MDM FILE FORMAT AND LOAD PROCESS

PREPARED BY:	AEMO MARKETS
VERSION:	1.1
EFFECTIVE DATE:	01 DECEMBER 2017
STATUS:	DRAFT

Approved for distribution and use by:

APPROVED BY:	PETER GEERS
TITLE:	EXECUTIVE GENERAL MANAGER, MARKETS

DATE: 02 /06 /2017

FOR INTERNAL USE ONLY

© 2014 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 resposes, issued as Final.
1.0	1 December 2017	Updated to incorporate changes due to Power of Choice reforms and corrections.

CONTENTS

1. INTRODUCTION 4

1.1 Purpose and Scope 4
1.2 Definitions and Interpretation 4
1.3 Related Documents 4

2. OVERVIEW 4

2.1 Outline of Metering Data Management (MDM) 4
2.2 Inputs Accepted by MDM 4

3. PROCESS OF LOADING METERING DATA 5

3.1 Outline 5
3.2 Security 5
3.3 aseXML Message Format 5
3.4 Header Information 6
3.5 Transaction Information 7
3.6 CSV Consumption Data 8
3.7 CSVIntervalData 10
3.8 CSV Profile Data 12
3.9 Data Collection Type Codes 12
3.10 File Format 13
3.11 Data Load of Metering Data Files via the Browser (Interactive Upload) 14
3.12 Data Load of Metering Data Files via the Batch Process 20

4. FILE VALIDATION 21

4.1 Principles 21
4.2 Validation of MDPVersionDT 21
4.3 Validation of Start and End dates of reads (Including meta-reads) 22
4.4 MSATS Data File Validations 22

5. MSATS ERROR CODES (MDM) 33

5.1 Validation Failure Error Codes (MDM) 34

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 MDM 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/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

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 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 *metering data* notification transaction.

The *metering data* notification transaction is submitted to MSATS by MDPs in the form of a comma separated values-wrapped aseXML file. Details of the creation and submission of this file are in section

3. There are a number of validation requirements undertaken on the *metering data* notification transaction before the data file is accepted by 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 MSATS for *settlements* processing, which can be delivered to MSATS via the browser or the batch interface.

(a) **Browser** – MDPs can load *metering data* manually using the 'Participant Inbox' screen and the 'Upload' action. 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.

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) **Batch** – MDPs can place *metering data* files 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.

3.2 Security

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

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

(b) 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*).

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

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

3.3 aseXML Message Format

To import *metering data* into 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 file must then be zipped and uploaded into MSATS using the browser interface, or by directly placing the file into the appropriate 'Participant Inbox' on the AEMO fileshare.

The 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.

(c) **Transaction Information:** This section contains all of the transaction-specific information, i.e. the actual *metering data* to be loaded.

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 containing the CSV payload data. The file shows the structure of a transaction containing interval data. Note the examples in this document refer to schema version r25, please refer to the latest version available from the AEMO website.

Schema Information

Header Information

Transaction Information

```
<?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,Period04,Period05,Period06,Period07,Period08,Period09,Period10,Period11,Period12,Period13,Period14,Period15,Period16,Period17,Period18,Period19,Period20,Period21,Period22,Period23,Period24,Period25,Period26,Period27,Period28,Period29,Period30,Period31,Period32,Period33,Period34,Period35,Period36,Period37,Period38,Period39,Period40,Period41,Period42,Period43,Period44,Period45,Period46,Period47,Period48,DCTC
1234567890,A1,20091010143542,20090415,EEEEEEEEEEEEEEEEEEEEEEEEAAAAAAAAAAAAAAAAAAAAAAA,3.422,3.825,4.163,3.456,3.979,3.401,3.567,3.556,3.948,3.732,3.743,3.898,3.859,3.899,3.749,3.396,3.685,3.827,3.448,3.562,3.949,3.465,3.462,3.618,3.699,3.838,3.68,4.158,3.705,4.149,3.633,3.514,4.022,4.077,3.916,3.501,3.429,3.796,3.645,3.695,4.079,3.36,3.962,3.432,3.852,3.965, 3.412, 4.002,COMMS</CSVIntervalData>
                        </MeterDataNotification>
                </Transaction>
        </Transactions>
</ase:aseXML>
```

Figure 1 XML 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 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
<MessageID>	A unique, Participant-generated identifier for the file being sent, which can contain up to 50 characters.	'MDA1-MSG-2466453' 'TCAUSTM017746632'

Field Name	Field Format	Example
<MessageDate>	Date and time identifier to be attached to the file, in the following format: yyyy-mm-ddThh:mm:ss.sss+10:00	2001-12-31T08:59:11+10:00
<TransactionGroup>	Identifies the type of transaction being processed. When uploading *metering data* the transaction group will always be 'MDMT'. It is a 4- character string.	MDMT
<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 2 is an example of the 'Header' section of an XML file:

```
<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 XML Header Example

3.5 Transaction Information

Table 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 Transaction information

Field Name	Field Format	Example
<Transactionid>	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.	'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	2001-12-31T08:59:11+10:00
<MeterDataNotificationversion>	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) CSVDataTypes 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 CSVDataTypes are detailed in the next section.	<CSVConsumptionData> <CSVIntervalData> <CSVProfileData> Optional Examples: <CSVIntervalData Name='Interval'> <CSVConsumptionData Name='Consumption'> <CSVProfileData Name='Profile'>

3.6 CSV Consumption Data

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

Each component of the CSVConsumptionData listed in Table 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 MDM 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 1: 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

Field Name	Field Format	Example
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
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* mulitpliers, therefore all consumption values and Meter Readings issued are multiplier adjusted.	1398.667

Figure 3 is an example of the Transaction Information of an aseXML file used in the loading of consumption data. Note the information components included next to the <CSVConsumptionData> 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">

        <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 XML Transaction Information

3.7 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 3 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 4 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
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:

[1] Data Collection Type Code.

```
<?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,EEEEEEEEEEEEEEEEEEEEEEEEAAAAAAAAAAAAAA
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 4 XML Transaction information

3.8 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.

3.9 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 5 DCTC Code – Metering Installation Type Code Mapping

DCTC Code	Metering Installation Type Code
COMMS	COMMS1, COMMS2, COMMS3, COMMS4
COMMS4D	COMMS4D

DCTC Code	Metering Installation Type Code
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*.

Table 6 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,Period04,Period05,Period06,Period07,Period08,Period09,Period10,Period11,Period12,Period13,Period14,Period15,Period16,Period17,Period18,Period19,Period20,Period21,Period22,Period23,Period24,Period25,Period26,Period27,Period28,Period29,Period30,Period31,Period32,Period33,Period34,Period35,Period36,Period37,Period38,Period39,Period40,Period41,Period42,Period43,Period44,Period45,Period46,Period47,Period48,DCTC
1234567890,A1,20091010143542,20090415,EEEEEEEEEEEEEEEEEEEEEEEEAAAAAAAAAAAAAAAAAAAAAAAA,3.422,3.825,4.163,3.456,3.979,3.401,3.567,3.556,3.948,3.732,3.743,3.898,3.859,3.899,3.749,3.396,3.685,3.827,3.448,3.562,3.949,3.465,3.462,3.618,3.699,3.838,3.68,4.158,3.705,4.149,3.633,3.514,4.022,4.077,3.916,3.501,3.429,3.796,3.645,3.695,4.079,3.36,3.962,3.432,3.852,3.965, 3.412, 4.002,COMMS</CSVIntervalData>

Figure 5 Example (MDM) Metering Data File with DCTC

3.10 File Format

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 Zip File Format

Field Name	Field Format	Example
Transaction Group	In the case of MDM transactions, the transaction group will be 'mdmt'.	4 Alphanumeric
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 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:

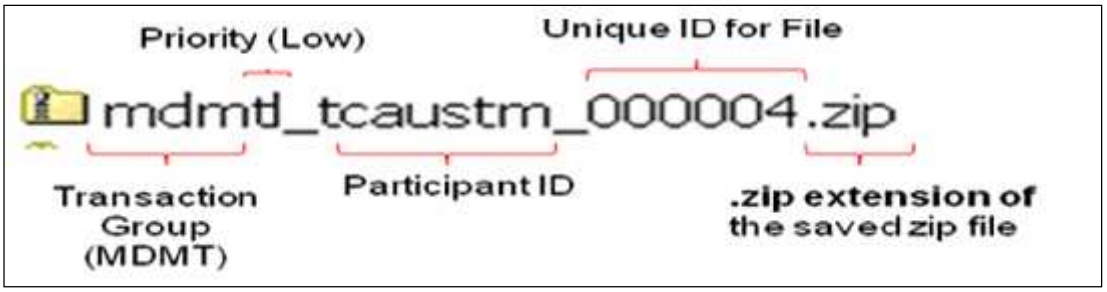


Figure 6 Filename Example

3.11 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

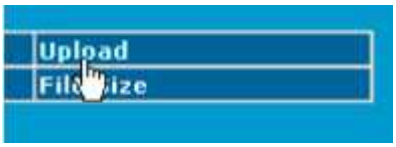
(d) Ensure that the UserID identified in the SecurityContext has been assigned a right that will allow access to the appropriate MSATS batch procedures.

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:

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.

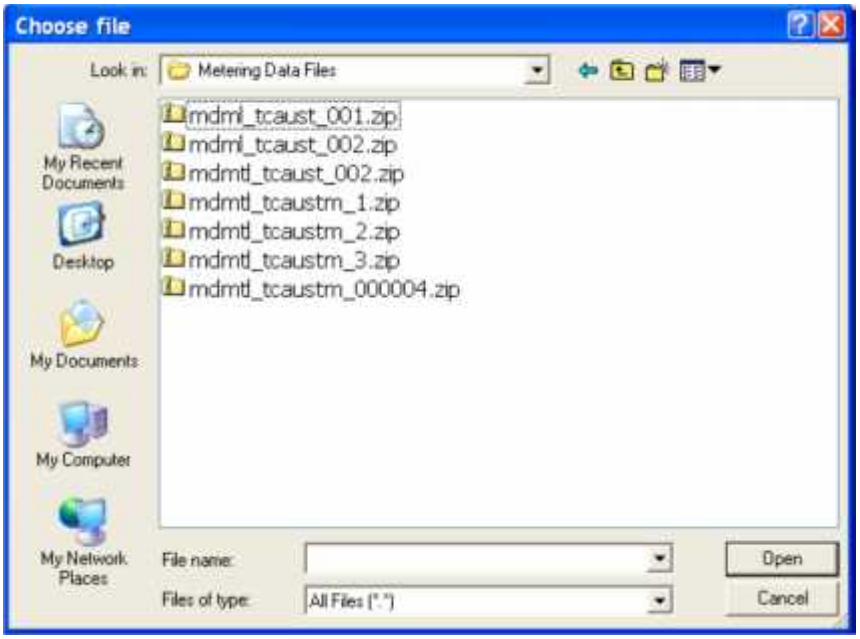
Note: File Upload

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

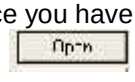
(b) The MDP can either enter the path to the file directly in the text field, or use the 'Browse' button to open the standard navigator type interface.

5. To select a file to import click on the

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.



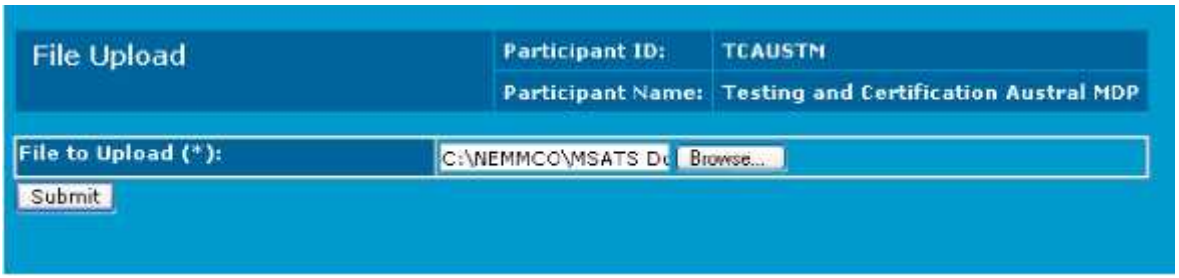
Note:

(a) Once you have found the file you wish to import, highlight it in the window and click on the 

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.



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.

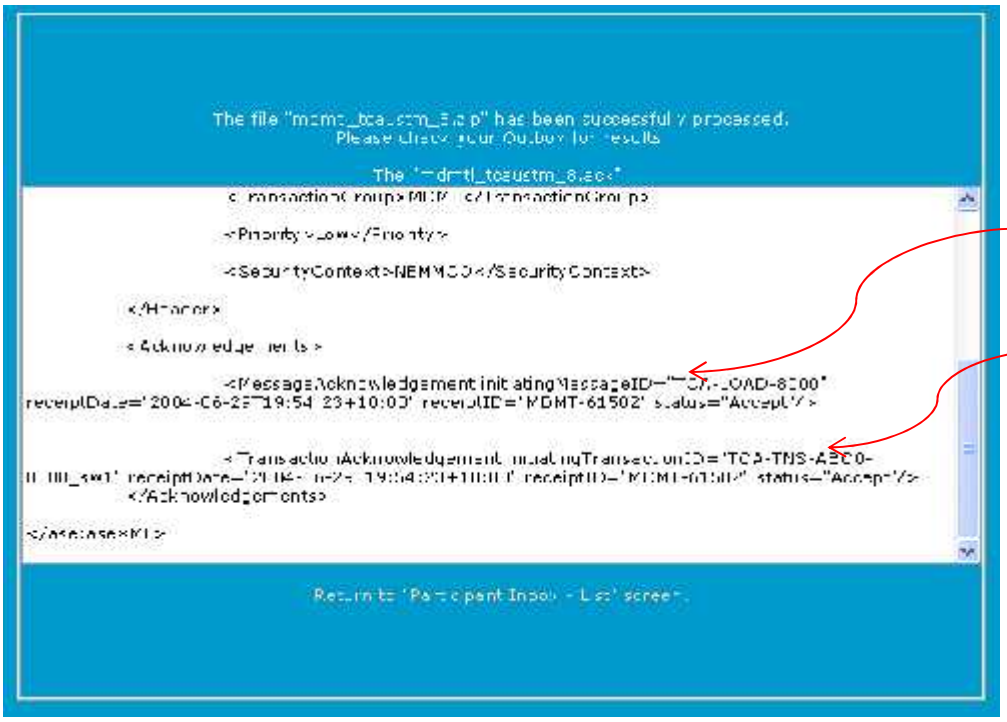
10. After completing the first level validation, MSATS will display the submitted information and the results of the validation in the form of an acknowledgement or .ack file.

    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:



(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.

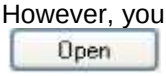
    There is one 'Meter Data Response' in the form of an .xml message in a .zip file for each 'Meter Data Notification' transaction in the original file. Given that MSATS only allows one 'Meter Data Notification' transaction per file, this means that there will be one 'Meter Data Response' file for each 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

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'.

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 

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 7 XML Response Message Format

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

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

The numeric part of the 'MessageID' and the 'TransactionID' matches the numeric part of the .zip file name.

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 '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 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 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>
```

The four digit code in the 'Code' element is an error code. 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
(d) MDP Version Date Time.

3.12 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. FILE VALIDATION

4.1 Principles

Validation of 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 validations undertaken in CATS (before it is accepted by the MDMs):

(a) Datastreams are active. For non-interval Datastreams, this includes for the duration of the 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 are no duplicate 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.

4.2 Validation of MDPVersionDT

For interval and non-interval Meter Readings:

(a) The load of any new *metering data* records into 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 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.

4.3 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 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 Meter Readings to determine whether they are eligible for use in the start and end date tests. 'MDPVersionDT' validation is still undertaken.

4.4 MSATS Data File Validations

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

4.4.1 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

Term	Description
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

4.4.2 Consumption Data

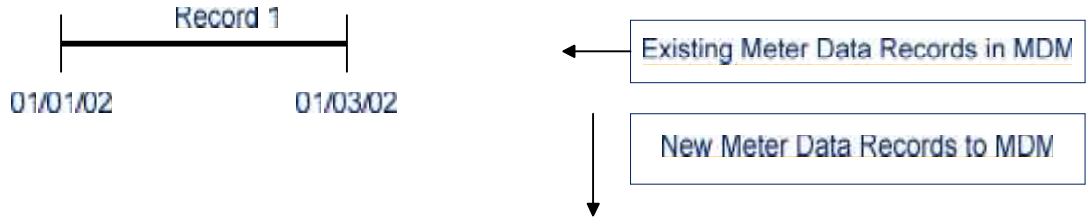
Table 9 Consumption Data Information

Term	Description
Submitting MDP	MDP is the Current MDP for that *NMI* in CATS for the 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

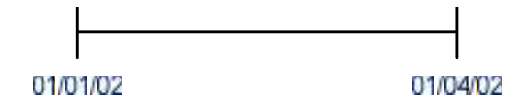
4.4.3 Consumption Meter Data Record date relationship examples

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

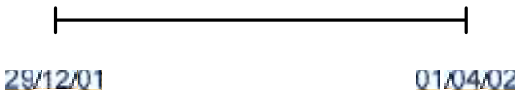
The following examples assume that the Datastream is active for the entire period.



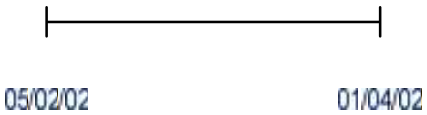
Scenarios in relation to Record Above



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.

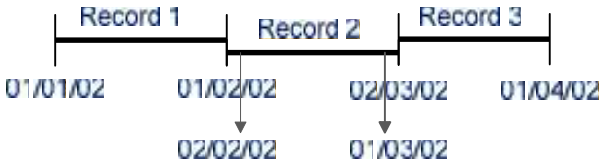


3. This record **will not load** unless record 1 is a forward estimate. In this case the start date of new record is after the start date of the existing record and therefore will fail the validation.

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.

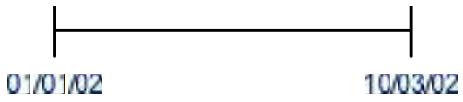
B. Three existing *metering data* records loaded into MDM

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

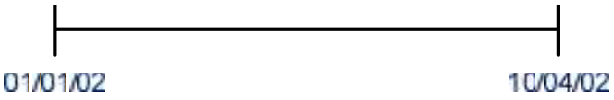


Scenarios in relation to Records Above

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

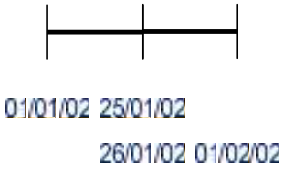


6. This record **will only load** if record 3 is a forward estimate (Record 3's read_type_flag = 'E'). If record 3 is forward estimate, then new record replaces previous records 1,2 and 3.



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

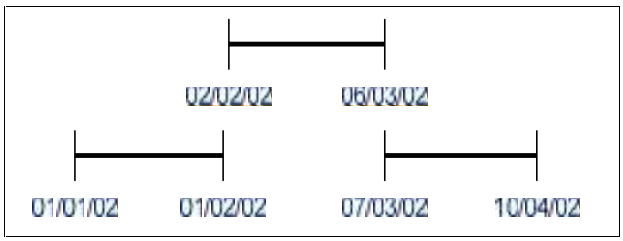
8. This record **will not load** unless record 1 is a forward estimate; - there is no date alignment with Record 1. Data gap would result 26/1/02 – 1/2/02

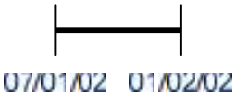


9. These records **will load**. If the 'meta-read' validation fails, each record of the 'meta-read' will be validated separately – these 2 will fail as they do not align with existing records (unless record 1 is an Estimate – then both records will load).

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

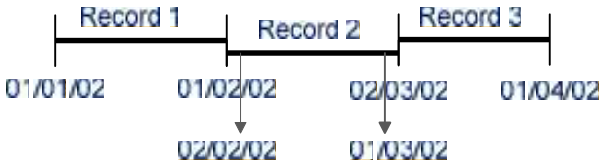
11. All of these records **will load**. New records will replace existing records 1,2 and 3. If the 'meta-read' validation fails, each record of the 'meta-read' will be validated separately – record 1 will load if it passes the 'MDPVersionDT' test, records 2 and 3 will fail as they do not align with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



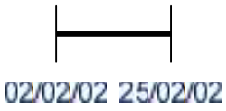


12. This record **will NOT load** unless record 1 is an Estimate – there is no start date alignment with record 1

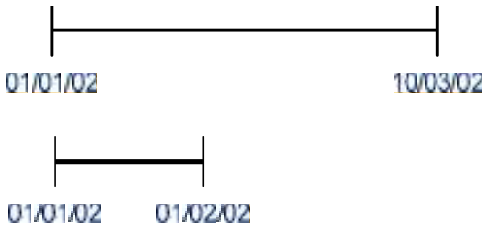
C. Overlapping Meter Readings in a transaction



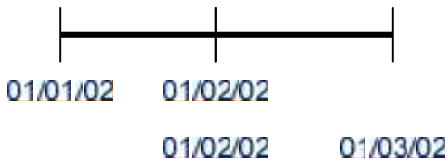
Scenarios in relation to Existing Records Above



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.



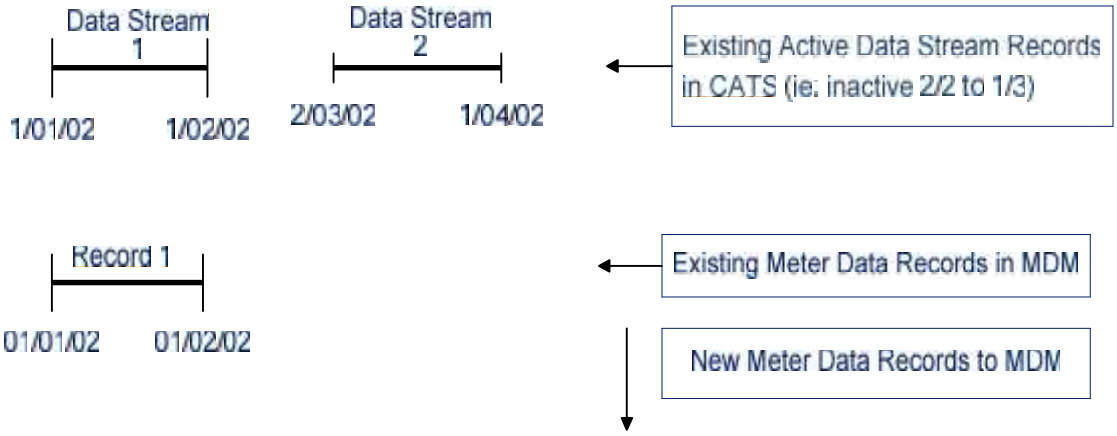
14. This first record **will only load** if record 3 is an Estimate (read_type_flag = 'E') and its 'MDPVersionDT' is greater than the max of 'MDPVersionDT' of the first 2 existing records. Record 2 will **not** load as it is a duplicate record in the file, (see Section 3.1).



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.

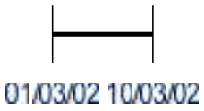
D. One existing *metering data* record loaded into MDM (example 1)

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



Scenarios in relation to Records Above

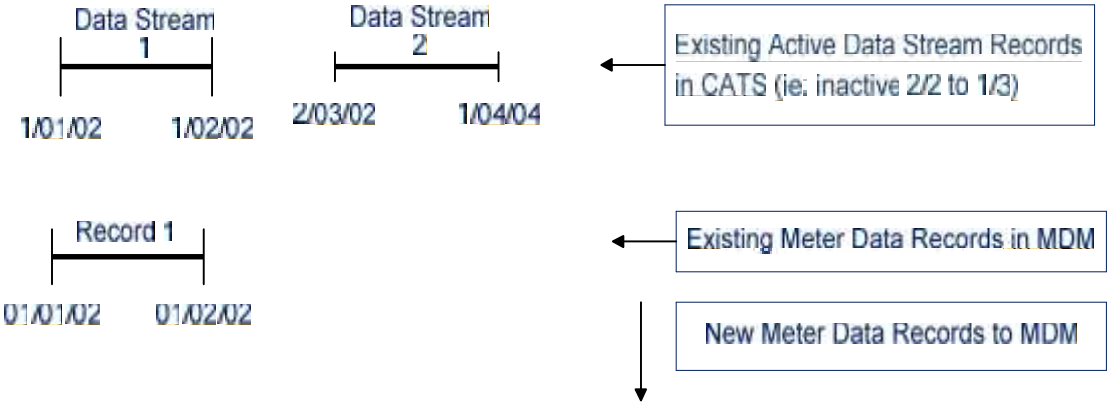
16. This record **will not load**, - as Datastream is not active at end date.



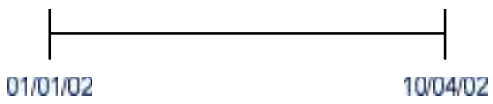
17. This record **will not load**, - as it spans period where Datastream is not active (Datastream not active at start date of record).

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.

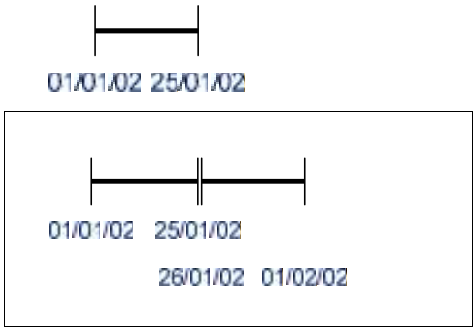


Scenarios in relation to Records Above



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.

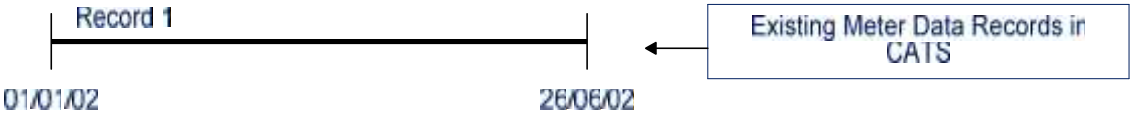
24. This record **will not load** unless record 1 is an estimate as new record does not align with the start date of the existing record.

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

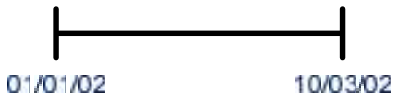
The following examples assume that the Datastream is not active for the entire period.



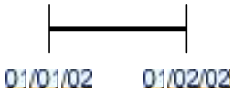
Note: As record 1 spans a period where the Datastream is now inactive, it is viewed as an invalid Meter Reading, and therefore is not included in any validation and will have the same impact on any new record loading result as if the record did not exist.

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).

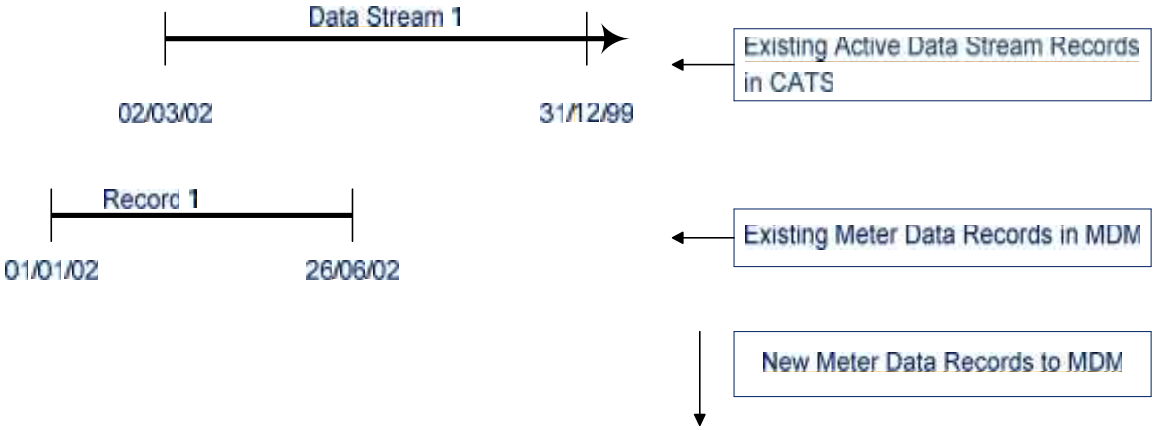
31. This record **will load,** existing record spanning 1/1 to 26/6 archived to history table. Existing record not validated against since existing read is invalid.

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.



Scenarios in relation to Records Above

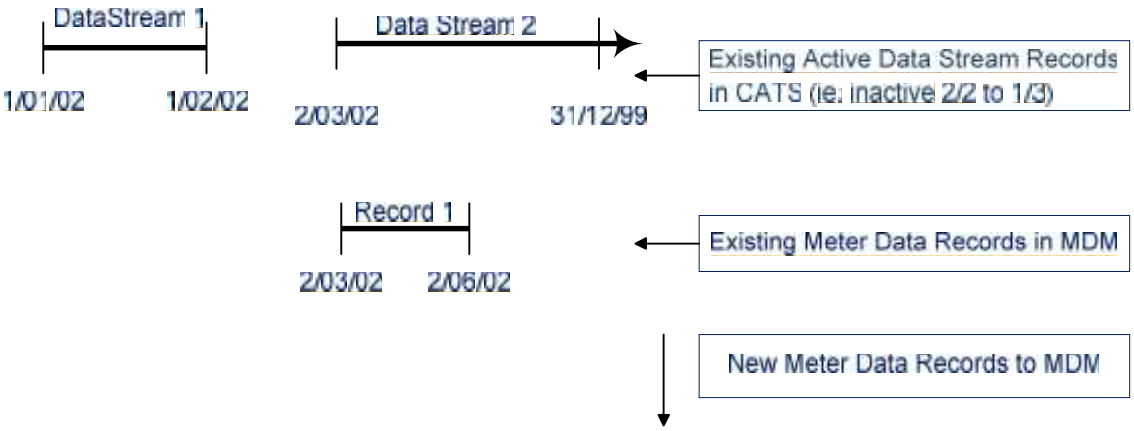
34. This record **will not load**, - as Datastream is not active at start or end date.

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.

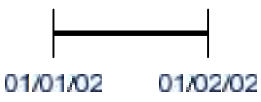
H. Data load in first period of active Datastream where existing *metering data* record loaded into MDM for second period of active Datastream

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

Example 1:



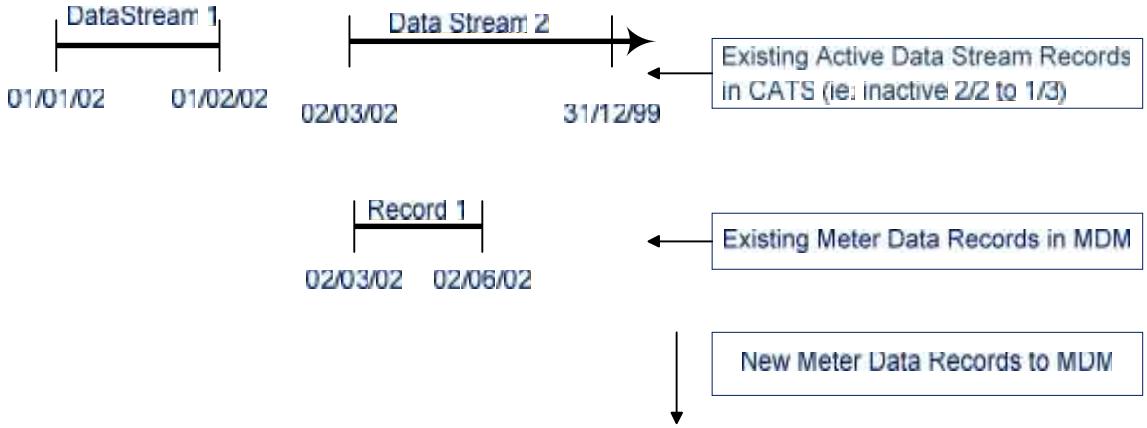
Scenarios in relation to Records Above



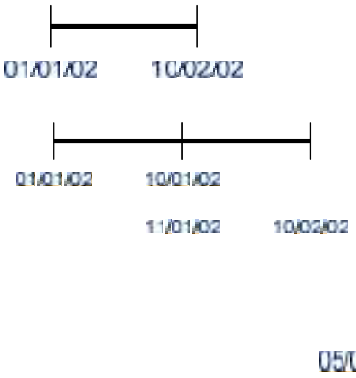
36. This record **will load**, data date range within Datastream active range.

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

Example 2:

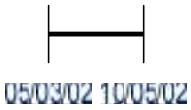


Scenarios in relation to Record Above



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

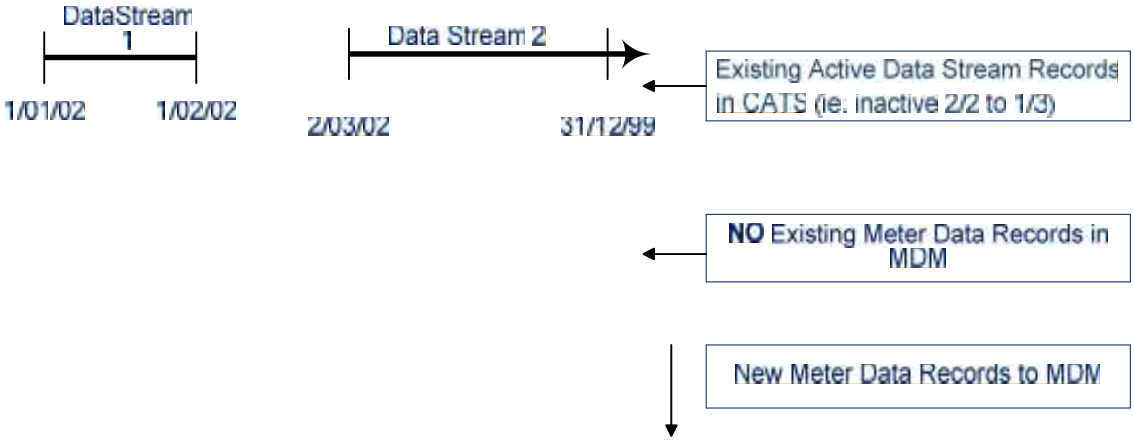
39. First record **will load**. Second record **will NOT** load as Datastream is inactive at the reading end date.



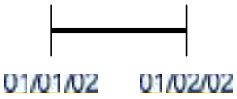
40. This record **will not load** unless the existing Meter Reading is an estimate. New record start date does not align with overlapping existing record start date.

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

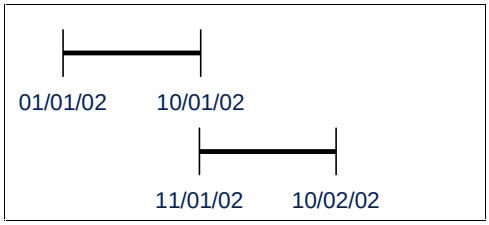
The following examples assume that the Datastream is not active for the entire period.



Scenarios in relation to Records Above



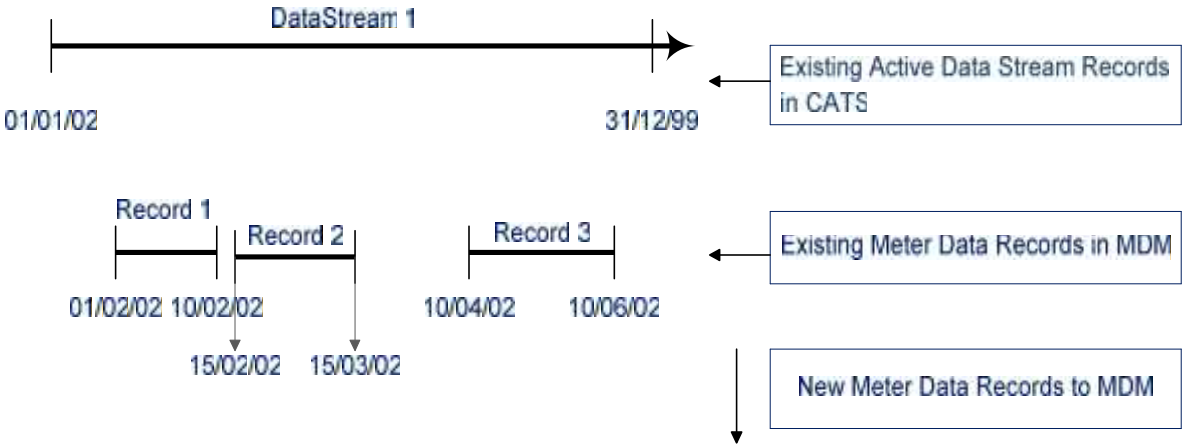
41. This record **will load**.



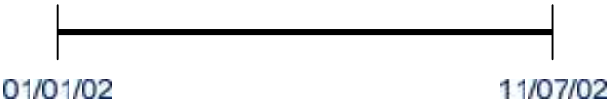
42. The first record **will load.** The second record **will not load** as the Datastream is not active at the end date of the record.

J. Three existing metering data records loaded into MDM

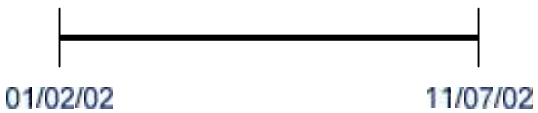
The following examples assume that the Datastream is active for the entire period.



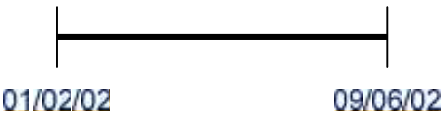
Scenarios in relation to Records Above



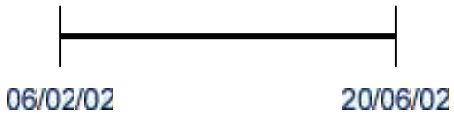
43. This record **will load**. The three existing records will be archived to history. As start date and end date of new record does not overlap any existing records, there is no requirement for alignment of dates.



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.

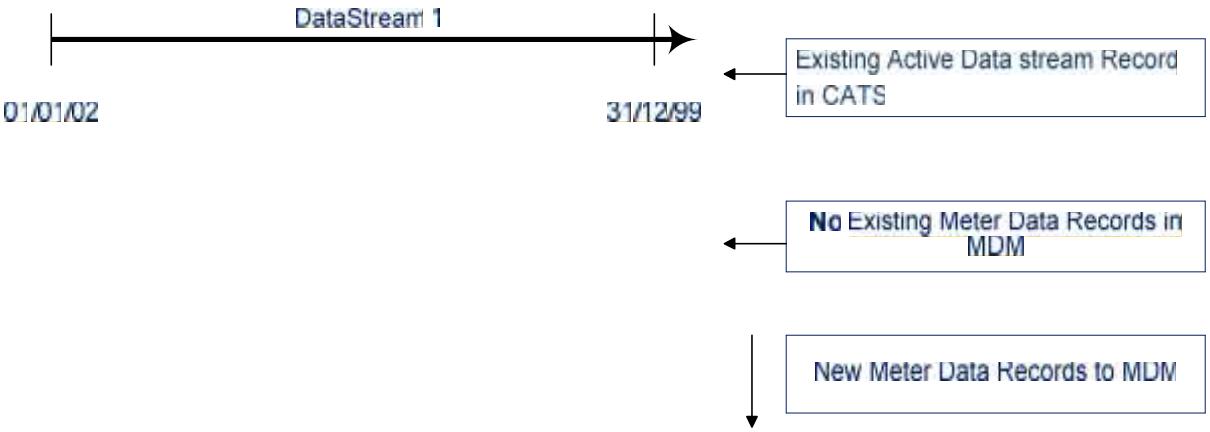


45. This record **will only load** if record 3 is an Estimate as the end date of the new record overlaps existing record 3 (forward estimate means read_type_flag = 'E'). If record 3 is a forward estimate, then new record replaces previous records 1, 2 and 3.

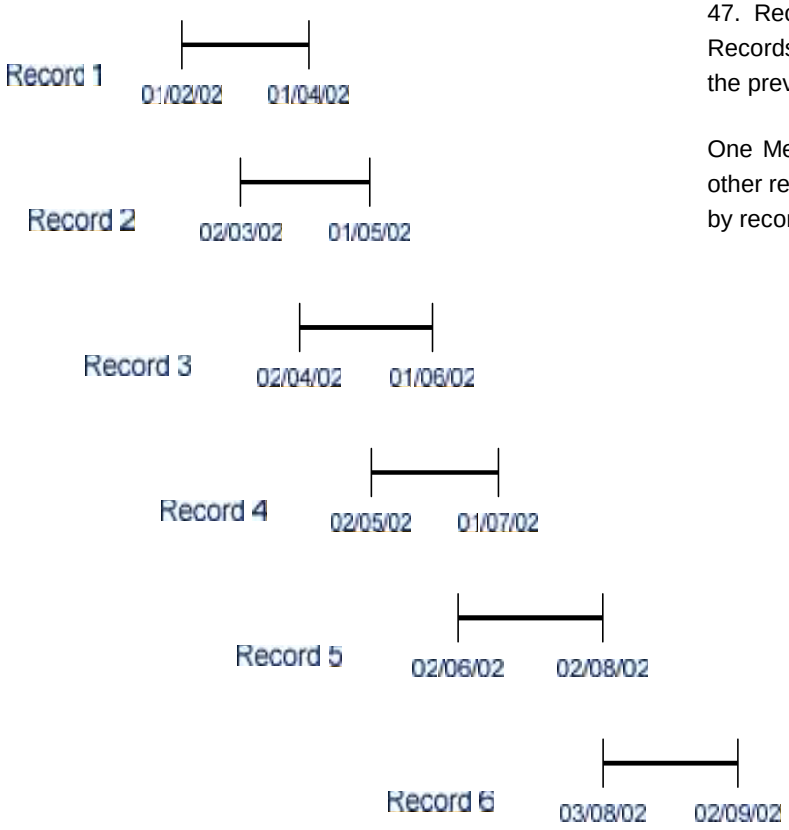


K. No existing *metering data* records loaded into MDM

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



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

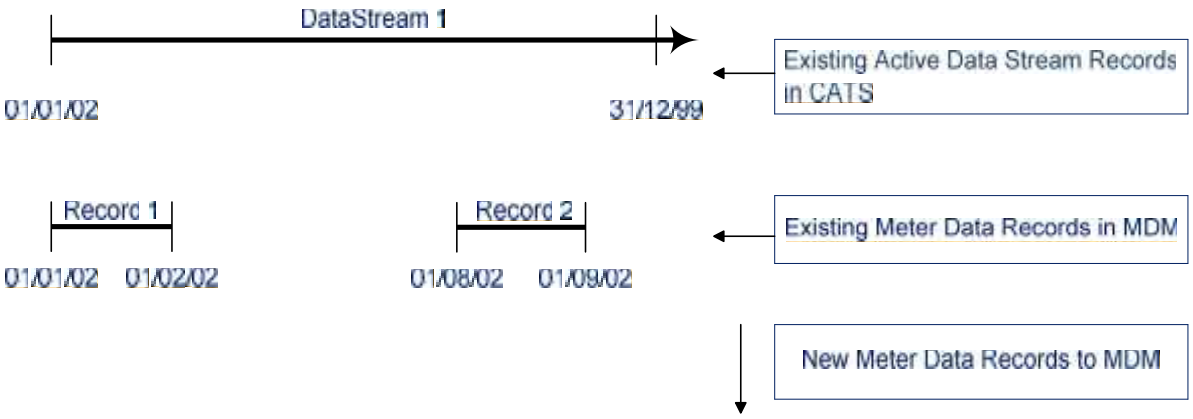


47. Records 1, 3, and 5, 6 and 7 **will load.** Records 2 and 4 will fail as they do not align with the previous loaded record.

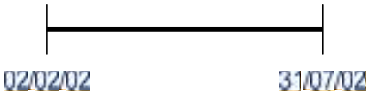
One Meta read is created (records 5, 6 & 7) – other read records will be processed on a record by record basis.

L. Existing *metering data* records loaded into MDM

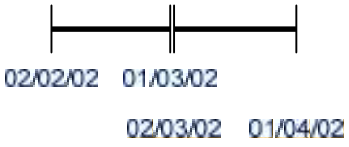
The following examples assume that the Datastream is active for the entire period.



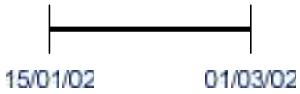
Note: Existing records are <u>not</u> Estimates



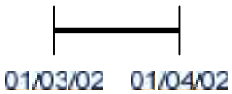
48. Record **will load.**



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.

5. MSATS ERROR CODES (MDM)

All submitted Meter Data Notification data files that do not pass 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.

5.1 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.