
,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.
111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111
,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.
111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111
,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.
111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111
,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.
111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111
,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.
111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111
,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.
111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111
,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,1.111,A,,,20190410032620,
900
</CSVIntervalData>
        <ParticipantRole>
          <Role>FRMP</Role>
        </ParticipantRole>
      </MeterDataNotification>
    </Transaction>
  </Transactions>
</ase:aseXML>
```

Figure 2 MTRD aseXML Message Format

3.4 Header Information

Table 1 details the fields to be included in the 'Header' section of the XML-wrapped file. Take note of the 'Field Formats' to be used when creating the file. If a field entry is typed in the wrong case (i.e. not capitals), the file may be rejected by MDM~~MSATS~~.

Table 1 HEADER INFORMATION

Field Name	Field Format	Example
<Description>	A free-text field that will allow the input of up to 30 characters. It is the description of the MDP is submitting the file. Can be upper or lower case	'Testing and Certification Australia MDP'
<From>	The Participant ID of the Participant submitting the file (MDP), which must be in uppercase. The field will allow entry of up to 8 characters.	'TCAUSTM'
<To>	The AEMO Participant ID, which must be typed in uppercase.	'NEMMCO'

Field Name	Field Format	Example
<MessageID>	A unique, Participant-generated identifier for the file being sent, which can contain up to 50 characters.	'SOMEMDP-MSG-M00001327'~~'MDA1-MSG-2466453'~~ 'TCAUSTM017746632'
<MessageDate>	Date and time identifier to be attached to the file, in the following format: yyyy-mm-ddThh:mm:ss.sss+10:00	'2019-04-10T05:55:27.000+10:00'~~2001-12-31T08:59:11+10:00~~
<TransactionGroup>	Identifies the type of transaction being processed. The specifics of which transaction group is to be used are detailed in the next section.~~When uploading *metering data* the transaction group will always be 'MDMT'. It is a 4- character string.~~	'MDMT' 'MTRD'
<Priority>	Identifies the priority of the transaction. In turn, this priority will determine the order in which transactions are processed.	'Low'
<SecurityContext>	Identifies the 'User ID' of the Participant User submitting the file.	'USER1' 'JSMITH'
<Market>		'NEM'

Figure 3~~2~~ is an example of the 'Header' section of an XML file:

```
<Header>
  <From>SOMEMDP</From>
  <To>NEMMCO</To>
  <MessageID>SOMEMDP-MSG-000222222</MessageID>
  <MessageDate>2019-04-10T05:55:27.000+10:00</MessageDate>
  <TransactionGroup>MTRD</TransactionGroup>
  <Priority>Low</Priority>
  <Market>NEM</Market>
</Header>
```

```
<Header>
            <From>MDA1</From>
            <To>NEMMCO</To>
            <MessageID>MDA1-MSG-34567856</MessageID>
            <MessageDate>2009-10-31T13:20:10.100+10:00</MessageDate>
            <TransactionGroup>MDMT</TransactionGroup>
            <Priority>Low</Priority>
            <SecurityContext>zz023</SecurityContext>
            <Market>NEM</Market>
      </Header>
```

~~Figure 2~~Figure 3 XML Header Example

3.5 Metering Data Transaction Types

As an Initiator of Meter Data Notifications, MDPs can send *metering data* as either a MTRD or MDMT transaction type. Table 2 describes which transaction type should be sent based on the Participant and Meter type.

Table 2 Metering data transaction

	MDMT Transaction Type	MTRD Transaction Type
Transaction Group	MDM	MTRD
Transaction Type	Meter Data Notification	Meter Data Notification
Initiator	Meter Data Provider	Meter Data Provider
Recipient	• AEMO	• AEMO • Entitled Participates (e.g. Retailer or Distributor or Meter Data Provider or Meter Coordinator)
Meter Type	• Accumulation	• Interval • Accumulation
Delivery Point	• MSATS Browser – using the 'Data Load Import' > 'Participant Inbox' screen • FTP Batch – using the '/Inbox' file share directory • B2M Push API – using the B2M API message resource. • B2M Pull API – using the B2M API message resource	• MSATS Browser – using the 'B2B Browser > 'Upload File' screen • FTP Batch – using the '/Inbox' or 'B2B/Inbox' file share directory • B2B Push API – using the B2M API message resource • B2B Pull API – using the B2M API message resource.

3.6 Transaction Information

Table 3~~2~~ details the fields to be included in the <'Transaction'> section of the XML-wrapped file. Take note of the 'Field Formats' to be used when entering actual metering data

~~Table 2~~Table 3 Transaction information

Field Name	Field Format	Example
<TransactionI~~i~~d>	A free-text field that allows the input of up to 50 characters. It is a unique identifier assigned by the Participant to the transaction.	'SOMEMDP-TNS-000222222'~~'MDA1-TNS-1887373'~~
<TransactionDate>	Date and time identifier to be attached to the transaction itself. This must be in the format: yyyy-mm-ddThh:mm:ss.sss+10:00	'2019-04-10T05:36:40.000+10:00'~~2001-12-31T08:59:11+10:00~~
<MeterDataNotificationV~~v~~ersion>	The schema version number that is currently in use.	'r25'

Field Name	Field Format	Example
CSVDataType	The Participant may upload *interval metering data, accumulation metering data* or *profile* data. The prefix for each of these data types will be one of these: • <CSVIntervalData> • <CSVConsumptionData> • <CSVProfileData> (sample *meter* data) CSV_Data_Types must be delivered in separate transactions. Optionally this field can contain the name attribute as per below: • <CSVIntervalData Name='Interval'> • <CSVConsumptionData Name ='Consumption'> • <CSVProfileData Name = 'Profile'> Specific files captured for each of these CSV Data_Types are detailed in the next section.	<CSVConsumptionData> <CSVIntervalData> <CSVProfileData> Optional Examples: <CSVIntervalData Name='Interval'> <CSVConsumptionData Name='Consumption'> <CSVProfileData Name='Profile'>

3.7 CSV Data Types

For the provision of *metering data*, MDPs must use "standalone" files held as CSV data in a file format defined in accordance with the transaction and meter type. Table 4 details the file format for an accumulation or interval meter to be included in each transaction type.

Table 4 CSV File Format by Transaction Type

	MDMT Transaction Type	MTRD Transaction Type
Data Type	*NMI* Datastream (e.g. N1, N2 for *interval metering data*, 11 for *accumulated metering data*). Deliver Validated nett *metering data* and consumption including any Substitutions and Estimations.	*NMI* Datastream (e.g. E1, B1 for *interval metering data* and 11 for *accumulated metering data*). Deliver Validated register level *metering data* including any Substitutions and Estimations.
CSV File Format	Meter Data Management Format (MDMF) as per the format described in Section 3.8 of this document.	Meter Data File Format (MDFF) as described in the MDFF Specification NEM12 NEM13.
CSVConsumptionData Data Type	As per the format described in Section 3.8.1 of this document.	NEM13 format as described in the MDFF Specification NEM12 NEM13 Section 5.
CSVIntervalData Data Type	Not Supported	NEM12 format as described in the MDFF Specification NEM12 NEM13 Section 4.

3.8 Meter Data Management Format (MDMF)

MDMF is a CSV meter data format through which MDPs can submit *metering data* to AEMO for the purpose of *market settlements*. The format supports *metering data* delivered for accumulation meters through the CSVConsumptionData data type that must form the transaction body of a MDMT transaction. As MDMF requires that MDPs nett the active energy import and export *metering data*, this format supports a signed reading. The specifications for the MDMF format are described within this document.

3.8.1 MDMF CSV~~-~~Consumption~~-~~Data

CSVConsumptionData is used when loading MDMF formatted *metering data* for Datastreams listed in MSATS with aDatastream type of 'C' (consumption) which is for a basic/type 6 *metering installation*. The *metering data* has not been profiled to into TIs.~~Meter Readings are not yet broken down into the 1/2 hourly intervals (as with CSVIntervalData)~~.

Each component of the CSVConsumptionData listed in Table 5~~3~~ should be separated by a comma in the XML file.

For CSVConsumptionData delivered to AEMO~~ (MSATS)~~, the suffix detail must conform to the NMI Procedure for consumption *metering data*. The suffix identifier provided in the MDMF CSVConsumptionData file must be identical to the datastream value entered into the MSATS CATS_NMI_DataStream table for the *connection point* identified by the *NMI*.

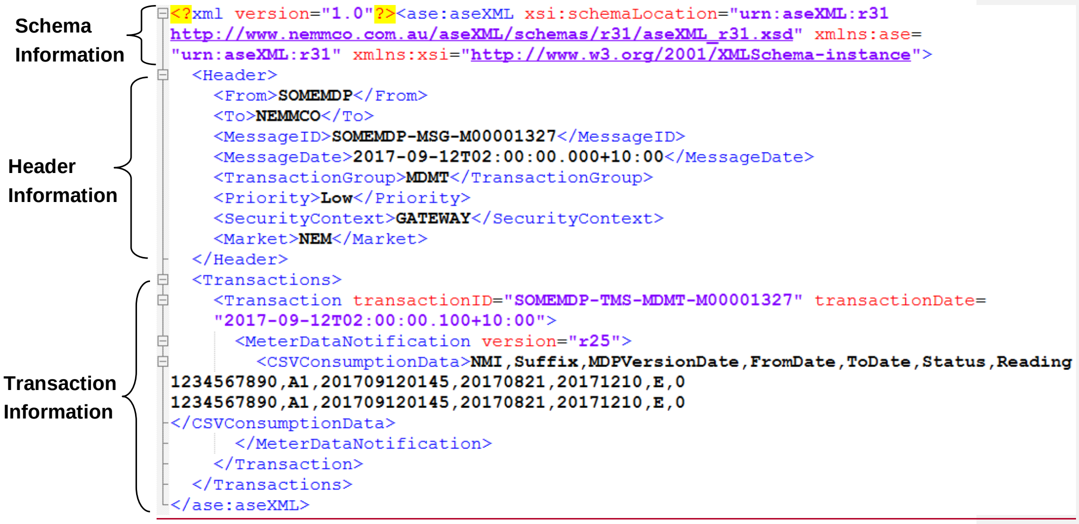
~~Table 3~~Table 5 MDMF CSV ConsumptionData~~Summary of Data Delivery~~

	~~**DELIVERY TO ENTITLED PARTICIPANT**~~ ~~e.g. LNSP, LR, FRMP~~	~~DELIVERY TO AEMO~~
~~Data Type~~	~~*NMI* Datastream (e.g. 11, 42)~~ ~~Deliver Validated *metering data* readings and consumption including any Substitutions and Estimations.~~	~~*NMI* Datastream (e.g. 11, 42)~~ ~~Deliver Validated *metering data* including any Substitutions and Estimations.~~
~~File Format~~	~~MDFF~~	~~'MDM'~~ ~~AEMO aseXML data file format.~~
~~Delivery Point~~	~~To Participants via *B2B e-Hub* inbox~~	~~To the MDP's MSATS inbox~~

Field Name	Field Format	Example
NMI	The *NMI*, which identifies the *connection point*. It consists of 10 alphanumeric characters.	'8166755454' 'VSSSS00001'
Suffix	The NMI suffix. This is the suffix for the Datastream as defined in the MSATS Procedures.	'11' '42'
MDPVersionDate	This is the date and time stamp the Participant system has assigned to the data record. It is the date and time the *metering data* was loaded into the MDP's system and must be in this format: yyyymmddhhmmss	'20010714083045'
FromDate	The first day of the Meter Reading period, in this format: yyyymmdd Time is taken to be at 00:00 hours on the start day of the Meter Reading period.	'20010501'

Field Name	Field Format	Example
ToDate	The last day of the reading period, in this format: yyyymmdd It is assumed to be taken at 23.59 hours.	'20010731'
Status	The quality flag of the metering data, which can be: A (Actual); E (Estimated); S (Substituted); or F (Final Substitutions) Refer Metrology Procedure: Part B for further detail on quality flags.	'A', 'E', 'S', or 'F'
Reading	This is the actual consumption value in kWh for the time period supplied (FromDate to the ToDate). In effect: Meter Reading at (ToDate – FromDate) = consumption reading. All values must be inclusive of *meter* multipliers, therefore all consumption values and Meter Readings issued are multiplier adjusted.	1398.667

Figure 4~~3~~ is an example of the ~~T~~transaction ~~I~~information of an aseXML MDMT payload~~file~~ used in the loading of MDMF~~consumption~~ *accumulation metering data*. Note the MDMF~~information~~ components included ~~next to~~ within the <CSVConsumptionData> tag~~section~~:



```
<?xml version="1.0"?><ase:aseXML xsi:schemaLocation="urn:aseXML:r31
http://www.nemmco.com.au/aseXML/schemas/r31/aseXML_r31.xsd" xmlns:ase=
"urn:aseXML:r31" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <From>SOMEMDP</From>
    <To>NEMMCO</To>
    <MessageID>SOMEMDP-MSG-M00001327</MessageID>
    <MessageDate>2017-09-12T02:00:00.000+10:00</MessageDate>
    <TransactionGroup>MDMT</TransactionGroup>
    <Priority>Low</Priority>
    <SecurityContext>GATEWAY</SecurityContext>
    <Market>NEM</Market>
  </Header>
  <Transactions>
    <Transaction transactionID="SOMEMDP-TMS-MDMT-M00001327" transactionDate=
    "2017-09-12T02:00:00.100+10:00">
      <MeterDataNotification version="r25">
        <CSVConsumptionData>NMI,Suffix,MDPVersionDate,FromDate,ToDate,Status,Reading
1234567890,A1,201709120145,20170821,20171210,E,0
1234567890,A1,201709120145,20170821,20171210,E,0
</CSVConsumptionData>
      </MeterDataNotification>
    </Transaction>
  </Transactions>
</ase:aseXML>
```

```
<?xml version="1.0" ?>
<ase:aseXML xmlns:ase="urn:aseXML:r25" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"    xsi:schemaLocation="urn:aseXML:r25
http://www.aemo.com.au/aseXML/schemas/r25/aseXML_r25.xsd">
        <Header>
                <From>MDA1</From>
                <To>NEMMCO</To>
                <MessageID>MDA1-MSG-34567856</MessageID>
                <MessageDate>2009-10-31T13:20:10.100+10:00</MessageDate>
                 <TransactionGroup>MDMT</TransactionGroup>
                 <Priority>Low</Priority>
                 <SecurityContext>zz023</SecurityContext>
                 <Market>NEM</Market>
        </Header>
        <Transactions>
                 <Transaction transactionID="MDA1-TNS-12343456" transactionDate="2009-10-
31T13:20:10.090+10:00">
                        <MeterDataNotification version="r25">

        <CSVConsumptionData>NMI,Suffix,MDPVersionDate,FromDate,ToDate,Status,Reading
1234567890,A1,20091010143542,20090415,20090714,E,3.245
1234567890,A2,20091010143542,20090415,20090714,A,.446</CSVConsumptionData>
                        </MeterDataNotification>
                </Transaction>
   </Transactions>
</ase:aseXML>
```

~~Figure 3~~Figure 4 XML Transaction Information

~~1.1 CSVIntervalData~~

~~CSVIntervalData is used when loading *metering data* for Datastreams listed in MSATS with a type of 'I' (Interval) which is for the following *metering installations*:~~

- ~~COMMS 1 to 4 (type 1 through to 4);~~
- ~~COMMS4D (whole current *metering installations* that meet the *minimum services specfication*;~~
- ~~COMMS4C (CT connected *metering installations* that meet the *minimum services specification);*~~
- ~~MRAM (*small customer metering installations* – Type 4A), VICAMI;~~
- ~~MRIM (Manually Read Interval Meter or type 5);~~
- ~~SAMPLE;~~
- ~~PROF; and~~
- ~~UMCP (unmetered supply or type 7).~~

~~The Meter Readings will be broken down into 48 intervals of 30 minute-data.~~

~~Each component of the CSVIntervalData listed in Table 3 should be separated by commas in the XML file.~~

~~For CSVIntervalData delivered to MSATS, the suffix detail must conform with the NMI Procedure for *interval metering data*. The Suffix value provided in the MDM CSVIntervalData file must be identical to the Datastream value entered into the MSATS CATS_NMI_DataStream table for the *NMI*. The suffix identifier for *interval metering data* (e.g. N1) is a net value for the contributing import and export *interval metering data* flows for the Interval Meter concerned. The net value for CSVIntervalData delivered to MSATS, being as follows:~~

~~(a) Where the *metering data* is in sub-intervals of 30 minutes, the *metering data* must be aggregated to 30-minute intervals before delivery; and~~

~~(b) Where the *metering data* comprises separate export and import Datastreams, the respective export and import intervals must be aggregated E – B flows to provide the NET 'N' value. (Note: the net *energy* for a *Generator* is generally negative).~~

~~**Table 4 Summary of Data Delivery**~~

	~~**DELIVERY TO ENTITLED *PARTICIPANT***~~ ~~I.E. LNSP, NSP2, LR, FRMP~~	~~DELIVERY TO *AEMO*~~
~~Data Type~~	~~*NMI* Datastream (e.g. E1, B1)~~ ~~Deliver Validated *interval metering data* including any Substitutions and Estimations.~~	~~NET value *NMI* Datastream (e.g. N1 = E1 – B1)~~ ~~Deliver Validated *metering data* including any Substitutions and Estimations as net *energy* aggregated to 30 minutes.~~
~~File Format~~	~~MDFF~~	~~'MDM'~~ ~~AEMO aseXML data file format.~~
~~Delivery Point~~	~~To the Registered Participants via *B2B e-Hub inbox*~~	~~To the MDP's MSATS inbox~~

~~**Table 5 CSVIntervalData**~~

~~Field Name~~	~~Field Format~~	~~Example~~
~~NMI~~	~~The *NMI*, which identifies the *connection point*. It consists of 10 alphanumeric characters.~~	~~8105157686~~ ~~8166755454~~ ~~VSSSS00001~~
~~Suffix~~	~~The *NMI* suffix. This is the suffix for the Datastream as defined in CATS (section 4.11.2).~~	~~N1~~ ~~N2~~
~~MDPVersionDate~~	~~This is the date and time stamp the Participant system assigned to the data record.~~ ~~It is the date and time the *metering data* was loaded into the MDP's system, which must be in this format:~~ ~~yyyymmddhhmmss~~	~~20010714083045~~
~~SettlementDate~~	~~The date the reading relates to ie, the Meter Reading date, which must be in this format:~~ ~~Yyyymmdd~~	~~20010724~~
~~Status~~	~~Status of the Meter Reading. A status will need to be included for each 30-minute interval value (therefore there could be 48 'A' characters included for each Datastream).~~ ~~Valid values are:~~ ~~A (Actual);~~ ~~E (Estimated);~~ ~~S (Substituted); or~~ ~~F (Final Substitutions)~~ ~~Refer Metrology Procedure: Part B for further detail on quality flags.~~	~~AAAAAAAAAAAAAAAAAAA~~ ~~AAAAAAAAAAAAAAAAAAA~~ ~~AAAAAAAAAA~~ ~~Or~~ ~~AAAAAASSSSSSSSAAAAA~~ ~~AAAAAAAAAAAAAAAAAAA~~ ~~AAAAFFFFFAA~~

~~Field Name~~	~~Field Format~~	~~Example~~
~~Period 1 – 48~~	~~The consumption (in kWh) for each of the 30-minute interval Meter Reading periods. Each consumption record must be separated with a comma.~~	~~P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,P11,P12,P13,P14,P15,P16,P17,P18,P19,P20,P21,P22,P23,P24,P25,P26,P27,P28,P29,P30,P31,P32,P33,P34,P35,P36,P37,P38,P39,P40,P41,P42,P43,P44,P45,P46,P47,P48,~~
~~DCTC[1]~~	~~Populate the Data Collection Type Code in accordance with section 3.9.~~	~~COMMS, COMMS4D, COMMS4C, MRIM, PROF, SAMPLE, MRAM VICAMI, UMCP.~~

~~Figure 4 is an example of the transaction information for a XML file used in the loading of *interval metering data*. Note the information components included next to the <CSVIntervalData> section:~~

```
<?xml version="1.0"?>
<ase:aseXML xmlns:ase="urn:aseXML:r25" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:aseXML:r25
http://www.aemo.com.au/aseXML/schemas/r25/aseXML_r25.xsd">
            <Header>
                        <From>MDA1</From>
                        <To>NEMMCO</To>
                        <MessageID>MDA1-MSG-34567856</MessageID>
                        <MessageDate>2009-10-31T13:20:10.100+10:00</MessageDate>
                        <TransactionGroup>MDMT</TransactionGroup>
                        <Priority>Low</Priority>
                        <SecurityContext>zz023</SecurityContext>
                        <Market>NEM</Market>
            </Header>
    <Transactions>
                        <Transaction transactionID="MDA1-TNS-12343456" transactionDate="2009-10-
31T13:20:10.090+10:00">
                                    <MeterDataNotification version="r25">
                                    <CSVIntervalData
Name="Interval">NMI,Suffix,MDPVersionDate,SettlementDate,Status,Period01,Period02,Period03,Pe
riod04,Period05,Period06,Period07,Period08,Period09,Period10,Period11,Period12,Period13,Period1
4,Period15,Period16,Period17,Period18,Period19,Period20,Period21,Period22,Period23,Period24,Pe
riod25,Period26,Period27,Period28,Period29,Period30,Period31,Period32,Period33,Period34,Period3
5,Period36,Period37,Period38,Period39,Period40,Period41,Period42,Period43,Period44,Period45,Pe
riod46,Period47,Period48,DCTC
1234567890,A1,20091010143542,20090415,EEEEEEEEEEEEEEEEEEEEEEEEAAAAAAAAAAAAA
AAAAAAAAAAA,3.422,3.825,4.163,3.456,3.979,3.401,3.567,3.556,3.948,3.732,3.743,3.898,3.859,3.
899,3.749,3.396,3.685,3.827,3.448,3.562,3.949,3.465,3.462,3.618,3.699,3.838,3.68,4.158,3.705,4.14
9,3.633,3.514,4.022,4.077,3.916,3.501,3.429,3.796,3.645,3.695,4.079,3.36,3.962,3.432,3.852,3.965,
3.412, 4.002,COMMS</CSVIntervalData>
                                    </MeterDataNotification>
                        </Transaction>
             </Transactions>
</ase:aseXML>
```

~~[1] Data Collection Type Code.~~

~~Figure 4 XML Transaction information~~

~~1.2 CSV Profile Data~~

~~CSVProfileData differs from CSVIntervalData in the aseXML transaction information, which reads <CSVProfileData>, rather than <CSVIntervalData>. CSVProfileData is currently used for sample *meters* (Metering Installation Type Codes of Sample or COMMS4) to supply *profile* data for the Controlled Load profile calculation process in the relevant Jurisdictions.~~

~~1.3 Data Collection Type Codes~~

~~The Data Collection Type Codes (DCTC) referred to in Table 4 are the Metering Installation Type Code in all but one case. The only exception is where a Metering Installation Type Code for a *metering installation* is a COMMS1, COMMS2, COMMS3 or COMMS4. The equivalent DCTC is COMMS. See table 5.~~

~~Table 6 DCTC Code – Metering Installation Type Code Mapping~~

~~DCTC Code~~	~~Metering Installation Type Code~~
~~COMMS~~	~~COMMS1, COMMS2, COMMS3, COMMS4~~
~~COMMS4D~~	~~COMMS4D~~
~~COMMS4C~~	~~COMMS4C~~
~~MRIM~~	~~MRIM~~
~~PROF~~	~~PROF~~
~~SAMPLE~~	~~SAMPLE~~
~~MRAM~~	~~MRAM~~
~~VICAMI~~	~~VICAMI~~
~~UMCP~~	~~UMCP~~

~~Rules governing the use DCTC codes are:~~

~~(a) It must be a maximum of eight characters in length.~~

~~(b) MSATS will not validate the DCTC code against previous file history for the *NMI*.~~

~~**Valid Combinations of Schema and Transaction Versions**~~

~~Schema Version~~	~~Transaction Version~~	~~Interval~~	~~Profile~~	~~Consumption~~
~~r25~~	~~r25~~	~~DCTC field required~~	~~DCTC field required~~	~~no DCTC field required~~

```
<MeterDataNotification version="r25">
                              <CSVIntervalData
Name="Interval">NMI,Suffix,MDPVersionDate,SettlementDate,Status,Period01,Period02,Period03,Pe
riod04,Period05,Period06,Period07,Period08,Period09,Period10,Period11,Period12,Period13,Period1
4,Period15,Period16,Period17,Period18,Period19,Period20,Period21,Period22,Period23,Period24,Pe
riod25,Period26,Period27,Period28,Period29,Period30,Period31,Period32,Period33,Period34,Period3
5,Period36,Period37,Period38,Period39,Period40,Period41,Period42,Period43,Period44,Period45,Pe
riod46,Period47,Period48,DCTC
1234567890,A1,20091010143542,20090415,EEEEEEEEEEEEEEEEEEEEEEEEAAAAAAAAAAAAA
AAAAAAAAAAA,3.422,3.825,4.163,3.456,3.979,3.401,3.567,3.556,3.948,3.732,3.743,3.898,3.859,3.
899,3.749,3.396,3.685,3.827,3.448,3.562,3.949,3.465,3.462,3.618,3.699,3.838,3.68,4.158,3.705,4.14
9,3.633,3.514,4.022,4.077,3.916,3.501,3.429,3.796,3.645,3.695,4.079,3.36,3.962,3.432,3.852,3.965,
3.412, 4.002,COMMS</CSVIntervalData>
```

~~Example (MDM) Metering Data File with DCTC~~

3.9 Meter Data File Format (MDFF)

MDFF is a CSV meter data format through which MDPs can submit register level *metering data* to AEMO for the purpose of *market settlements*. The format supports *metering data* delivered for both accumulation and interval meters where *interval metering data* is "NEM12" through the CSVIntervalData data type and *accumulated metering data* is "NEM13" through the CSVConsumptionData data type within the transaction body of a MTRD transaction. As MDFF requires that MDPs provide active and reactive energy as separate register level import and export *metering data*, this format does not support a signed reading. The specifications for the MDFF format are described within MDFF Specification NEM12 NEM13.

3.9.1 MDFF CSVConsumptionData

MDFF CSVConsumptionData should be delivered in the NEM13 format as described in the MDFF Specification NEM12 NEM13 Section 5.

3.9.2 MDFF CSVInterval

MDFF CSVIntervalData should be delivered in the NEM12 format as described in the MDFF Specification NEM12 NEM13 Section 4.

3.10 File Format

For MDPs Initiating transactions through AEMO's FTP messaging protocol, ~~O~~once all the information in the aseXML file is correct, it must be saved and converted into a .zip file. The .zip file must have a name that conforms to the following standard.

~~Table 7~~Table 6 Zip File Format

Field Name	Definition~~Field Format~~	Format~~Example~~
Transaction ~~Group~~	In the case of MDMT transactions, the transaction group will be 'MDMT~~mdmt~~'. In the case of MTRD transactions, the transaction will be'MTRD'.	4 Alphanumeric

Field Name	Definition~~Field Format~~	Format~~Example~~

Priority	h = High
	m = Medium
	l = Low
	Messages within each priority group are processed in last modified order.
	The priority for *metering data* transactions is l.
		1 Character

Unique ID	Unique ID that can be generated by Participant systems. The first part may be the Participant ID.	30 Alphanumeric characters
Extension	The Data file (XML-wrapped CSV file) should be saved as .zip file. The .zip extension is the only extension recognised by AEMO~~MSATS~~. Once recognised the batch handler will pick up and process the file.	3 characters

An example of a file name (once zipped) is shown below:

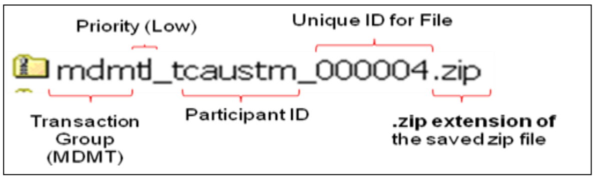


Figure 5 MDMT Filename Example

3.11 File Size

Participants must ensure that Messages containing aseXML Transactions do not exceed an uncompressed Message size of 10 MB for MTRD and 10 MB for MDMT

The number of Transactions within a single MTRD or MDMT transaction must not exceed 1000.

3.12 API common header parameter

For MDPs Initiating transactions through AEMO’s e-Hub B2B or B2M API messaging protocol, once all the information in the aseXML payload is correct, a messageContextID must be provided for the messages exchange. The participant/e-Hub uses the contextID of the original request when delivering its corresponding acknowledgements.

- For MDMT the format of the messageContextID is described in the B2M e-Hub Technical Guide.
- For MDMF the format of the messageContextID is described in the B2B SMP Technical Guide.

~~1.4 Data Load of Metering Data Files via the Browser (Interactive Upload)~~

~~Once set up with an appropriate MSATS login ID and password, an MDP will be able to import *metering data* using the ‘File Upload’ screen (or ‘Participant Inbox’).~~

~~The browser interface should also be considered as an emergency option for delivery of *metering data* if the batch processing system fails.~~

~~The browser interface allows an MDP to interactively manage its file transfer activities. MDPs will be able to view, upload, and delete files from their 'Participant Inbox', and read any acknowledgments from their 'Participant Outbox'.~~

~~Important Note:~~

~~To use the browser interface to deliver *metering data* to MSATS, MDPs must be set up with the following access rights in the system:~~

~~(a) Active Participant ID~~

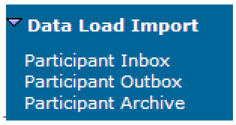
~~(b) Full access to the 'Participant Mailbox' entity ('Inbox', 'Outbox' and 'Archive')~~

~~(c) Active User ID and password~~

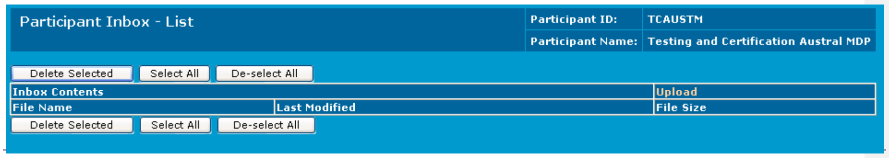
~~(d) Ensure that the UserID identified in the SecurityContext has been assigned a right that will allow access to the appropriate MSATS batch procedures.~~

~~To import *metering data* interactively using the browser interface:~~

1. ~~Log onto the MSATS browser with appropriate access rights.~~
2. ~~From the 'Data Load Import' menu select the 'Participant Inbox' sub-menu option on the MSATS menu bar.~~



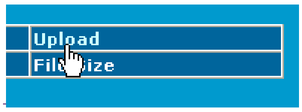
~~The following screen will display in the main window:~~


Participant Inbox - List

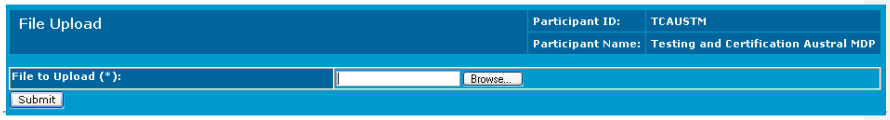
Participant ID: TCAUSTM

Participant Name: Testing and Certification Austral MDP

3. ~~To import data into MSATS click on the 'Upload' hyperlink above the 'File Size' column.~~



4. ~~The following screen will display in the main window.~~


File Upload

Participant ID: TCAUSTM

Participant Name: Testing and Certification Austral MDP

File to Upload (*):

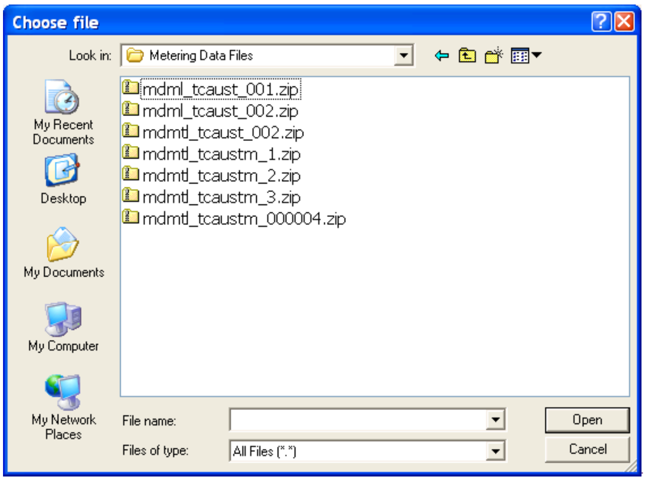
~~Note: File Upload~~

~~(a) Using this screen the MDP can select one of their data files to import into MSATS.~~

~~(b) The MDP can either enter the path to the file directly in the text field, or use the 'Browse' button to open the standard navigator type interface.~~

~~5. To select a file to import click on the~~ Browse... ~~button. A 'Choose File' window should now appear.~~

~~6. The 'Choose File' window will allow the user to browse through their local network to select the relevant file to import.~~



Look in: Metering Data Files

mdml_tcaust_001.zip
mdml_tcaust_002.zip
mdmtl_tcaust_002.zip
mdmtl_tcaustm_1.zip
mdmtl_tcaustm_2.zip
mdmtl_tcaustm_3.zip
mdmtl_tcaustm_000004.zip

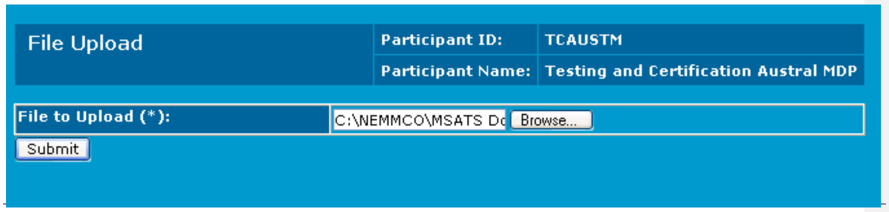
Files of type: All Files (*.*)

~~Note:~~

~~(a) Once you have found the file you wish to import, highlight it in the window and click on the~~ Open ~~button. (Alternatively you can double-click on the file to attach.)~~

~~(b) Ensure you select the zipped version of your data file. The MSATS batch handlers that pick up and process these files will only identify those files with a '.zip' extension.~~

~~7. The 'File Upload' screen will now appear, and the 'File to Upload' field will be populated with the location details of the file you just selected.~~



File Upload

Participant ID: TCAUSTM

Participant Name: Testing and Certification Austral MDP

File to Upload (*): C:\NEMMCO\MSATS Do

~~8. To import the file click on the~~ Submit ~~button. MSATS will now attempt to upload the file.~~

~~Note:~~

~~(a) The time taken to upload a file will vary depending on the size of the file, and the speed of the connection.~~

~~(b) When uploading via the browser interface the size of the file to be uploaded (before compression into a .zip file) is limited to a maximum of 1MB.~~

~~(c) The action of clicking 'Submit' will place the .zip file into the 'Participant Inbox' directory.~~

~~(d) MSATS will then process the .zip file and create an acknowledgment file in the Participant Outbox.~~

~~9. During this initial upload, MSATS will perform a first level validation on the aseXML file. MSATS will ensure that the following information is correct:~~

~~(a) The UserID nominated in the SecurityContext element of the message is permitted to perform the type of batch transaction being submitted (MDM *metering data* batch entity).~~

~~(b) The XML is well formed (i.e. that it meets the rules for writing XML).~~

~~(c) The file is valid according to the rules specified in the aseXML schema.~~

~~(d) The schema and transaction versions are supported by MSATS.~~

~~(e) The TransactionID has not previously been submitted.~~

~~(f) The file size does not exceed the 1MB unzipped limit.~~

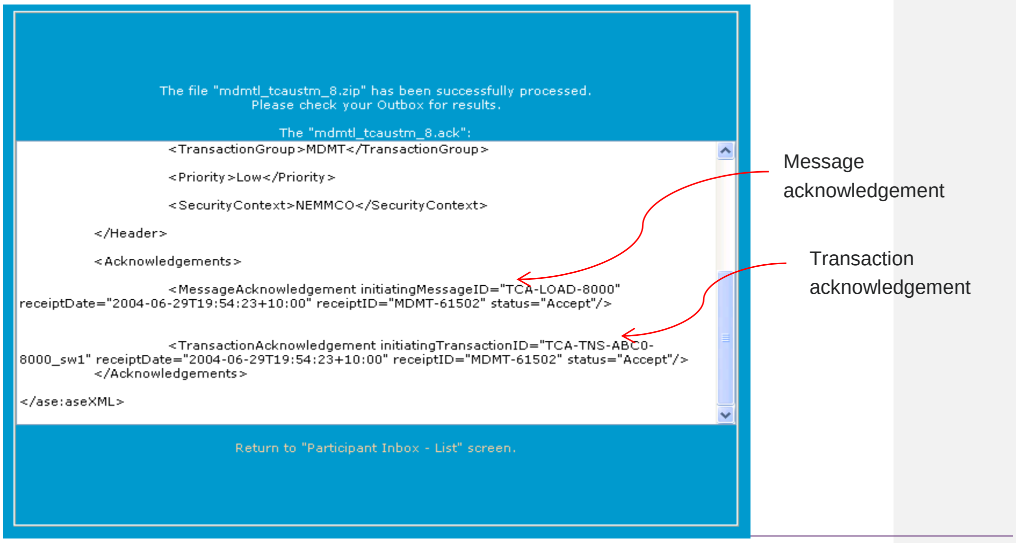
~~10. After completing the first level validation, MSATS will display the submitted information and the results of the validation in the form of an acknowledgement or .ack file.~~

~~Note:~~

~~(a) If the validation can be completed immediately, the message is displayed on the screen.~~

~~(b) If MSATS can load the data in the message, the acknowledgment screen will display an XML transaction that acknowledges the status of the message as a whole and of the transaction in the message.~~

~~(c) An example of the acknowledgement is as follows:~~



The file "mdmtl_tcaustm_8.zip" has been successfully processed.
Please check your Outbox for results.

The "mdmtl_tcaustm_8.ack":

```
                    <TransactionGroup>MDMT</TransactionGroup>

                    <Priority>Low</Priority>

                    <SecurityContext>NEMMCO</SecurityContext>

          </Header>

          <Acknowledgements>

                    <MessageAcknowledgement initiatingMessageID="TCA-LOAD-8000"
receiptDate="2004-06-29T19:54:23+10:00" receiptID="MDMT-61502" status="Accept"/>


                    <TransactionAcknowledgement initiatingTransactionID="TCA-TNS-ABC0-
8000_sw1" receiptDate="2004-06-29T19:54:23+10:00" receiptID="MDMT-61502" status="Accept"/>
          </Acknowledgements>

</ase:aseXML>
```

Return to "Participant Inbox - List" screen.

~~(d) That there is both a message acknowledgement and a transaction acknowledgement.~~

~~(e) There is only ever one message acknowledgement per batch file.~~

~~(f) Depending on the number of transactions in the message there could be multiple transaction acknowledgements.~~

~~(g) For this transaction group, there will only be one transaction in a message, but the message can consist of many rows.~~

~~(h) A transaction can contain consumption or interval .csv data, but not both.~~

~~(i) Messages and transactions that pass the validation have a status of 'Accept'. Messages and transactions that fail the validations have a status of 'Reject'.~~

~~(j) It is possible for the message to be accepted but not the transaction. This would be the case if, for example, the message is well formed, the header details are correct but the nominated user did not have the rights to perform this specific transaction within the Transactions element~~

~~(k) Once the .zip file message (data file) has been uploaded it will remain in the 'Participant Inbox' until MSATS has processed it. Once processed, an .ack file will be placed in the corresponding 'Participant Outbox'. Once the .ack file has been read and processed by the MDP's system, the original .zip file will need to be deleted from the 'Participant Inbox' following which MSATS will delete the .ack file.~~

~~11. If *metering data* that was successfully loaded from the .xml file (i.e. the transaction had an 'Accept' message), the .csv data will undergo a second level functional validation. Once this is complete MSATS will generate a message containing a 'Meter Data Response' transaction and place it into the MDP's 'Participant Outbox' in a .zip file. The second level validation consists of the following:~~

~~(a) the MDP submitting the data is correct based on the MDP of record in MSATS for all *NMIs* and all intervals and periods of *metering data* submitted;~~

~~(b) the *NMI* Datastream, as identified by the suffix has a Datastream Status Code of 'A' (Active) for the period of data provided; and~~

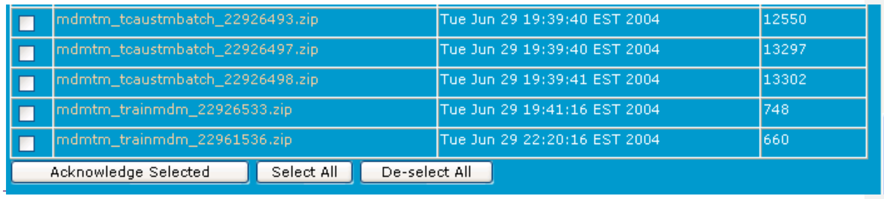
~~(c) that the start and end dates of the data record being validated do not overlap any existing records in such a way that the new record would replace only part of the period covered by an existing record. (See section 3 on validation).~~

~~Note:~~
~~If a consumption data file contains multiple Meter Readings for the same *NMI* and suffix they will be validated to see if they would form a 'virtual single read' or 'meta-read'. The file is checked to see if all data periods considered together will represent a continuous period of time – i.e. one Meter Reading's end date must be the day before the next one's start date). Multiple Meter Readings that form a 'meta-read' are, for the remainder of the validations, treated as a single Meter Reading. If the 'meta-read' fails any of the other validations, the group of Meter Readings are not loaded. Each of the individual Meter Readings, however, are then validated independently and loaded if they pass the validation. If the 'meta-read' passes all of the other validations, each of the individual Meter Readings that make the virtual Meter Reading will be loaded as separate records.~~

~~12. Navigate to the 'Participant Outbox – List' screen.~~

~~There is one 'Meter Data Response' in the form of an .xml message in a .zip file for each 'Meter Data Notification' transaction in the original file. Given that MSATS only allows one 'Meter Data Notification' transaction per file, this means that there will be one 'Meter Data Response' file for each file that was loaded. Each response file will appear in the Participant Outbox. The filename of the response appears in the format of:~~

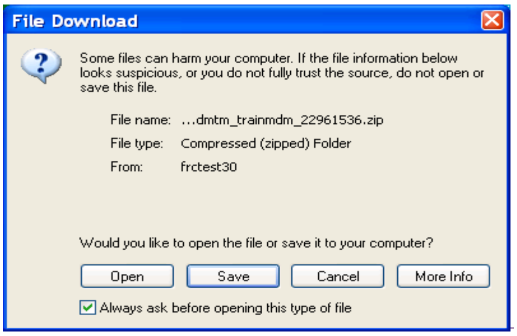
~~(a) Transaction Group & Priority = mdmtl~~
~~(b) Underscore = _~~
~~(c) User ID = <participantid>batch~~
~~(d) Underscore = _~~
~~(e) Unique Message ID = e.g. 22899696~~



□	mdmtm_tcaustmbatch_22926493.zip	Tue Jun 29 19:39:40 EST 2004	12550
□	mdmtm_tcaustmbatch_22926497.zip	Tue Jun 29 19:39:40 EST 2004	13297
□	mdmtm_tcaustmbatch_22926498.zip	Tue Jun 29 19:39:41 EST 2004	13302
□	mdmtm_trainmdm_22926533.zip	Tue Jun 29 19:41:16 EST 2004	748
□	mdmtm_trainmdm_22961536.zip	Tue Jun 29 22:20:16 EST 2004	660

~~Participants can then click on the ‘File Name’ hyperlink to view the ‘Meter Data Response’ file, which contains details of second level processing.~~

~~13. To view the ‘Meter Data Response’ message, click on the ‘File Name’ hyperlink. You may then be asked if you wish to ‘Open the file’ or ‘Save it to your computer’.~~



File Download

Some files can harm your computer. If the file information below looks suspicious, or you do not fully trust the source, do not open or save this file.

File name: …dmtm_trainmdm_22961536.zip
File type: Compressed (zipped) Folder
From: frctest30

Would you like to open the file or save it to your computer?

☑ Always ask before opening this type of file

~~Select the option that best suits the needs of your organisation. (This message may or may not appear depending on your Windows settings for opening files with a .zip extension.)~~

~~14. You should save the file if you wish to keep a permanent copy. However, you may want to open it first, the process which is described in these steps. Click the~~ Open ~~button to open the .zip file.~~

~~You will now be able to see the .xml file inside the .zip file.~~

~~15. Once the .zip file is opened (either immediately or later after you’ve downloaded it), you then need to open the .xml file contained in the .zip file so you can read its contents.~~

~~Depending on the application you use to open compressed files and your Windows settings, either single-click the file name (if it’s underlined) or double-click the file name to open it.~~

~~The file will be opened in whatever application you have associated with .xml files. In the examples in this documentation, it is Internet Explorer.~~

~~Figure 7 is an example of a ‘Meter Data Response’ message:~~

```
<?xml version="1.0" ?>
<ase:aseXML xmlns:ase="urn:aseXML:r25" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:aseXML:r25
http://www.aemo.com.au/aseXML/schemas/r25/aseXML_r25.xsd">
        <Header>
                <From>MDA1</From>
                <To>NEMMCO</To>
                <MessageID>MDA1-MSG-34567856</MessageID>
                <MessageDate>2009-10-31T13:20:10.100+10:00</MessageDate>
                <TransactionGroup>MDMT</TransactionGroup>
                <Priority>Medium</Priority>
                <SecurityContext>zz023</SecurityContext>
                <Market>NEM</Market>
        </Header>
        <Transactions>
                <Transaction transactionID="MDMT-TNS-12343456" transactionDate="2009-10-
31T13:20:10.090+10:00"
 initiatingTransactionID="MDA1-TNS-12343456">
                        <MeterDataResponse version="r6">
                        <ActivityID>67856</ActivityID>
                        <AcceptedCount>1</AcceptedCount>
                        <LoadDate>2009-11-29T19:52:50+10:00</LoadDate>
                        </MeterDataResponse>
                </Transaction>
        </Transactions>
</ase:aseXML>
```

~~**Figure 6 XML Response Message Format**~~

~~*Note: version'r6' is correct for this response report under the r25 schema*~~

~~1.5 Data Load of Metering Data Files via the Batch Process~~

~~MDPs can also deliver data files into MSATS using the batch process. This would involve placing the aseXML data file directly into the 'Participant Inbox' directory at the appropriate AEMO network location.~~

~~The information to be included in the aseXML file is identical to that detailed in section 3.3. To load data directly:~~

1. ~~Create the aseXML *metering data* file and save it as a zip file (ensure the file is saved according to AEMO aseXML standards and that the file is under 1MB before it is compressed into a .zip file).~~

2. ~~The file should be loaded into the appropriate 'Participant Inbox' with an extension of .tmp (this is to ensure the system does not attempt to process a partially loaded file).~~

3. ~~Once loaded rename the file to have its correct extension (i.e. change the name from XXX.tmp to XXX.zip).~~

4. ~~The batch handlers will detect this .zip file in the INBOX directory and perform a first level validation.~~

   (a) ~~MSATS will produce an acknowledgment file (.ack) and place it in the 'Outbox' directory.~~
   (b) ~~This .ack file will contain the results of the first level validation.~~
   (c) ~~Having received the .ack file, you need to delete the .zip file from your 'Inbox'.~~
   (d) ~~MSATS will detect that the .zip file has been deleted and delete the .ack file from the 'Outbox'.~~

~~5. Assuming that the acknowledgment indicated that the transaction passed the first level validations, the data loaded from the file will now undergo second level validation processing.~~

~~On completion of second level validation processing, a message containing a 'Meter Data Response' transaction (in a .zip file) will appear in the MDP's 'Participant Outbox' directory.~~

~~6. This file is identical to the one you would receive if you submitted the file containing the *metering data* using the browser. You can either:~~

~~(a) Follow the steps for 'Data Load of Metering Data Files via the Browser', beginning at step 12 to view and acknowledge the message using the browser; or~~

~~(b) Copy the file from the 'Participant Outbox' folder to your own system and then write an .ack file in your 'Participant Inbox' folder to acknowledge its receipt. MSATS will then delete the original.~~

4. METER DATA MESSAGING EXCHANGE

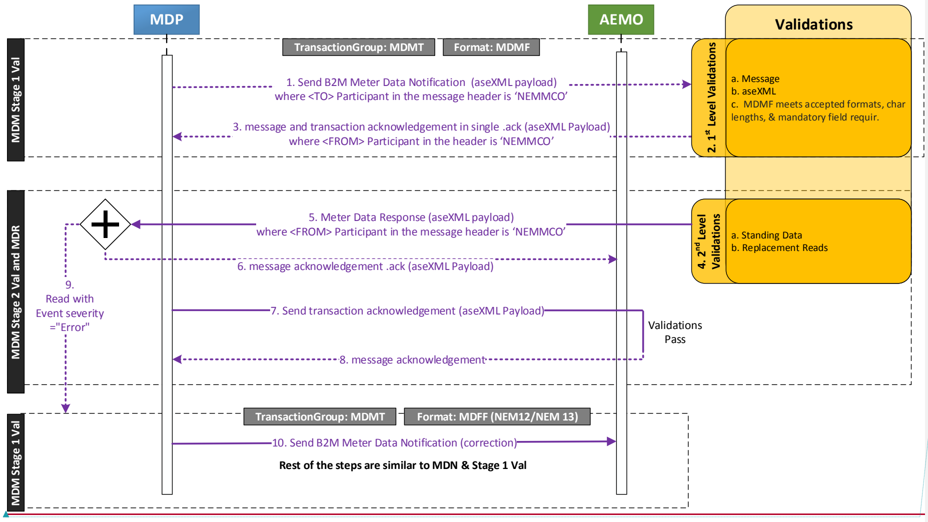
Metering data for the purpose of *market settlements* is to be exchanged between an MDP acting as the Initiator and AEMO as the ~~is a~~ Recipient. This exchange is through Interactions as Business Documents or Business Signals where:

(a) Business Documents are Notifications or Responses between an MDP and AEMO and contain important relevant business information.
(b) Business Signals are used to indicate the receipt, acceptance/rejection of a Business Document.
(c) Business Documents and Business Signals are mapped onto aseXML Transactions and Acknowledgements, respectively.

The B2M and B2B Procedures define the specifics of these Interactions and should be referred to in order to understand how a Participant undertakes Interactions with AEMO and other *Registered Participants*~~market Participants~~. The following section provides a high level example of the Interaction between MDPs and AEMO for the purpose of submitting m*etering data*.

4.1 MDMT Messaging Exchange

Figure 7 and Table 8 provide an illustration of the Interaction between an MDP and AEMO for the purpose of submitting a MDMT Meter Data Notification.


MDP
AEMO
Validations
MDM Stage 1 Val
TransactionGroup: MDMT
Format: MDMF
1. Send B2M Meter Data Notification (aseXML payload)
where <TO> Participant in the message header is 'NEMMCO'
3. message and transaction acknowledgement in single .ack (aseXML Payload)
where <FROM> Participant in the header is 'NEMMCO'
2. 1st Level Validations
a. Message
b. aseXML
c. MDMF meets accepted formats, char lengths, & mandatory field requir.
MDM Stage 2 Val and MDR
5. Meter Data Response (aseXML payload)
where <FROM> Participant in the message header is 'NEMMCO'
6. message acknowledgement .ack (aseXML Payload)
4. 2nd Level Validations
a. Standing Data
b. Replacement Reads
9.
Read with
Event severity
="Error"
7. Send transaction acknowledgement (aseXML Payload)
Validations
Pass
8. message acknowledgement
MDM Stage 1 Val
TransactionGroup: MDMT
Format: MDFF (NEM12/NEM 13)
10. Send B2M Meter Data Notification (correction)
Rest of the steps are similar to MDN & Stage 1 Val

Field Code Changed

Figure 6 MDMT Sequence Diagram

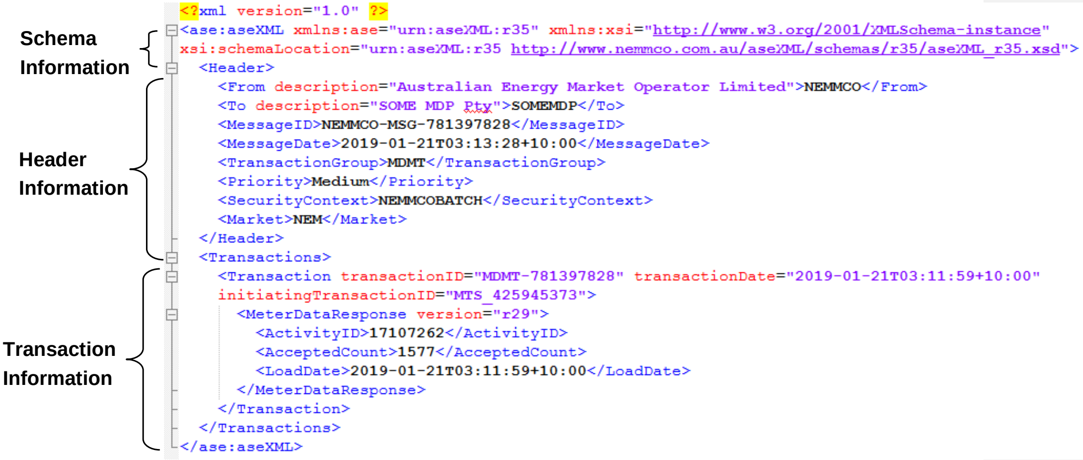
Table 7 MDMT Process Steps

Process Step ID	Process Step Definition	Process Step Type	Comments
Step 1	In the normal processing scenario, The Initiating MDP submits a MDMT Meter Data Notification message with *metering data* to AEMO for B2M *market settlements* via either of the following methods: • MSATS Browser – using the 'Data Load Import' > 'Participant Inbox' screen • Batch – using the '/Inbox' file share directory • B2M Push API – using the B2M API message resource. • B2M Pull API – using the B2M API message resource.	MDP / AEMO	<TO> Participant in the message header should be set to 'NEMMCO'
Step 2	AEMO ~~shall~~ performs level 1 validations of the Message, Transaction(s), format of the MDMF CSV Interval or Consumptions file, and the submitted meter data.	AEMO	The following types of validations are executed a. Message b. aseXML schema c. MDMF meets accepted formats, character lengths, & mandatory field requirements

Process Step ID	Process Step Definition	Process Step Type	Comments
Step 3	Where the Message passes validation then AEMO generates a positive *ase:MessageAcknowledgement* and continues with further processing of the aseXML Transaction(s) contained within the Message. If the Transaction(s) pass validation then AEMO generates positive *ase:TransactionAcknowledgements.*	MDP / AEMO	<FROM> Participant in the header of the Message and Transaction Acknowledgement shall be set to 'NEMMCO'. The Message and Transaction Acknowledgement(s) are bundled as a single Message Acknowledgement file or payload so the MDP isn't required to send Acknowledgements to close the hokey pokey messaging protocol.
Step 4	AEMO ~~shall~~ performs level 2 CATS Standing Data and replacement read validations of the meter data submitted within the MDMT Transaction(s).	AEMO	The following types of validations are executed a. CATS Standing Data b. Replacement Reads
Step 5	AEMO generates and sends to the Initiating MDP a MDMT Meter Data Response Message containing the results of the level 2 validations.	MDP / AEMO	<FROM> Participant in the header of the Message and Transaction Acknowledgements shall be set to 'NEMMCO'.
Step 6-8	The Initiating MDP validates the MDMT Meter Data Response and exchanges *ase:MessageAcknowledgement* and *ase:TransactionAcknowledgements* with AEMO as per the hokey pokey messaging protocol.	MDP / AEMO	
Step 9-10	Where the MDMT Meter Data Response generated by AEMO contains reads that were rejected by the AEMO level 2 validation process the MDP recreates the MDMT Meter Data Notification with the corrected reads.	MDP / AEMO	

4.2 MDMT Validation Response

Figure 8 is an example of a 'Meter Data Response' message.



```
<?xml version="1.0" ?>
<ase:aseXML xmlns:ase="urn:aseXML:r35" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:aseXML:r35 http://www.nemmco.com.au/aseXML/schemas/r35/aseXML_r35.xsd">
  <Header>
    <From description="Australian Energy Market Operator Limited">NEMMCO</From>
    <To description="SOME MDP Pty">SOMEMDP</To>
    <MessageID>NEMMCO-MSG-781397828</MessageID>
    <MessageDate>2019-01-21T03:13:28+10:00</MessageDate>
    <TransactionGroup>MDMT</TransactionGroup>
    <Priority>Medium</Priority>
    <SecurityContext>NEMMCOBATCH</SecurityContext>
    <Market>NEM</Market>
  </Header>
  <Transactions>
    <Transaction transactionID="MDMT-781397828" transactionDate="2019-01-21T03:11:59+10:00"
    initiatingTransactionID="MTS_425945373">
      <MeterDataResponse version="r29">
        <ActivityID>17107262</ActivityID>
        <AcceptedCount>1577</AcceptedCount>
        <LoadDate>2019-01-21T03:11:59+10:00</LoadDate>
      </MeterDataResponse>
    </Transaction>
  </Transactions>
</ase:aseXML>
```

Note: version'r29' is correct for this response report under the r35 schema

Figure 7 XML MDMT Validation Response Message

The 'ActivityID' is a unique ID assigned by AEMO~~MSATS~~ used for internal MDM processing.

The numeric part of the 'MessageID' and the 'TransactionID' matches the numeric part of the ~~.zip~~ file name for an FTP delivered message or messageContextID for an API delivered message.

The end of the first line in the 'Transactions' element contains the initiating 'TransactionID'. This is the 'TransactionID' that was supplied by the MDP in the MDMT 'MeterDataNotification' transaction. This identifier is the key piece of information for identifying which original transaction this response refers to.

The value in the 'AcceptedCount' element is the number of rows that were accepted (i.e. loaded) and 'LoadDate' is the date and time MDM~~MSATS~~ loaded the accepted data.

The code within the 'Event' element(s) contains the outcome of the validations.

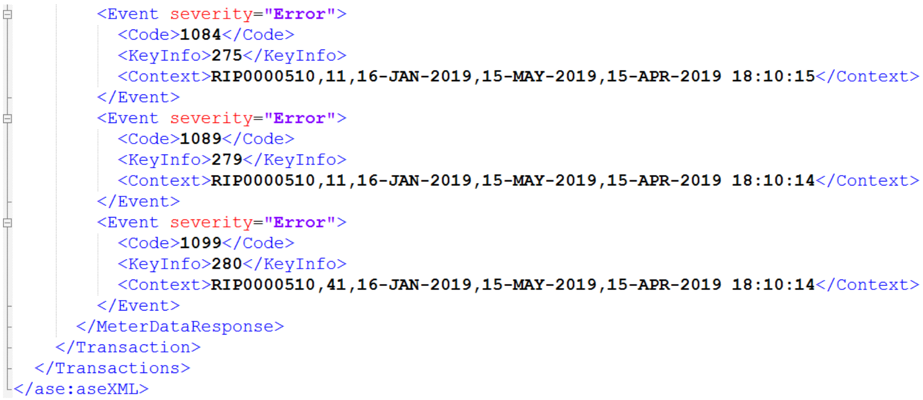
~~A code of '0' means that all of the data was successfully loaded.~~

If any errors are encountered, summary information about each failed Meter Reading is displayed in an 'Event' element (i.e. one 'Event' element for each failed Meter Reading). Thus, the 'AcceptedCount' plus the number of error events should equal the number of reads submitted.

In the example that follows, there are three~~two~~ rejected reads.

~~<Event severity='Error'>~~
~~<Code>1089</Code>~~
~~<KeyInfo>2</KeyInfo>~~
~~<Context>RIP0000510,11,26-SEP-02,30-DEC-02 11:22:39</Context>~~
~~</Event>~~
~~<Event severity='Error'>~~
~~<Code>1099</Code>~~
~~<KeyInfo>3</KeyInfo>~~
~~<Context>RIP0000510,11,04-OCT-02,10-OCT-02 11:22:39</Context>~~
~~</Event>~~

~~</MeterDataResponse>~~
~~</Transaction>~~
~~</Transactions>~~
~~</ase:aseXML>~~



```
        <Event severity="Error">
          <Code>1084</Code>
          <KeyInfo>275</KeyInfo>
          <Context>RIP0000510,11,16-JAN-2019,15-MAY-2019,15-APR-2019 18:10:15</Context>
        </Event>
        <Event severity="Error">
          <Code>1089</Code>
          <KeyInfo>279</KeyInfo>
          <Context>RIP0000510,11,16-JAN-2019,15-MAY-2019,15-APR-2019 18:10:14</Context>
        </Event>
        <Event severity="Error">
          <Code>1099</Code>
          <KeyInfo>280</KeyInfo>
          <Context>RIP0000510,41,16-JAN-2019,15-MAY-2019,15-APR-2019 18:10:14</Context>
        </Event>
      </MeterDataResponse>
    </Transaction>
  </Transactions>
</ase:aseXML>
```

Figure 8 MDMT Events

The four digit code in the 'Code' element is an error code.

- Error code 1084 represents the error 'Invalid NMI Suffix: Possible causes include invalid CSV data or invalid NMI Suffix for record.'
- Error code 1089 represents the error 'There is a record in the system that overlaps this record with a 'Version Date' that is after or equal to the 'Version Date' of this record.'
- Error code 1099 represents the error 'Read failed as part of Meta-read'.

A full list of error codes and descriptions are available in MSATS. The list of error codes is found under 'Administration/Codes Maintenance/Error Codes' or via a C1 report the on 'Error Codes' table.

The 'KeyInfo' element contains the row number. Note that the first row number that contains metering data is row 2. Row 1 contains the column headings.

The 'Context' element contains each of the following, separated by commas.

(a) NMI

(b) Datastream

(c) Start Date

~~(c)~~(d) End Date

~~(d)~~(e) MDP Version Date Time.

4.3 MTRD Message Exchange

Figure 10 and Table 9 provide an illustration of the Interaction between an MDP and AEMO for the purpose of submitting a MDMT Meter Data Notification.

Field Code Changed

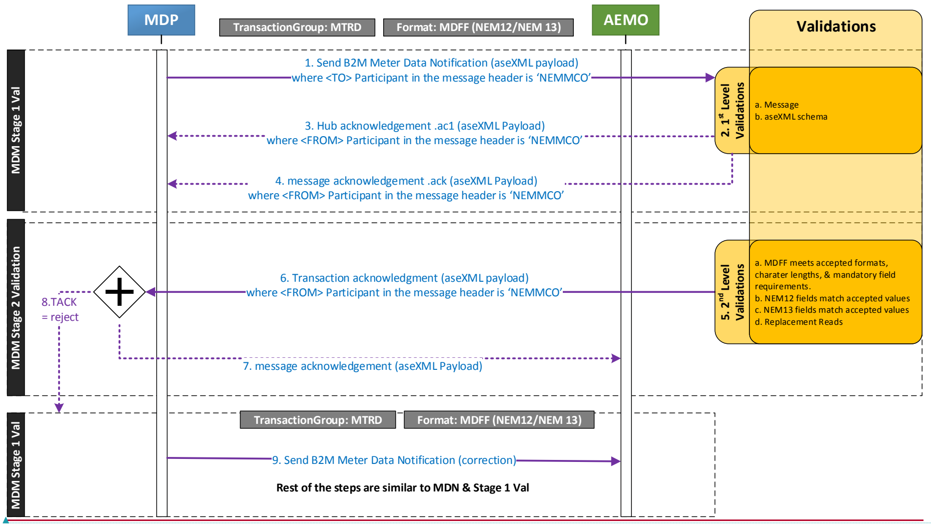

MDP
TransactionGroup: MTRD
Format: MDFF (NEM12/NEM 13)
AEMO
Validations
MDM Stage 1 Val
1. Send B2M Meter Data Notification (aseXML payload) where <TO> Participant in the message header is 'NEMMCO'
2. 1st Level Validations
a. Message
b. aseXML schema
3. Hub acknowledgement .ac1 (aseXML Payload) where <FROM> Participant in the message header is 'NEMMCO'
4. message acknowledgement .ack (aseXML Payload) where <FROM> Participant in the message header is 'NEMMCO'
MDM Stage 2 Validation
5. 2nd Level Validations
a. MDFF meets accepted formats, charater lengths, & mandatory field requirements.
b. NEM12 fields match accepted values
c. NEM13 fields match accepted values
d. Replacement Reads
6. Transaction acknowledgment (aseXML payload) where <FROM> Participant in the message header is 'NEMMCO'
8.TACK = reject
7. message acknowledgement (aseXML Payload)
MDM Stage 1 Val
TransactionGroup: MTRD
Format: MDFF (NEM12/NEM 13)
9. Send B2M Meter Data Notification (correction)
Rest of the steps are similar to MDN & Stage 1 Val

Figure 9 MTRD Sequence Diagram

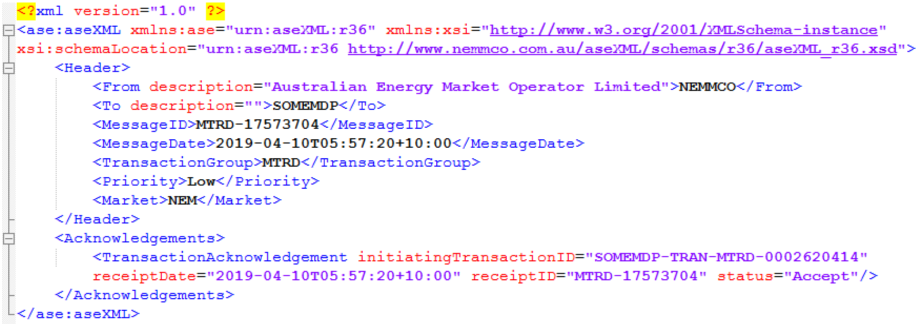
Table 8 MTRD Process Steps

Process Step ID	Process Step Definition	Process Step Type	Comments
Step 1	In the normal processing scenario, The Initiating MDP submits a B2M MTRD Meter Data Notification message with *metering data* to AEMO for B2M *market settlements* via either of the following methods: • MSATS Browser – using the 'B2B Browser > 'Upload File' screen • Batch – using the '/Inbox' or 'B2B/Inbox' file share directory • B2B Push API – using the B2M API message resource. • B2B Pull API – using the B2M API message resource.	MDP / AEMO	<TO> Participant in the message header should be set to 'NEMMCO'. MDFF Header record (100) ToParticipant should be set to 'NEMMCO'. The B2B Synch API is not available for B2M MTRD Meter Data Notification submissions.
Step 2	AEMO ~~shall~~ performs level 1 validations of the Message and Transaction(s).	AEMO	The following types of validations are executed • Message • aseXML Schema
Step 3	AEMO validates the Message, generates and sends to the Initiating MDP a Hub Acknowledgement.	MDP / AEMO	Definition of Hub Acknowledgement • FTP protocol - .ac1 • Webservice Protocol – Hub ack payload on API invocation

Process Step ID	Process Step Definition	Process Step Type	Comments
Step 4	Where the Message passes validation then AEMO generates a positive *ase:MessageAcknowledgement* and continues with further processing of the aseXML Transaction(s) contained within the Message.	MDP / AEMO	<FROM> Participant in the header of the Message Acknowledgement shall be set to 'NEMMCO'. Definition of Message Acknowledgement • FTP protocol - .ack • Webservice Protocol – ack payload
Step 5	AEMO performs level 2 validations of the MDFF and meter data submitted within the MTRD Transaction(s).	AEMO	The following types of validations are executed: • MDFF meets accepted formats, character lengths, & mandatory field requirements. • NEM12 fields match accepted values. • NEM13 fields match accepted values. • Replacement Reads.
Step 6	AEMO generates and sends *ase:TransactionAcknowledgements* to the Initiating MDP a containing the results of the level 2 validations.	MDP / AEMO	<FROM> Participant in the header of the Transaction Acknowledgement shall be set to 'NEMMCO'.
Step 7	The Initiating MDP validates *ase:TransactionAcknowledgements* and sends a *ase:MessageAcknowledgements* to AEMO as per the hokey pokey messaging protocol.	MDP / AEMO	
Step 8-9	Where the *ase:TransactionAcknowledgements* generated by AEMO is a reject or partial the MDP recreates the MTRD Meter Data Notification with the corrected reads.	MDP / AEMO	

4.4 MTRD Transaction Acknowledgement Validation Response

Figure 11 is an example of an *ase:TransactionAcknowledgements/@status="Accept"* for a MTRD transaction.



```
<?xml version="1.0" ?>
<ase:aseXML xmlns:ase="urn:aseXML:r36" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:aseXML:r36 http://www.nemmco.com.au/aseXML/schemas/r36/aseXML_r36.xsd">
    <Header>
        <From description="Australian Energy Market Operator Limited">NEMMCO</From>
        <To description="">SOMEMDP</To>
        <MessageID>MTRD-17573704</MessageID>
        <MessageDate>2019-04-10T05:57:20+10:00</MessageDate>
        <TransactionGroup>MTRD</TransactionGroup>
        <Priority>Low</Priority>
        <Market>NEM</Market>
    </Header>
    <Acknowledgements>
        <TransactionAcknowledgement initiatingTransactionID="SOMEMDP-TRAN-MTRD-0002620414"
        receiptDate="2019-04-10T05:57:20+10:00" receiptID="MTRD-17573704" status="Accept"/>
    </Acknowledgements>
</ase:aseXML>
```

Figure 10 XML MTRD Transaction Acknowledgment 'Accept' Validation Response Format

The 'recieptID' is a unique ID assigned by AEMO for internal MDM processing while 'receiptDate' is the date and time that the processing occurred.

The numeric part of the 'MessageID' and the 'TransactionID' matches the numeric part of the file name for an FTP delivered message or messageContextID for an API delivered message.

The end of the first line in the 'Transactions' element contains the initiating 'TransactionID'. This is the 'TransactionID' that was supplied by the MDP in the MTRD 'MeterDataNotification' transaction. This identifier is the key piece of information for identifying which original transaction the *ase:TransactionAcknowledgements* refers to.

[Note: The following section on 'Event' codes is under development with the 5MS Systems Working Group to determine the exact format of returned 'Events'. The description below details AEMO's recommended approach]

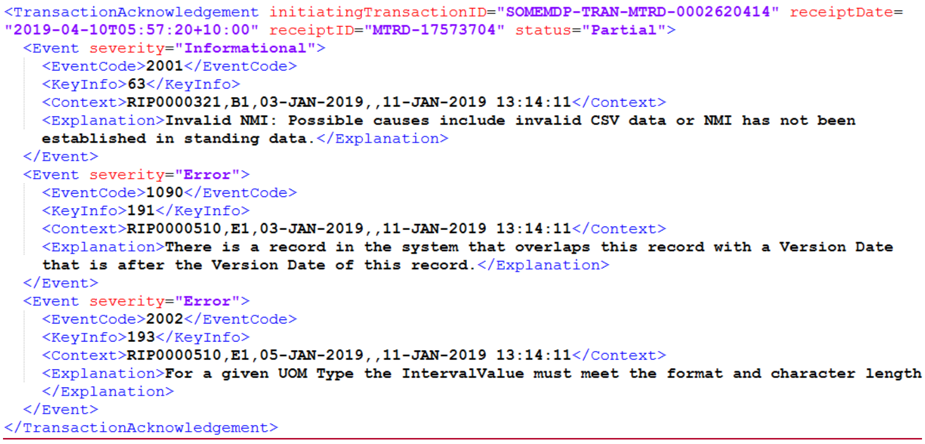
The code within the 'Event' element(s) contains the outcome of the validations. For MTRD transactions an 'Event' can have a severity of either 'Information' or 'Error'.

- An 'Event' severity of 'Information' is used to indicate that an AEMO validation has identified that key attributes of the Meter Reading cannot be validated against CATS Standing Data. While the Meter Reading will be loaded into AEMO's MDM, the Meter Reading will not be used for *market settlements* until the setup of CATS Standing Data has been completed. 'Information' events should be addressed by Participants without delay by submitting the required CATS Change Requests to setup the CATS Standing Data required to support the Meter Reading.

- An 'Event' severity of 'Error' is used to indicate that an AEMO validation has identified the Meter Reading cannot be accepted and will not be loaded into AEMO's MDM. 'Error' events should be addressed by the MDP without delay by correcting the MDFF (NEM12/NEM13) value(s) that caused the validation failure and resending the MTRD transaction to AEMO and all entitled Participants.

If any 'Error' or 'Information' events are encountered, summary information about each failed Meter Reading is displayed in an 'Event' element (i.e. one 'Event' element for each failed Meter Reading). Only rejected reads shall be displayed.

In the example that follows, there is an 'Information' read and two rejected 'Error' reads for a NEM12 MDFF file.



```
<TransactionAcknowledgement initiatingTransactionID="SOMEMDP-TRAN-MTRD-0002620414" receiptDate=
"2019-04-10T05:57:20+10:00" receiptID="MTRD-17573704" status="Partial">
  <Event severity="Informational">
    <EventCode>2001</EventCode>
    <KeyInfo>63</KeyInfo>
    <Context>RIP0000321,B1,03-JAN-2019,,11-JAN-2019 13:14:11</Context>
    <Explanation>Invalid NMI: Possible causes include invalid CSV data or NMI has not been
    established in standing data.</Explanation>
  </Event>
  <Event severity="Error">
    <EventCode>1090</EventCode>
    <KeyInfo>191</KeyInfo>
    <Context>RIP0000510,E1,03-JAN-2019,,11-JAN-2019 13:14:11</Context>
    <Explanation>There is a record in the system that overlaps this record with a Version Date
    that is after the Version Date of this record.</Explanation>
  </Event>
  <Event severity="Error">
    <EventCode>2002</EventCode>
    <KeyInfo>193</KeyInfo>
    <Context>RIP0000510,E1,05-JAN-2019,,11-JAN-2019 13:14:11</Context>
    <Explanation>For a given UOM Type the IntervalValue must meet the format and character length
    </Explanation>
  </Event>
</TransactionAcknowledgement>
```

Figure 11 MTRD Events

The four digit code in the 'EventCode' element is an error code.

- Event code 2001 [Actual MSATS event code TBA] represents an Informational warning 'Invalid NMI: Possible causes include invalid CSV data or NMI has not been established in CATS Standing Data.'
- Event code 1090 represents the error 'There is a record in the system that overlaps this record with a Version Date that is after the Version Date of this record.'
- Event code 2002 [Actual MSATS event code TBA] represents the error 'There is a record in the system that overlaps this record with a Version Date that is after the Version Date of this record.'

A full list of event codes and descriptions are available in MSATS. The list of error codes is found under 'Administration/Codes Maintenance/Error Codes' or via a C1 report the on 'Error Codes' table.

The 'KeyInfo' element contains the row number. Note that the first row number that contains metering data is row 2. Row 1 contains the column headings.

The 'Context' element contains each of the following, separated by commas.

(a) NMI

(b) Datastream

(c) Start Date

(d) End Date (accumulation meters only)

(e) Update Date Time

5. FILE VALIDATION

5.1 MDMT Validations~~Principles~~

Validation of *metering data*~~single Meter Readings~~ is based on alignment with overlapping start and end dates of existing records as well as on the MDP version date and time. Exceptions are where the existing overlapping Meter Reading is an Estimate.

~~Initial v~~Validations undertaken ~~in CATS~~ (before ~~it is accepted by the~~sending *metering data* to MDMs):

(a) Datastreams are active. For non-interval Datastreams, this includes for the duration of the *metering data*~~Meter Readings~~.

(b) The MDP must be the Current MDP on the 'To Date' for a non-interval Meter Reading and the 'Read Date' if it is interval/*profile* data.

(c) The TNI is assigned to a Profile Area.

(d) There is~~are~~ no duplicate *metering data*~~Meter Readings~~ within the input file (*NMI*, Datastream, start date, or Meter Reading date if interval date) If there is a duplicate record the first record is accepted and following records are rejected.

(e) Validation of start (1000 days from submit date) and end date (1000 days from submit date) for a non-interval Meter Reading. For an interval/*profile* Meter Reading the date must be no more than 1000 days before the submit date or 1000 days into the future.

5.1.1 MDMT Validation of MDPVersionDT

For interval and non-interval *metering data*~~Meter Readings~~:

(a) The load of any new *metering data* records into MDM~~MSATS~~, which are to replace existing *metering data* records for a *NMI*/Datastream, will be validated for 'MDPVersionDT' where the MDP Participant ID is the same for both Meter Readings supplied. The 'MDPVersionDT' for the new data record must be greater than the 'MDPVersionDT' of the existing record in MDM.

(b) The load of new data records into MDM~~MSATS~~ to replace existing *metering data* records for a *NMI*/Datastream where the existing data record was supplied by a different MDP Participant ID will be accepted. No validation is undertaken against 'MDPVersionDT' in this situation.

A separate error code exists where the 'MDPVersionDT' is the same – to distinguish from those where the 'MDPVersionDT' is less than the record existing in MDM.

For 'meta-reads', the maximum 'MDPVersionDT' of the 'meta-read' is used (i.e. the maximum of the 'MDPVersionDT' of all the individual records that make up the 'meta-read'). This is then compared with the maximum 'MDPVersionDT' of all of the overlapping existing records in MDM in the 'meta-read' start and end date range. This could allow some of the rows in the 'meta-read' to replace records in MDM that have a later 'MDPVersionDT', but as the incoming *metering data* file is created from the MDP's metering database, all the records in the incoming file should be the latest, if one or more of them has a later 'MDPVersionDT' than that exists in MDM.

5.1.2 MDMT Validation of Start and End dates of reads (Including meta-reads)

The process of the validation of start and end dates of Accumulation Meter Reading records in a single transaction will be as follows:

(a) Sort the *metering data*~~Meter Readings~~ in start date order.

(b) Create a 'meta-read', which consist of all records that align with each other, and use the start date and the end date of the 'meta-read' for validation (e.g. two Meter Readings – one 1/3 to 31/3, one 1/4 to 30/4, meta-read 1/3 to 30/4) (Overlapped new Meter Readings will not form 'meta-reads' since their dates do not align and will be validated as single Meter Readings, each in turn).

(c) The start and end dates of the 'meta-read' (which may be a single Meter Reading) must either align with existing valid current Meter Readings or fall in periods where there is no current Meter Reading – this allows the new 'meta-read' to fill in gaps in Meter Readings.

(d) Estimate Test: Existing Estimates (read type flag = 'E') are not considered in the start and end date validation process of new Meter Readings (regardless of whether the new Meter Reading is an Estimated Meter Reading).

(e) Datastream Inactive Test: Existing Meter Readings that span periods that the Datastream is now inactive (due to retrospective changes) are also not considered in the start and end date validation process of new Meter Readings (these existing Meter Readings are now effectively invalid due to the Meter Reading spanning an inactive period).

(f) If a 'meta-read' fails validation, each component Meter Reading of the 'meta-read' is considered separately with the above validations.

Note: the Estimate Test and the Datastream Inactive Test are performed against existing *Metering data*~~Meter Readings~~ to determine whether they are eligible for use in the start and end date tests. 'MDPVersionDT' validation is still undertaken.

5.1.3 MDMT~~MSATS~~ Data File Validations

All submitted MDMT 'Meter Data Notification' data files must pass the following MDMT~~MSATS~~ validations before they are accepted and loaded into MDM~~MSATS~~.

~~1.5.1 Interval Data~~

~~Table 8 Interval Data Information~~

~~Term~~	~~Description~~
~~Submitting MDP~~	~~MDP is the Current MDP for the Meter Reading date for that *NMI* in CATS.~~
~~NMI, NMI SUFFIX~~	~~*NMI* and *NMI* suffix must exist in CATS~~
~~Period (Readings)~~	~~48 numeric values within string (47 commas)~~ ~~String should contain no alpha characters (i.e. 0-9, ',' and '.' only)~~
	~~No blank fields and no double commas~~
~~Status~~	~~Length of string is 48. String can only contain 'A' – Actuals, 'S' – Substitute, 'F' – Final Substitute and 'E' – Estimates.~~
~~Settlement Date~~	~~Valid date format~~
~~MDPVersionDT~~	~~Valid date format~~

5.1.4 Consumption Data

Table 9 Consumption Data Information

Term	Description
All CSV Fields	No blank fields (no double commas) and valid date formats.
Submitting MDP	MDP is the Current MDP for that *NMI* in CATS ~~for~~on the 'To Date'~~Meter Reading period~~.
NMI, NMI SUFFIX	*NMI* and NMI suffix must exist in CATS
Reading	Only 1 value within string (0 commas) String should contain no alpha characters (i.e. 0-9, ',' and '.' only)
Status	Length of String is 1. Can only contain 'A' – Actuals, 'S' – Substitute, 'F' – Final Substitute and 'E' – Estimates.
From Date	Must be in valid date format and be between start date and end date of the NMI Suffix in CATS.
To Date	Must be in valid data format and be between start date and end date of the NMI Suffix in CATS.
	~~No blank fields and no double commas~~
~~MDP Version Date~~	~~Valid date format~~

5.2 MTRD Validations

Validation of *metering data* is based on alignment with overlapping start and end dates of existing records as well as on the MDP version date and time. Exceptions are where the existing overlapping Meter Reading is an Estimate.

Validations undertaken before *metering data* are accepted by MDM:

(a) Where the NMI exists in MSATS, the MDP must be the Current MDP on the 'To Date' for a non-interval Meter Reading and the 'Read Date' if it is interval/profile data.

(b) There is no duplicate metering data within the input file (NMI, Datastream, start date, or Meter Reading date if interval date) If there is a duplicate record the first record is accepted and following records are rejected.

(c) Validation of start (1000 days from submit date) and end date (1000 days from submit date) for a non-interval Meter Reading. For an interval/profile Meter Reading the date must be no more than 1000 days before the submit date or 1000 days into the future.

5.2.1 MTRD Validation of UpdateDateTime

For interval and non-interval *metering data*:

(a) The load of any new *metering data* records into MDM, which are to replace existing *metering data* records for a NMI/Datastream, will be validated for 'UpdateDateTime' where the MDP Participant ID is the same for both Meter Readings supplied. The 'UpdateDateTime' for the new data record must be greater than or equal to the 'UpdateDateTime' of the existing record in MDM.

(b) The load of new data records into MDM to replace existing *metering data* records for a NMI/Datastream where the existing data record was supplied by a different MDP Participant ID will be accepted. No validation is undertaken against 'UpdateDateTime' in this situation.

For 'meta-reads', the maximum 'UpdateDateTime' of the 'meta-read' is used (i.e. the maximum of the 'UpdateDateTime' of all the individual records that make up the 'meta-read'). This is then compared with the maximum 'UpdateDateTime' of all of the overlapping existing records in MDM in the 'meta-read' start and end date range. This could allow some of the rows in the 'meta-read' to replace records in MDM that have a later 'UpdateDateTime', but as the incoming *metering data* file is created from the MDP's metering database, all the records in the incoming file should be the latest, if one or more of them has a later 'UpdateDateTime' than that exists in MDM.

5.2.2 MTRD Validation of Start and End dates of reads (Including meta-reads)

The process of the validation of start and end dates of Accumulation Meter Reading records in a single transaction will be as follows:

(a) Sort the *metering data* in start date order.

(b) Create a 'meta-read', which consist of all records that align with each other, and use the start date and the end date of the 'meta-read' for validation (e.g. two Meter Readings – one 1/3 to 31/3, one 1/4 to 30/4, meta-read 1/3 to 30/4) (Overlapped new Meter Readings will not form 'meta-reads' since their dates do not align and will be validated as single Meter Readings, each in turn).

(c) The start and end dates of the 'meta-read' (which may be a single Meter Reading) must either align with existing valid current Meter Readings or fall in periods where there is no current Meter Reading – this allows the new 'meta-read' to fill in gaps in Meter Readings.

(d) Estimate Test: Existing Estimates (read type flag = 'E') are not considered in the start and end date validation process of new Meter Readings (regardless of whether the new Meter Reading is an Estimated Meter Reading).

(e) Datastream Inactive Test: Existing Meter Readings that span periods that the Datastream is now inactive (due to retrospective changes) are also not considered in the start and end date validation process of new Meter Readings (these existing Meter Readings are now effectively invalid due to the Meter Reading spanning an inactive period).

(f) If a 'meta-read' fails validation, each component Meter Reading of the 'meta-read' is considered separately with the above validations.

Note: the Estimate Test and the Datastream Inactive Test are performed against *metering data* to determine whether they are eligible for use in the start and end date tests. 'UpdateDateTime' validation is still undertaken.

5.2.3 MTRD Data Filer Validations

All submitted MTRD 'Meter Data Notification' data files must pass the following MDMT validations before they are accepted and loaded into MDM.

5.2.4 MTRD Interval Data

Table 10 Interval Data Information

Term	Description
All CSV Fields	The CSV fields in the 100-900 records meets accepted formats, character lengths, and mandatory field requirements as described in the MDFF Specification NEM12 NEM13.
Submitting MDP	Where a NMI has been setup in the CATS Standing Data, the MDP is the Current MDP for the Meter Reading date for that *NMI* in CATS.

Term	Description
UOM	• The UOM must be known, for a target UOM and for the UOM factor to be applied. • The UOM must match the allowed values in accordance with the MDFF Specification NEM12 NEM13 (appendix B).
IntervalLength	• IntervalLength is a supported Interval. • The IntervalDate must not be greater than 1000 days in the past or future.
IntervalValue	• For a given UOM Type the IntervalValue must meet the format and character length described in MDFF Specification NEM12 NEM13 (appendix B). • IntervalValue cannot be a null, negative or exponential value.
QualityMethod	• QualityMethod has the following format: QMM, where Quality Flag (Q) = 1st character and Method Flag (MM) = 2nd character. • The value of Quality Flag (the 1st character of QualityMethod) on the 300 record must match the Quality Flags described in MDFF Specification NEM12 NEM13 (appendix C). • Where the value of the Quality Flag (1st character of QualityMethod) on the 300 record is 'V' then validation is against the quality flag of QualityMethod populated on the 400 record. • The value of the Quality Flag (1st character of QualityMethod) used in the 400 record must match the Quality Flags described in MDFF Specification NEM12 NEM13 (appendix C). • Where the value of Quality Flag (the 1st character of QualityMethod) on the 300 or 400 record is not 'A' or 'N', then the Method Flag (2nd two characters of QualityMethod) are mandatory. • Where the Method Flag (2nd two characters of QualityMethod) is mandatory it must match the Method Flag described in MDFF Specification NEM12 NEM13 (appendix D). • Only one QualityMethod can be applied to an Interval.
Interval event record (400)	• An Interval event record (400) must immediately follow a 300 record where QualityFlag for the 300 record is 'V' or where the quality flag is 'A' and reason codes 79, 89, and 61 are used.
NMISuffix	• The first character of the NMISuffix ('E' from 'E1', 'B' from 'B1') must be known to determine the sign based on the National Metering Identifier Procedure (section 7).
ReasonCode	• ReasonCode on the 300 and ReasonCode on the 400 is mandatory where the QualityFlag is 'F' and 'S'. • ReasonCode on the 300 must not be populated if QualityFlag is 'V'. • ReasonCode on the 300 or 400 must match to one of the values described in the MDFF Specification NEM12 NEM13 (appendix E). • Only one ReasonCode can be applied to an Interval.
ReasonDescription	• ReasonDescription of ReasonCode on the 300 or 400 is mandatory where the ReasonCode is '0'.
TransCode	• Transaction Code (Record500) in the meter data file / message must match to one of the values described in the MDFF Specification NEM12 NEM13 (appendix A).

5.2.5 MTRD Consumption Data

Table 11 Consumption Data Information

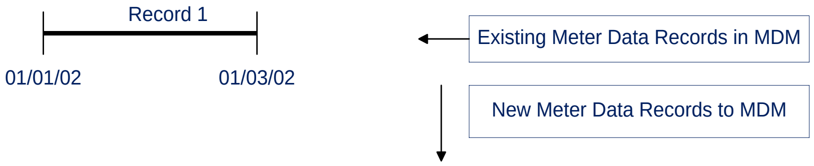
Term	Description
All CSV fields	• The CSV fields in the 100-900 records meets accepted formats, character lengths, and mandatory field requirements as described in the MDFF Specification NEM12 NEM13.

Term	Description
Submitting MDP	- Where a NMI has been setup in the CATS Standing Data, the MDP is the Current MDP for the Meter Reading date for that *NMI* in CATS.
UOM	- The UOM must be known, for a target UOM and for the UOM factor to be applied. - The UOM must match the allowed values in accordance with the MDFF Specification NEM12 NEM13 (appendix B).
PreviousRegisterReadDateTime & CurrentRegisterReadDateTime	The PreviousRegisterReadDateTime and CurrentRegisterReadDateTime must not be either greater than 1000 days in the past of future.
Quantity	- For a given UOM Type the Quantity must meet the format and character length described in MDFF Specification NEM12 NEM13 (appendix B). - Quantity cannot be a null or negative value.
PreviousQualityMethod	- PreviousQualityMethod has the following format: QMM, where Quality Flag (Q) = 1st character and Method Flag (MM) = 2nd character. - The value of the Quality Flag (1st character of PreviousQualityMethod) on the 250 record must match the Quality Flags described in MDFF Specification NEM12 NEM13 (appendix C). - Where the value of Quality Flag (the 1st character of PreviousQualityMethod) on the 250 record is not 'A' then the Method Flag (2nd two characters of PreviousQualityMethod) are mandatory. - Where the Method Flag (2nd two characters of PreviousQualityMethod) is mandatory it must match the Method Flag described in MDFF Specification NEM12 NEM13 (appendix D).
CurrentQualityMethod	- CurrentQualityMethod has the following format: QMM, where Quality Flag (Q) = 1st character and Method Flag (MM) = 2nd character. - The value of the Quality Flag (1st character of CurrentQualityMethod) on the 250 record must match the Quality Flags described in MDFF Specification NEM12 NEM13 (appendix C). - Where the value of Quality Flag (the 1st character of CurrentQualityMethod) on the 250 is not 'A', then the Method Flag (2nd two characters of CurrentQualityMethod) are mandatory. - Where the Method Flag (2nd two characters of CurrentQualityMethod) is mandatory it must match the Method Flag described in MDFF Specification NEM12 NEM13 (appendix D).
PreviousReasonCode & CurrentReasonCode	- *PreviousReasonCode* & *CurrentReasonCode* must match to one of the values described in the MDFF Specification NEM12 NEM13 (appendix E). - Description of *ReasonCode* for *PreviousRegisterRead* is Mandatory where the PreviousReasonCode is '0'. - Description of *ReasonCode* for *CurrentRegisterRead* is Mandatory where the CurrentReasonCode is '0'.
PreviousTransCode & CurrentTransCode	Transaction Code (Record550.PreviousTransCode or Record550.CurrentTransCode) in the meter data file / message must match to one of the values described in the MDFF Specification NEM12 NEM13 (appendix A).
DirectionIndicator	DirectionIndicator must be I or E.

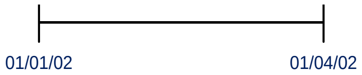
5.3 Consumption Meter Data Record date relationship examples

A. One existing *metering data* record loaded into MDM

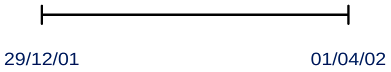
The following examples assume that the Datastream is active for the entire period.


Record 1
01/01/02
01/03/02
Existing Meter Data Records in MDM
New Meter Data Records to MDM

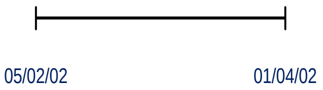
Scenarios in relation to Record Above



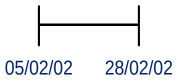
1. This record **will load**, - existing record archived to history.



2. This record **will load**, - existing record 1 archived to history. In this case start date of new record is before start date of existing record and the end date of new record is after the end date of the existing record.



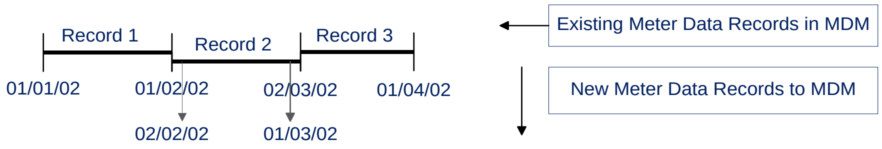
3. This record **will not load** unless record 1 is a forward estimate. In this case the start date of new record is after the start date of the existing record and therefore will fail the validation.



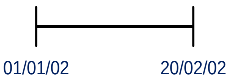
4. This record **will not load** unless record 1 is a forward estimate. In this example the start date of the new record is after the start date of existing record, the end date of the new record is before end date of existing record, and therefore will fail the validation.

B. Three existing *metering data* records loaded into MDM

The following examples assume that the Datastream is active for the entire period.

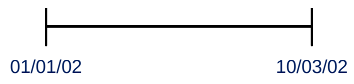


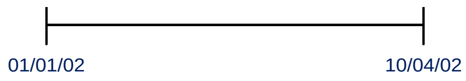
Scenarios in relation to Records Above



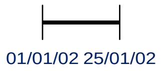
5. This record **will not load** unless record 2 is a forward estimate; - there is no date continuity to record 3. Data gap would result 21/2/02 – 1/3/02

6. This record **will only load** if record 3 is a forward estimate (Record 3’s read_type_flag = ‘E’). If record 3 is forward estimate, then new record replaces previous records 1,2 and 3.

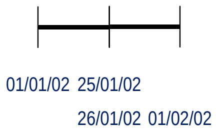




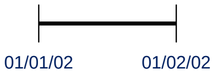
7. This record **will load**, - replaces previous records 1, 2 and 3.



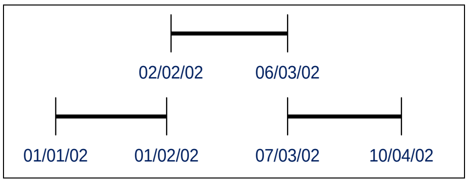
8. This record **will not load** unless record 1 is a forward estimate; - there is no date alignment with Record 1. Data gap would result 26/1/02 – 1/2/02



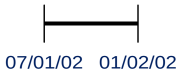
9. These records **will load**. If the 'meta-read' validation fails, each record of the 'meta-read' will be validated separately – these 2 will fail as they do not align with existing records (unless record 1 is an Estimate – then both records will load).



10. This record **will load**, - replaces previous record 1, if the version date is greater than existing record.

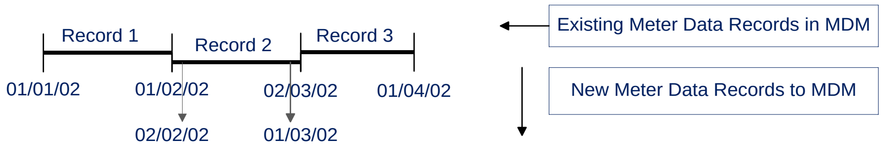


11. All of these records **will load**. New records will replace existing records 1,2 and 3. If the 'meta-read' validation fails, each record of the 'meta-read' will be validated separately – record 1 will load if it passes the 'MDPVersionDT' test, records 2 and 3 will fail as they do not align with existing records (unless records 2 & 3 are Estimates).

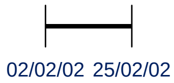


12. This record will NOT load unless record 1 is an Estimate – there is no start date alignment with record 1

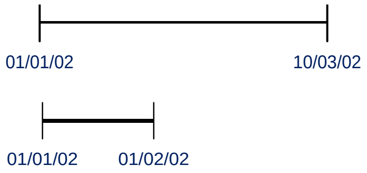
C. Overlapping Meter Readings in a transaction



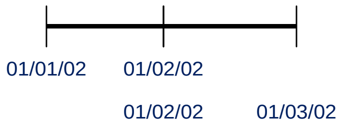
Scenarios in relation to Existing Records Above



13. This record **will NOT load** unless record 2 is an Estimate – there is no data continuity to replace record 2.
Data gap would result 26/2/02 – 1/3/02.



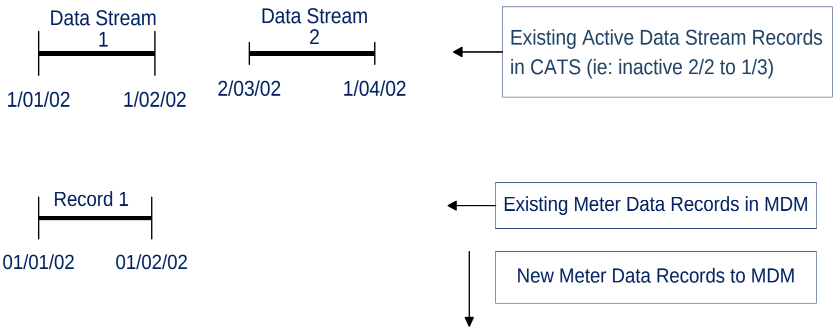
14. This first record **will only load** if record 3 is an Estimate (read_type_flag = 'E') and its 'MDPVersionDT' is greater than the max of 'MDPVersionDT' of the first 2 existing records. Record 2 will **not** load as it is a duplicate record in the file, (see Section 3.1).



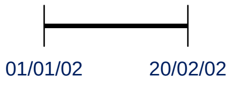
15. The first record **will load,** and the second record **will not load,** unless the first new read is an Estimate and has an MDP Version ID that is less than the second record. Start and end dates are INCLUSIVE, meaning that the start of a subsequent record must be 1 day after the end date of the previous record.

D. One existing *metering data* record loaded into MDM (example 1)

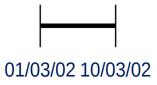
The following examples assume that the Datastream is not active for the entire period.


Data Stream 1
1/01/02
1/02/02
Data Stream 2
2/03/02
1/04/02
Existing Active Data Stream Records in CATS (ie: inactive 2/2 to 1/3)
Record 1
01/01/02
01/02/02
Existing Meter Data Records in MDM
New Meter Data Records to MDM

Scenarios in relation to Records Above



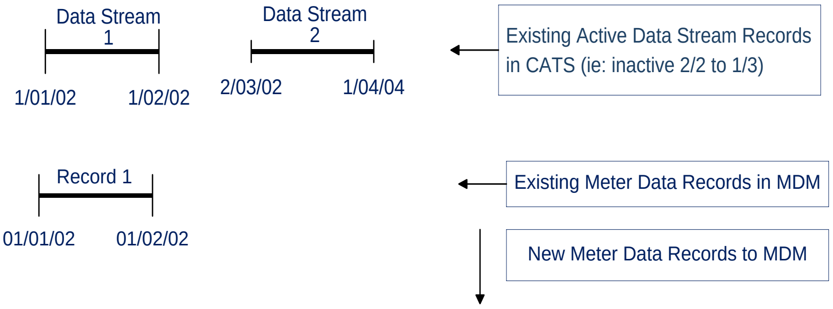
16. This record **will not load**, - as Datastream is not active at end date.



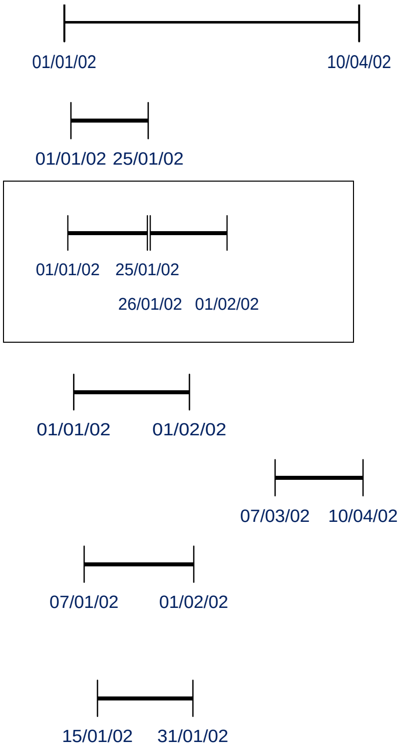
17. This record **will not load**, - as it spans period where Datastream is not active (Datastream not active at start date of record).

E. One existing *metering data* record loaded into MDM (example 2)

The following examples assume that the Datastream is not active for the entire period.


Data Stream 1
1/01/02
1/02/02
Data Stream 2
2/03/02
1/04/04
Existing Active Data Stream Records in CATS (ie: inactive 2/2 to 1/3)
Record 1
01/01/02
01/02/02
Existing Meter Data Records in MDM
New Meter Data Records to MDM

Scenarios in relation to Records Above



18. This record **will not load**, - as it spans period where Datastream is not active

19. This record **will load** if the existing record is an Estimate (read_type_flag = 'E').

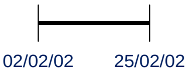
20. These 2 records **will load**. New records will replace the existing record. If the 'meta-read validation' fails, the individual records of the 'meta-read' will be validated separately – records **will not load** as they do not align with existing record.

21. This record **will load**, - replaces existing record 1 if the version date is greater than existing record.

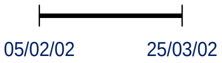
22. This record **will load** – as new record does not overlap any existing record. There will be a gap in the *metering data* for period where Datastream is inactive and also up to 6/3/02.

23. This record will not load unless record 1 is an Estimate as new record does not align with the start date of the existing record.

24. This record will not load unless record 1 is an estimate as new record does not align with the start date of the existing record.



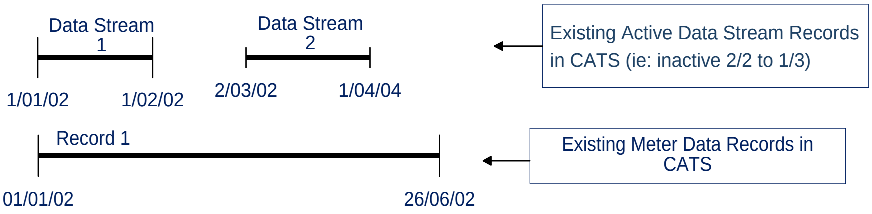
25. This record **will not load** – spans period where Datastream is inactive



26. This record **will not load** – spans period where Datastream is inactive.

F. One existing *metering data* record loaded into MDM that now spans period of inactive Datastream

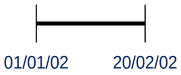
The following examples assume that the Datastream is not active for the entire period.


Data Stream 1
1/01/02
1/02/02
Data Stream 2
2/03/02
1/04/04
Existing Active Data Stream Records in CATS (ie: inactive 2/2 to 1/3)
Record 1
01/01/02
26/06/02
Existing Meter Data Records in CATS

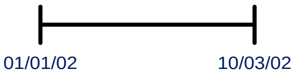
Note: As record 1 spans a period where the Datastream is now inactive, it is viewed as an invalid Meter Reading, and therefore is not included in any validation and will have the same impact on any new record loading result as if the record did not exist.

New Meter Data Records to MDM

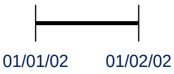
Scenarios in relation to Records Above



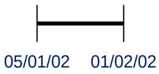
27. This record **will not load**, - as Datastream is not active at end date.



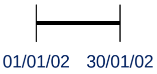
28. This record **will not load**, - as it spans period where Datastream is not active



29. This record **will load,** existing record spanning 1/1 to 26/6 archived to history table.

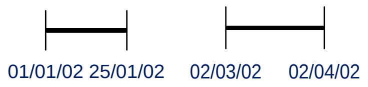


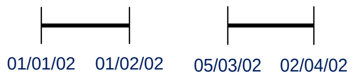
30. This record **will load** - existing record is not validated against as it is an invalid record (spans period where Datastream is inactive).



31. This record **will load,** existing record spanning 1/1 to 26/6 archived to history table. Existing record not validated against since existing read is invalid.

32. Both Records **will load** even though gap exists between 25/1 and 1/2; existing record spanning 1/1 to 26/06 archived to history table. In the future may be different MDPs for these two active Datastream periods. Existing read is invalid so it is NOT validated against

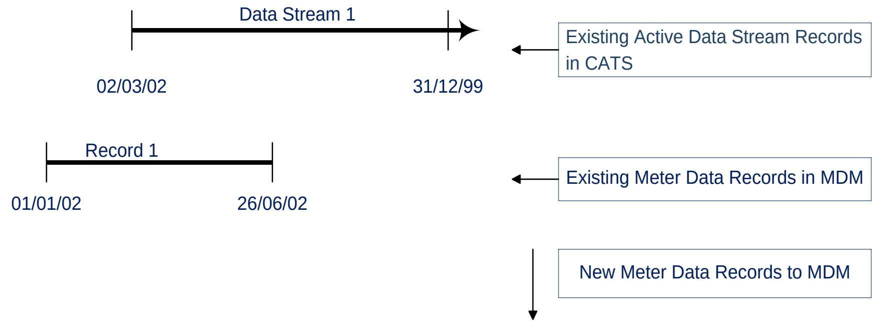




33. Both these records **will load**. Existing record spanning 1/1 to 26/06 archived to history table.

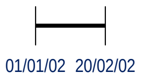
G. One existing metering data record loaded into MDM which now spans period of inactive Datastream

The following examples assume that the Datastream is now not active for the entire period, but has no gaps.

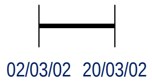


Data Stream 1
02/03/02
31/12/99
Existing Active Data Stream Records in CATS
Record 1
01/01/02
26/06/02
Existing Meter Data Records in MDM
New Meter Data Records to MDM

Scenarios in relation to Records Above



34. This record **will not load**, - as Datastream is not active at start or end date.

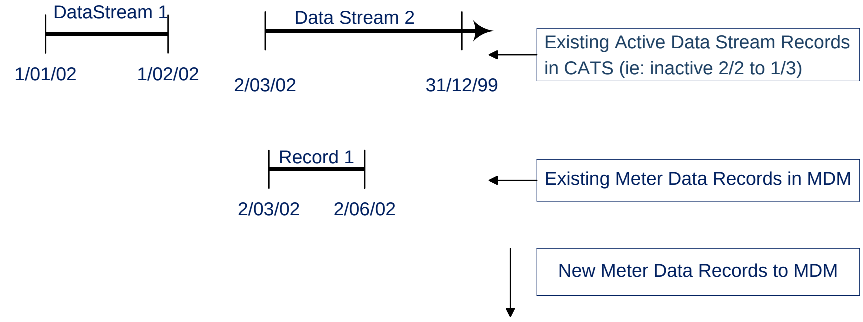


35. This record **will load**, - existing record spanning 1/1 to 26/6 archived to history table. Existing read is invalid so it is NOT validated against.

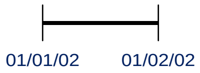
H. Data load in first period of active Datastream where existing *metering data* record loaded into MDM for second period of active Datastream

The following examples assume that the Datastream is not active for the entire period.

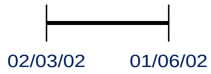
Example 1:


DataStream 1
1/01/02
1/02/02
Data Stream 2
2/03/02
31/12/99
Existing Active Data Stream Records in CATS (ie: inactive 2/2 to 1/3)
Record 1
2/03/02
2/06/02
Existing Meter Data Records in MDM
New Meter Data Records to MDM

Scenarios in relation to Records Above

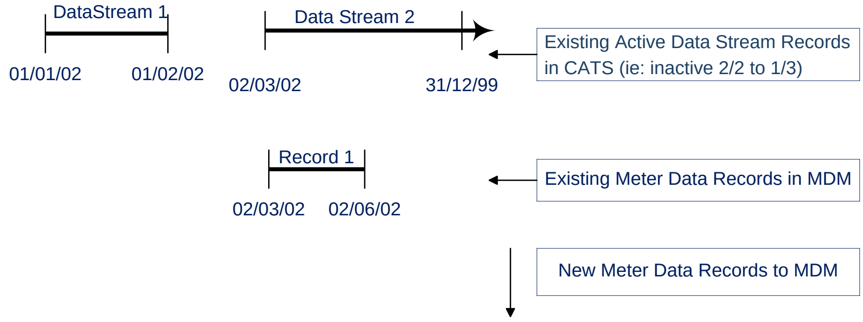


36. This record **will load**, data date range within Datastream active range.

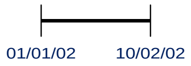


37. This record **will only load** if the existing record is a forward estimate (read_type_flag = 'E').

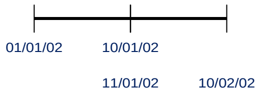
Example 2:


DataStream 1
01/01/02
01/02/02
Data Stream 2
02/03/02
31/12/99
Existing Active Data Stream Records in CATS (ie: inactive 2/2 to 1/3)
Record 1
02/03/02
02/06/02
Existing Meter Data Records in MDM
New Meter Data Records to MDM

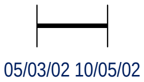
Scenarios in relation to Record Above



38. This record **will not load**, as Datastream is inactive at the end date.



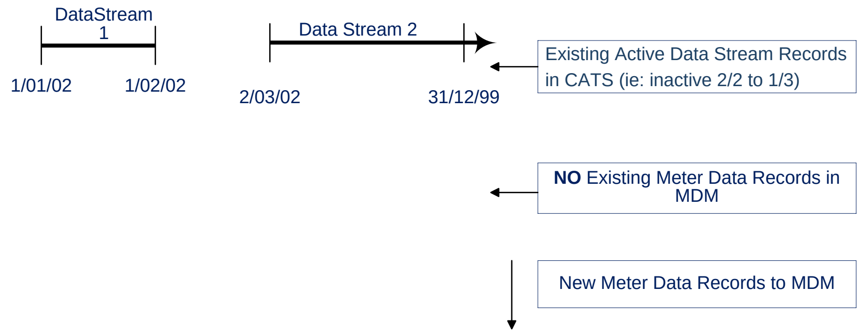
39. First record **will load**. Second record **will NOT** load as Datastream is inactive at the reading end date.



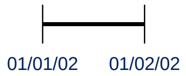
40. This record **will not load** unless the existing Meter Reading is an estimate. New record start date does not align with overlapping existing record start date.

I. Data load in first period of active Datastream where no existing *metering data* record loaded into MDM

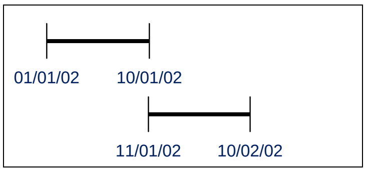
The following examples assume that the Datastream is not active for the entire period.


DataStream 1
1/01/02
1/02/02
Data Stream 2
2/03/02
31/12/99
Existing Active Data Stream Records in CATS (ie: inactive 2/2 to 1/3)
NO Existing Meter Data Records in MDM
New Meter Data Records to MDM

Scenarios in relation to Records Above



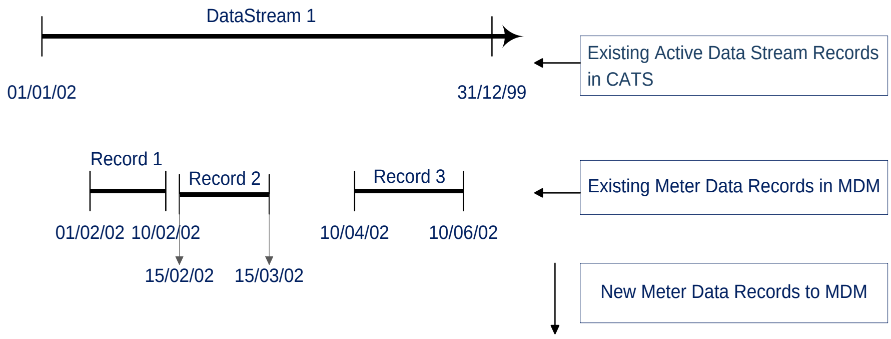
41. This record **will load**.



42. The first record **will load.** The second record **will not load** as the Datastream is not active at the end date of the record.

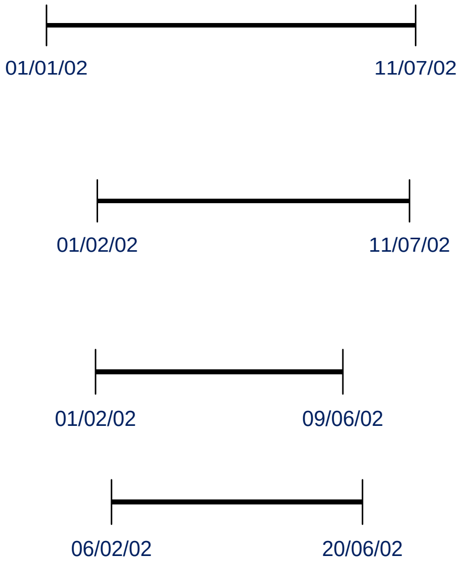
J. Three existing metering data records loaded into MDM

The following examples assume that the Datastream is active for the entire period.


DataStream 1
01/01/02
31/12/99
Existing Active Data Stream Records in CATS
Record 1
Record 2
Record 3
01/02/02
10/02/02
15/02/02
15/03/02
10/04/02
10/06/02
Existing Meter Data Records in MDM
New Meter Data Records to MDM

Scenarios in relation to Records Above

43. This record **will load**. The three existing records will be archived to history. As start date and end date of new record does not overlap any existing records, there is no requirement for alignment of dates.

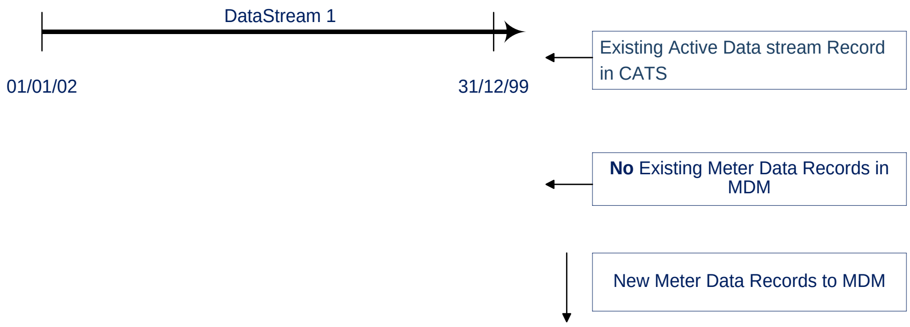


44. This record **will load**. The three existing records will be archived to history. As new record start date overlaps first existing record, there is a requirement for alignment of start dates.

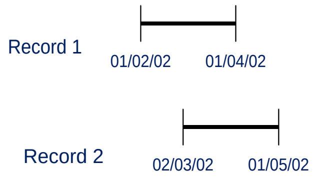
45. This record **will only load** if record 3 is an Estimate as the end date of the new record overlaps existing record 3 (forward estimate means read_type_flag = ‘E’). If record 3 is a forward estimate, then new record replaces previous records 1, 2 and 3.

K. No existing *metering data* records loaded into MDM

The following examples assume that the Datastream is active for the entire period.

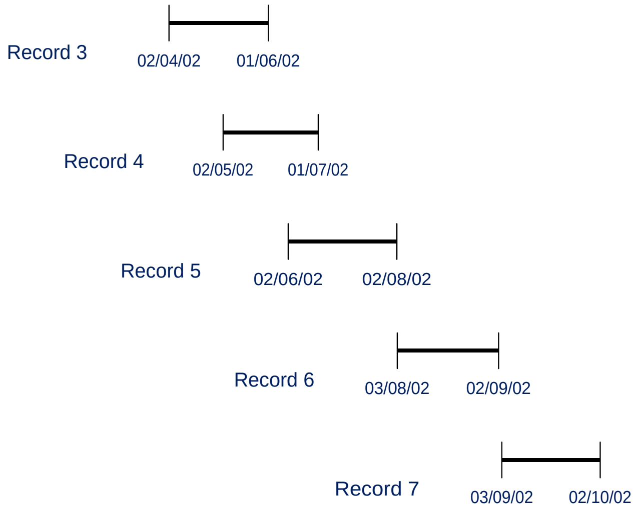

DataStream 1
01/01/02
31/12/99
Existing Active Data stream Record in CATS
No Existing Meter Data Records in MDM
New Meter Data Records to MDM

Note: all of the following records are in the one metering data file.



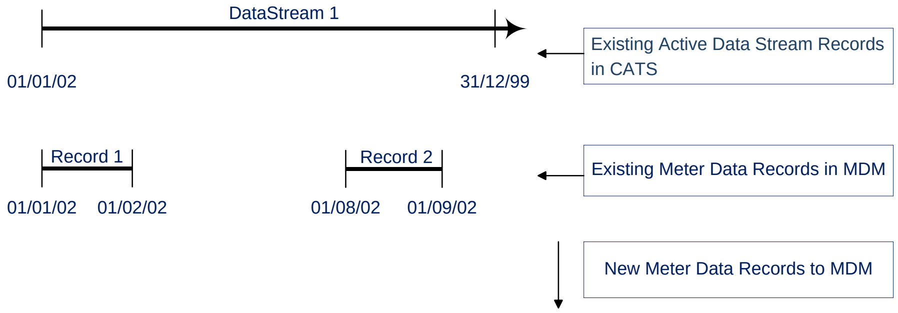
47. Records 1, 3, and 5, 6 and 7 **will load.** Records 2 and 4 will fail as they do not align with the previous loaded record.

One Meta read is created (records 5, 6 & 7) – other read records will be processed on a record by record basis.

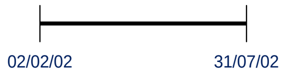


L. Existing *metering data* records loaded into MDM

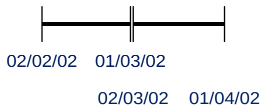
The following examples assume that the Datastream is active for the entire period.


DataStream 1
01/01/02
31/12/99
Existing Active Data Stream Records in CATS
Record 1
01/01/02
01/02/02
Record 2
01/08/02
01/09/02
Existing Meter Data Records in MDM
New Meter Data Records to MDM

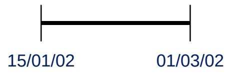
Note: Existing records are <u>not</u> Estimates



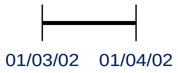
48. Record **will load.**



49. Both records **will load**. Data gap still exists between 2/4/02 and 31/7/02.



50. Record **will not load**. New record does not align with existing record, and record 1 is not an Estimate.



51. Record **will load**. Data gap still exists between 2/2/02 and 28/2/02, and 2/4/02 and 31/7/02.

6. MSATS ERROR CODES (MDM)

All submitted Meter Data Notification data files that do not pass MDMT or MTRD~~MSATS~~ validations prior to data load will create an error. See the Hints and Tips – CATS & NMI Discovery for more detail on the common error codes.

6.1 Validation Failure Error Codes (MDM)

All error codes relating to validation failures of MDM metering data files are tabulated in MSATS. This listing is located under Administration > Codes Maintenance > Error Codes.