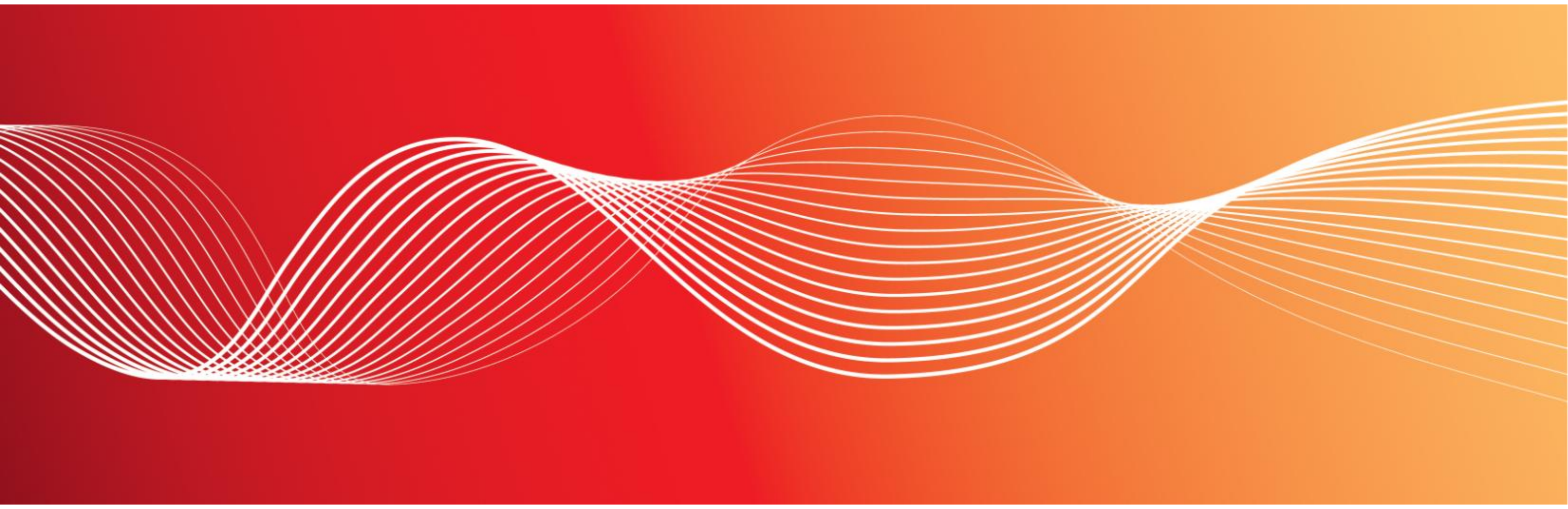
TECHNICAL GUIDE TO BULK DATA TOOL IN MSATS

MARCH 2012



Version: 0.03
Reference: ELECMARKDEV-3-168

Disclaimer

This document is made available to you on the following basis:

Purpose

This Technical Guide to Bulk Data Tool in MSATS ("Guide") has been produced by the Australian Energy Market Operator Limited (AEMO) to provide information about AEMO's IT systems available to participants as at the date of publication.

No substitute

This Guide is not a substitute for, and should not be read in lieu of, the current National Electricity or Gas Rules ("Rules"), National Electricity or Gas Law ("Law"), or any other relevant laws, codes, rules, procedures or policies. Further, the contents of this Guide do not constitute legal or business advice and should not be relied on as a substitute for obtaining detailed advice about the Law, the Rules, or any other relevant laws, codes, rules, procedures or policies, or any aspect of the national electricity market or the electricity industry.

No warranty

While AEMO has used due care and skill in the production of this Guide, neither AEMO, nor any of its employees, agents and consultants make any representation or warranty as to the accuracy, reliability, completeness or suitability for particular purposes of the information in this Guide.

Limitation of liability

To the extent permitted by law, AEMO and its advisers, consultants and other contributors to this Guide (or their respective associated companies, businesses, partners, directors, officers or employees) shall not be liable for any errors, omissions, defects or misrepresentations in the information contained in this Guide, or for any loss or damage suffered by persons who use or rely on such information (including by reason of negligence, negligent misstatement or otherwise). If any law prohibits the exclusion of such liability, AEMO's liability is limited, at AEMO's option, to the re-supply of the information, provided that this limitation is permitted by law and is fair and reasonable.

Copyright	Copyright © 2012 Australian Energy Market Operator Limited. All rights reserved.
Distribution	Available to AEMO participants.
Prepared by	Information Management & Technology - Electricity IT Solutions (EITS)
Last update	Monday, 2 April 2012 12:35
Documents made obsolete	The release of this document is the first version of Technical Guide to Bulk Data Tool in MSATS, and makes obsolete no other documents.

Further Information

For further information, please visit AEMO's website www.aemo.com.au or contact:

AEMO Help Desk	Phone: 1300 236 600 (1300 AEMO 00) and follow the prompts. E-mail: helpdesk@aemo.com.au

Contents

Figures

Tables

Abbreviations and Symbols

These abbreviations, symbols, and special terms assist the reader's understanding of the terms used in this document. For definitions of these terms, the reader should always refer to the applicable market Rules.

Table 1: Abbreviations and explanations

Abbreviation	Abbreviation explanation
AEMO	Australian Energy Market Operator
EST	Eastern Standard Time (in Australia)
BDT	Bulk Data Tool
BCT	Bulk Change Tool
FCAS	frequency control ancillary services
FTP	File transfer protocol
EMMS	[wholesale] Electricity Market Management System; software, hardware, network and related processes to implement the National Electricity Market (NEM)
MSATS	[retail electricity] Market Settlement and Transfer Solution
MW	Megawatt
NEM	National Electricity Market
NER	National Electricity Rules; also often just called the Rules

NMI	[electricity] National Metering Identifier
SRA	Settlements Residue Auction.
CATS	Consumer Administration & Transfer Solution
CR	Change Request

Special terms

Table 2: Terms and definitions

Term	Definition
Market time	Eastern Standard Time (in Australia).
MarketNet	AEMO’s private network available to participantIDs.
ParticipantID	Registered participant identifier
Rules	National Electricity Rules

1 Introduction

1.1 Purpose

The intention of this document is to:

- Describe the Bulk Data Tool (BDT).
- Show the relationship between the aseXML data for BDT and the processing of that data.
- Provide a reliable reference for the intended usage of the aseXML elements relevant to the BDT.

1.2 Audience

This document is relevant to:

- Implementers of interfaces to AEMO's systems.
- Maintainers of interfaces to AEMO's systems.

1.3 What's in this guide

In addition to introductory material and references, this guide includes:

- Description of the general intended usage of the Bulk Data Tool (BDT). See "BDT general usage" on page 3.
- Thorough and complete description of all aseXML relevant to the BDT, both for delivery of data to AEMO and the response files back to the providing participant. The description of the aseXML descriptions is intended to be kept up-to-date, effectively providing a reliable reference for on-going use. See "aseXML for BDT" on page 9 and "AseXML types" on page 17.

Text in this format, indicates the details of a document, help desk, or web page are listed in the section "References" on page 78.

2 Background

2.1 Published materials

The first release of aseXML supporting Bulk Data Tool was r9 version 1.5, dated 20 August 2002. The next, and last, change was r15, dated 22 March 2009, which, for the schema file for BDT, was for documentation rather than structural changes to the aseXML schema.

2.2 Process summary

In essence, the mechanism to deliver an efficient process facilitating the initial one-off population of standing data fields in MSATS is to use a specially-configured version of the Bulk Data Tool (BDT) so that:

- Each relevant participant places a file with the standing data updates or inserts in their MSATS participant inbox. The file uses the BDT request transaction defined in aseXML.
- The BDT file handler generates an acknowledgement file, in the same way as the normal process used by MSATS. If the acknowledgement file is negative, no further processing is possible.
- When processed, the BDT file handler places a response file in the MSATS participant outbox. The response file uses the BDT response transaction defined in aseXML.

3 BDT general usage

The Bulk Data Tool assists participants to load or modify large amounts of standing data into MSATS in an efficient manner.

To minimise impact on MSATS, the BDT directly inserts or updates the standing data into the MSATS database, and avoids as much overhead as possible. So, using the BDT:

- Inserts or updates the current record.
- Obviates use of the change request (CR).
- Avoids issue of notifications.
- Ignores support of the History Model.

The standing data that can be updated by BDT is described in aseXML by the complexType of ElectricityStandingData (for more details, see section 4 “aseXML for BDT” on page 9):

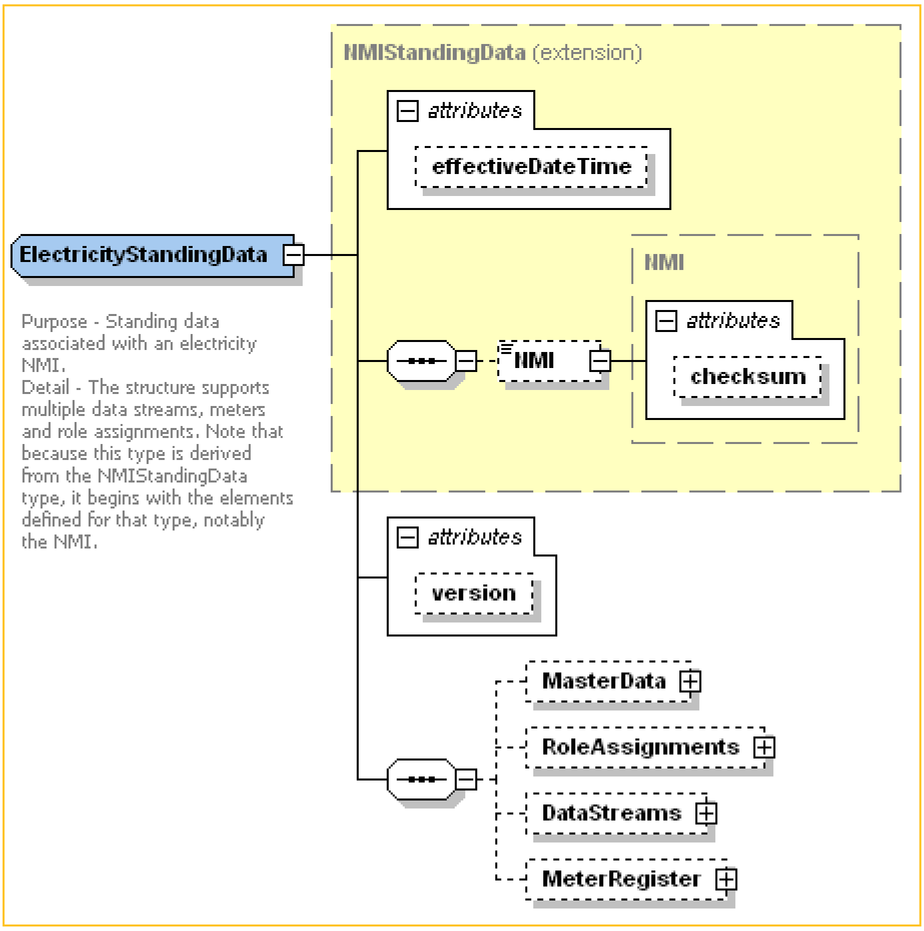

NMIStandingData (extension)
attributes
effectiveDateTime
ElectricityStandingData
NMI
attributes
NMI
checksum
Purpose - Standing data associated with an electricity NMI.
Detail - The structure supports multiple data streams, meters and role assignments. Note that because this type is derived from the NMIStandingData type, it begins with the elements defined for that type, notably the NMI.
attributes
version
MasterData
RoleAssignments
DataStreams
MeterRegister

Figure 1: ElectricityStandingData type

In practice, the use of BDT is to satisfy a particular business scenario and by the circumstances of a given scenario determine the acceptable data for that scenario.

3.1 Interaction with file server

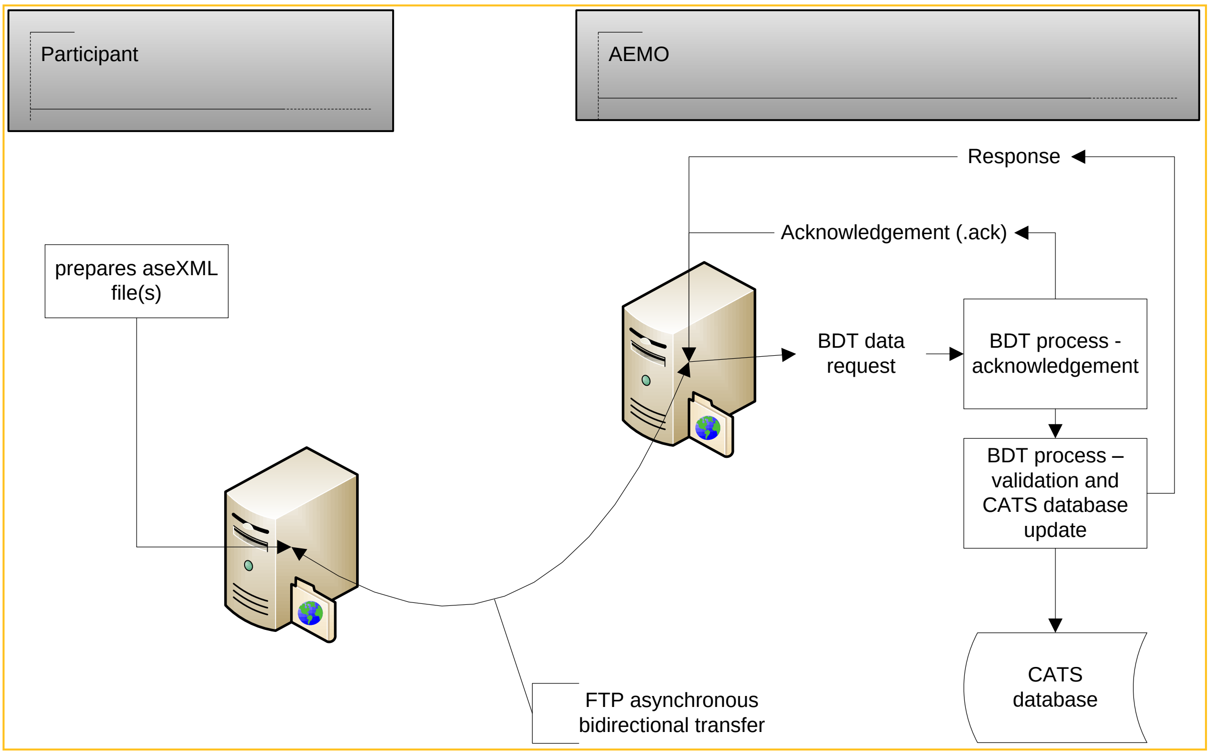

Participant
AEMO
Response
Acknowledgement (.ack)
prepares aseXML file(s)
BDT data request
BDT process - acknowledgement
BDT process – validation and CATS database update
CATS database
FTP asynchronous bidirectional transfer

Figure 2: BDT process overview

Place a .ZIP file in the inbox for MSATS on the relevant file server (pre-production or production), where the .ZIP file contains one file with one ase:CATSBulkDataRequest transaction. Each ase:CATSBulkDataRequest transaction only contains details on one or more NMIs.

The initial processing validates the aseXML file for validity against the schema and checks there are no internal inconsistencies at the file level. The acknowledgement file name is the same for positive and negative acknowledgements; it is the content of the file that determines whether further processing is possible or not.

The BDT process puts the acknowledgement files in the outbox for the submitting participant.

Once processed, the response file appears in the outbox, containing an ase:CATSBulkDataResponse transaction in response to each input transaction. Each NMI is accepted or rejected independently of other NMIs in the transaction.

Files go through the normal MSATS acknowledgement process, but without archiving by AEMO in the visible archive files on the file server.

3.2 File naming convention

The BDT process identifies its input files by a file mask. Typically the file mask is BDT*.zip, but the actual file masks (there may be more than one file mask) are specified for each business usage of BDT.

The acknowledgement file has the same name as the input file, except the suffix is '.ack'.

The response file has the string '_response' added to the same name as the input file.

3.3 Duplicate filenames

When the BDT generates a response file name that already exists, a number is added to the end of the '_response' string to produce a unique file name. The process of creating a unique file name is to:

- Avoid overwriting a response file.
- Accept a BDT file with the same filename as has already been processed.

3.4 aseXML

All BDT files conform to a specific aseXML schema and configuration of non-versioned schema files, depending upon the circumstances. AEMO notifies all affected parties of the aseXML details for a given business usage.

3.5 Business rules

Historical records (non-current) cannot be updated. A current record is where MaintActFlg = 'A' and EndDate = 31-Dec-9999.

AEMO cannot accept any attempt to update data that may alter the results of a previous run of MSATS settlements.

Inserts have a StartDate of the day before the processing date, so the inserted data is effectively current data.

3.6 Validations

Duplication of any key in a transaction is confusing because there is a non-deterministic choice between which one needs to be used and which one is to be rejected. For this reason, duplicates of keys in a transaction cause rejection of the whole transaction. The business keys depend upon the specific intention, and the business keys are the primary keys of the underlying database tables, being:

- NMI.
- NMI and Role.
- NMI and Meter Serial.

- NMI and Suffix.
- NMI and Meter Serial and Register Identifier.

Multiple transactions in a file are rejected for the same reason as duplication of a key in a transaction. Similarly, multiple files in a ZIP are inappropriate; so the BDT process simply processes the first file and ignores any others.

Any validation failure for part of a NMI invalidates the whole NMI, but does not invalidate other NMIs in the transaction.

Core validations are:

- Compliance with aseXML using the release specified for the particular scenario.
- Data is acceptable only for Meter and Register records of status “C” (Current) and for NMIs of all statuses except for “X” (Extinct).

Configurable validations include:

- BDT group. For some purposes, use of BDT is relevant for a related group of participants. When relevant, AEMO agrees details with all affected parties.

Size of file

Limit the size of request files to 10 Mb zipped (approx. 100 Mb unzipped). Although the size limit is not rigorously enforced, the considerations for the limit include the processing resources at AEMO’s end, plus ensuring all participants get access to processing time.

Event codes for BDT

Every NMI has at least one ase:Event returned.

An ase:Event with a code of 0 indicates the NMI was accepted by the BDT.

An ase:Event with a code other than 0 indicates the NMI was rejected by the BDT.

Multiple error events can be reported. Where BDT rejects a NMI, the specific errors are also reported as additional events and these can occur at both the NMI and row level.

Table 3: Codes, descriptions and actions

Code	Description	Recommended action or comments
0	OK	
5000	NMI rejected by BDT	
5001	Invalid Value	
5002	Forbidden Field	
5003	Value Outside Range Check Boundary	
5004	Invalid Number	

Code	Description	Recommended action or comments
5005	Code Value Invalid	
5006	Active Data Stream Present	
5007	EMBNet Value Not Null	
5008	Aggregate Flag Not Yes	
5009	No MeterRegister Records	
5010	Invalid Profile for Jurisdiction	
5011	Invalid Jurisdiction Code	
5012	Internal Error. Invalid property name	
5013	Grouping Error	
5014	NMI Checksum Invalid	
5015	Pending Change Request	
5016	Internal Error	
5017	Role can not be Submitted	
5018	Role not Present	
5019	Grouping/Role Assignment Mismatch	
5020	Not a Tier 1 NMI	
5021	Maximum Field Length Exceeded	
5022	Required Field not Present	
5023	Forbidden Field Present	
5024	Required Role not Assigned	
5025	Updating Inactive Record	
5085	Field may not be submitted as a permitted role is not active over the date range of the record	
5086	Field value must be comprised only of alphabetic characters	
5087	Field value must be comprised only of alphanumeric characters	
5088	Field value must be comprised only of uppercase alphabetic or numeric characters	

Code	Description	Recommended action or comments
5089	Length of the field value must be exactly equal to the defined field length	
5090	Field value must be comprised only of characters in Character Set 1	Character Set 1 matches the regular expression "^[\x20-\x7E-[&<>'\"]]*$"
5091	Field value must be comprised only of characters in Character Set 2	Character Set 2 matches the regular expression "^[\x09\x0A\x0D\x20-\x7E-[&<>'\"]]*$"
5092	Code not active over the date range of this record	
5093	Parent record is not active over the date range of this record	
5094	Null values not allowed for this field	
5095	Parent record does not exist	
5096	This field already has a value and may not be updated	
5097	This field has an existing value that marks the record as not updateable	

Based on the returned information and the original submitted input file, a participant is able to update their local database and distribute these changes to their related participants, as appropriate.

3.7 Operation

AEMO manually runs the BDT process, business days and business hours only (with specifics determined by circumstances and announced through normal channels).

The BDT process skips files already acknowledged, to avoid reprocessing.

The BDT process expects participants to follow the normal MSATS acknowledgement process (that is, delete on .ACK, etc.). The difference with the BDT process is that the files are not archived in the archive folders on the file server.

4 aseXML for BDT

Two transactions for BDT, being ase:CATSBulkDataRequest and ase:CATSBulkDataResponse, were introduced in r9 of aseXML.

Acknowledgements conform to the format used for all aseXML transactions.

4.1 CATSBulkDataRequest

The element in aseXML implementing the BDT request transaction has the XPath *aseXML/ Transactions/Transaction/CATSBulkDataRequest*, with the short name of ase:CATSBulkDataRequest.

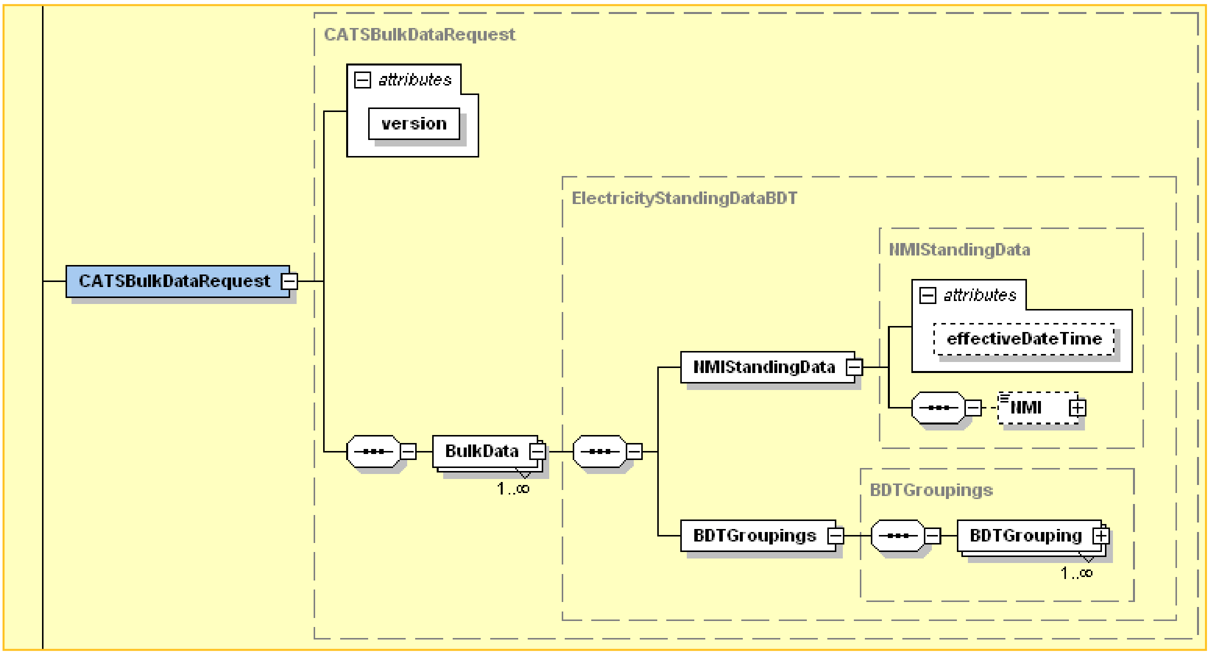
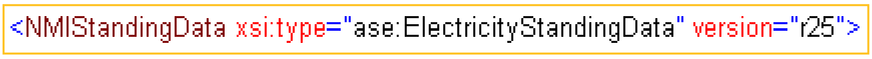


Figure 3: ase:CATSBulkDataRequest element

To implement the BDT request transaction, explicitly use the ase:ElectricityStandingData type for the ase:NMIStandingData element, like:



```
<NMIStandingData xsi:type="ase:ElectricityStandingData" version="r25">
```

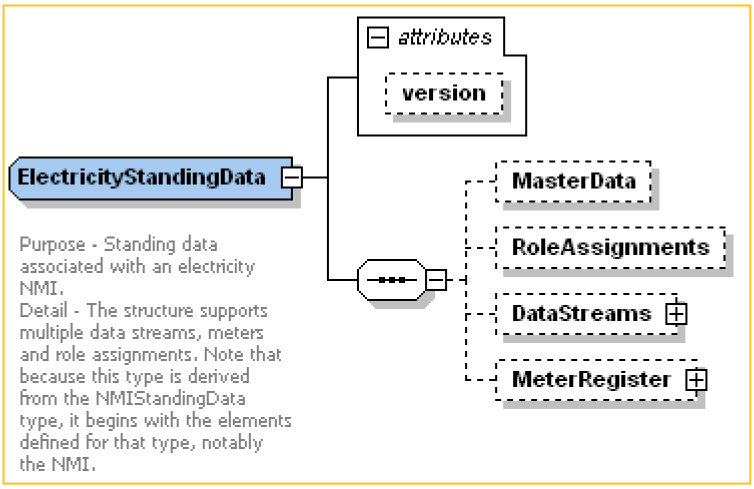

attributes
version
ElectricityStandingData
MasterData
RoleAssignments
DataStreams
MeterRegister
Purpose - Standing data associated with an electricity NMI.
Detail - The structure supports multiple data streams, meters and role assignments. Note that because this type is derived from the NMIStandingData type, it begins with the elements defined for that type, notably the NMI.

Figure 4: ase:ElectricityStandingData type

For the BDT request transaction, the following table lists 3 levels of aseXML nodes in alphabetical sequence, with each XPath relative to ase:CATSBulkDataRequest.

Table 4: aseXML nodes in ase:CATSBulkDataRequest

XPath to aseXML node	aseXML type and restrictions
.	*Type ase:CATSBulkDataRequest (complex)*, (§5.1.31 on page 36)
@version	r9—*Type ase:ReleaseIdentifier (string with pattern),* (§5.1.125 on page 68); use="required"
BulkData	*Type ase:ElectricityStandingDataBDT (complex)*, (§5.1.48on page 47); maxOccurs="unbounded"
BulkData /BDTGroupings	*Type ase:BDTGroupings (complex)*, (§5.1.29 on page 35)
BulkData /BDTGroupings /BDTGrouping	*Type ase:BDTGrouping (complex)*, (§5.1.28 on page 34); maxOccurs="unbounded"
BulkData /NMIStandingData	*Type ase:ElectricityStandingData* (§5.1.47 on page 46) implementing (using xsi:type) *(abstract) Type ase:NMIStandingData* (complex) (§5.1.119 on page 66)
BulkData /NMIStandingData /DataStreams	*Type ase:ElectricityDataStreams (complex)*, (§5.1.40 on page 39); minOccurs="0"
BulkData /NMIStandingData /MasterData	*Type ase:ElectricityMasterStandingData (complex)* (§5.1.41 on page 39); minOccurs="0"
BulkData /NMIStandingData /MeterRegister	*Type ase:ElectricityMeters (complex)* (§5.1.46 on page 46) ; minOccurs="0"
BulkData /NMIStandingData/NMI	*Type ase:NMI (complex)* (§5.1.98 on page 62); minOccurs="0"
BulkData /NMIStandingData /RoleAssignments	*Type ase:RoleAssignments (complex)* (§5.1.129 on page 70); minOccurs="0"

XPath to aseXML node	aseXML type and restrictions
BulkData /NMIStandingData @effectiveDateTime	xsd:dateTime; use="optional"
BulkData /NMIStandingData @version	r25—*Type ase:ReleaseIdentifier (string with pattern)* (§5.1.125 on page 68); use="optional"

4.2 CATSBulkDataResponse

The element in aseXML implementing the BDT response transaction has the XPath *aseXML/ Transactions/Transaction/CATSBulkDataResponse*, with the short name of ase:CATSBulkDataResponse.

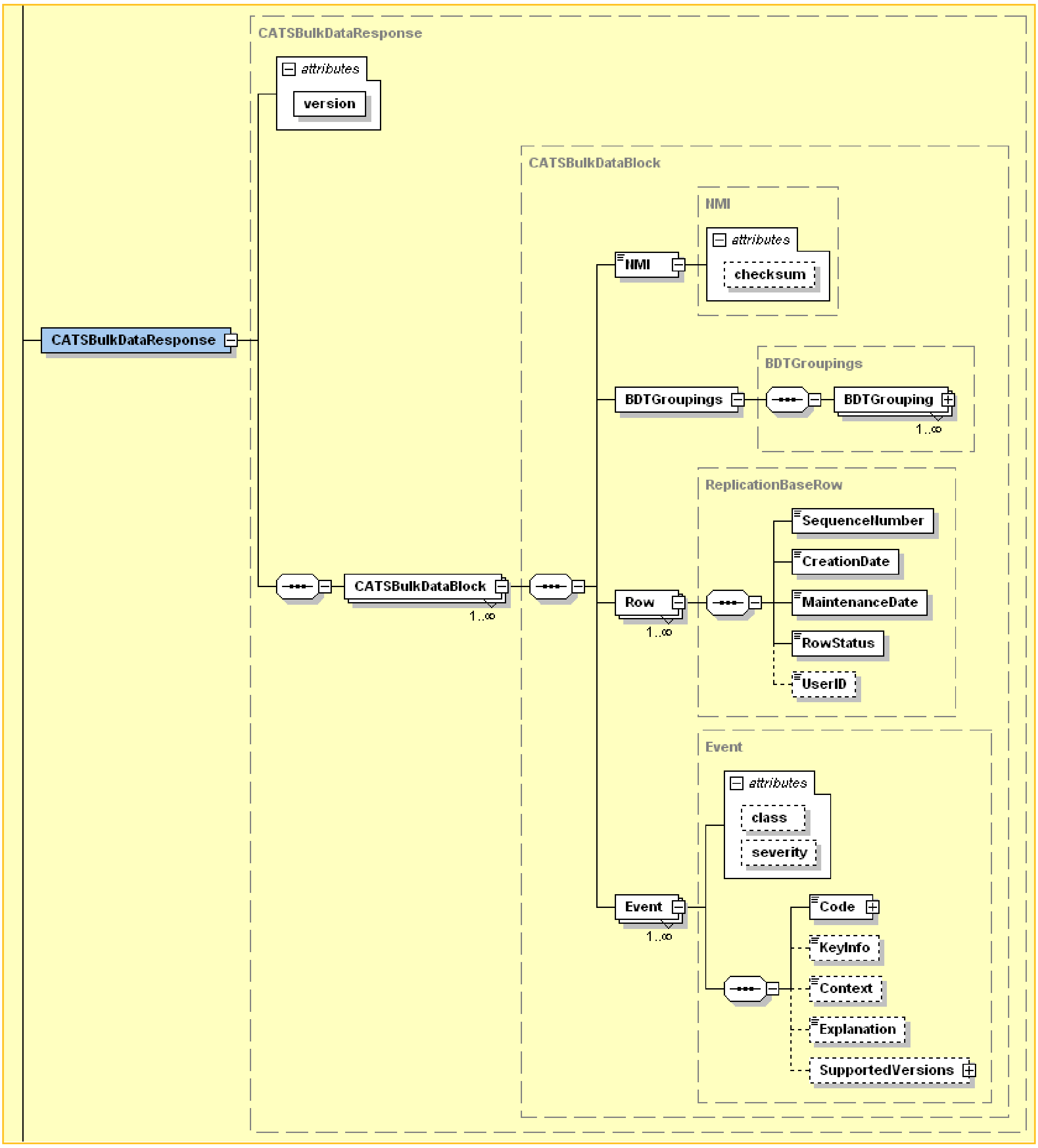


Figure 5: ase:CATSBulkDataResponse element

For the BDT response transaction, the following table lists 2 levels of aseXML nodes in alphabetical sequence, with each XPath relative to ase:CATSBulkDataResponse.

Table 5: aseXML nodes in ase:CATSBulkDataResponse

XPath to aseXML node	aseXML type and restrictions
.	*Type ase:CATSBulkDataResponse (complex)*, (§5.1.32 on page 36)
@version	r9—*Type ase:ReleaseIdentifier (string with pattern)* (§5.1.125 on page 68); use="required"

XPath to aseXML node	aseXML type and restrictions
CATSBulkDataBlock	*Type ase:CATSBulkDataBlock (complex)*, (§5.1.30 on page 35); maxOccurs="unbounded"
CATSBulkDataBlock /BDTGroupings	*Type ase:BDTGroupings (complex)*, (§5.1.29 on page 35)
CATSBulkDataBlock /Event	*Type ase:Event (complex)* (§5.1.54 on page 50); maxOccurs="unbounded"
CATSBulkDataBlock /NMI	*Type ase:NMI (complex)* (§5.1.98 on page 62)
CATSBulkDataBlock /Row	*Type ase:ReplicationBaseRow (complex)*, (§5.1.126 on page 69); maxOccurs="unbounded"

4.3 Acknowledgements

The element in aseXML implementing the BDT acknowledgement has the XPath of /aseXML /Acknowledgements, with the short name of ase:Acknowledgements. There are two parts of a BD acknowledgement:

- Message level, uses the XPath of /aseXML/Acknowledgements/MessageAcknowledgement, with the short name of ase:MessageAcknowledgement.
- Transaction level, uses the XPath of /aseXML/Acknowledgements /TransactionAcknowledgement, with the short name of ase:TransactionAcknowledgement.

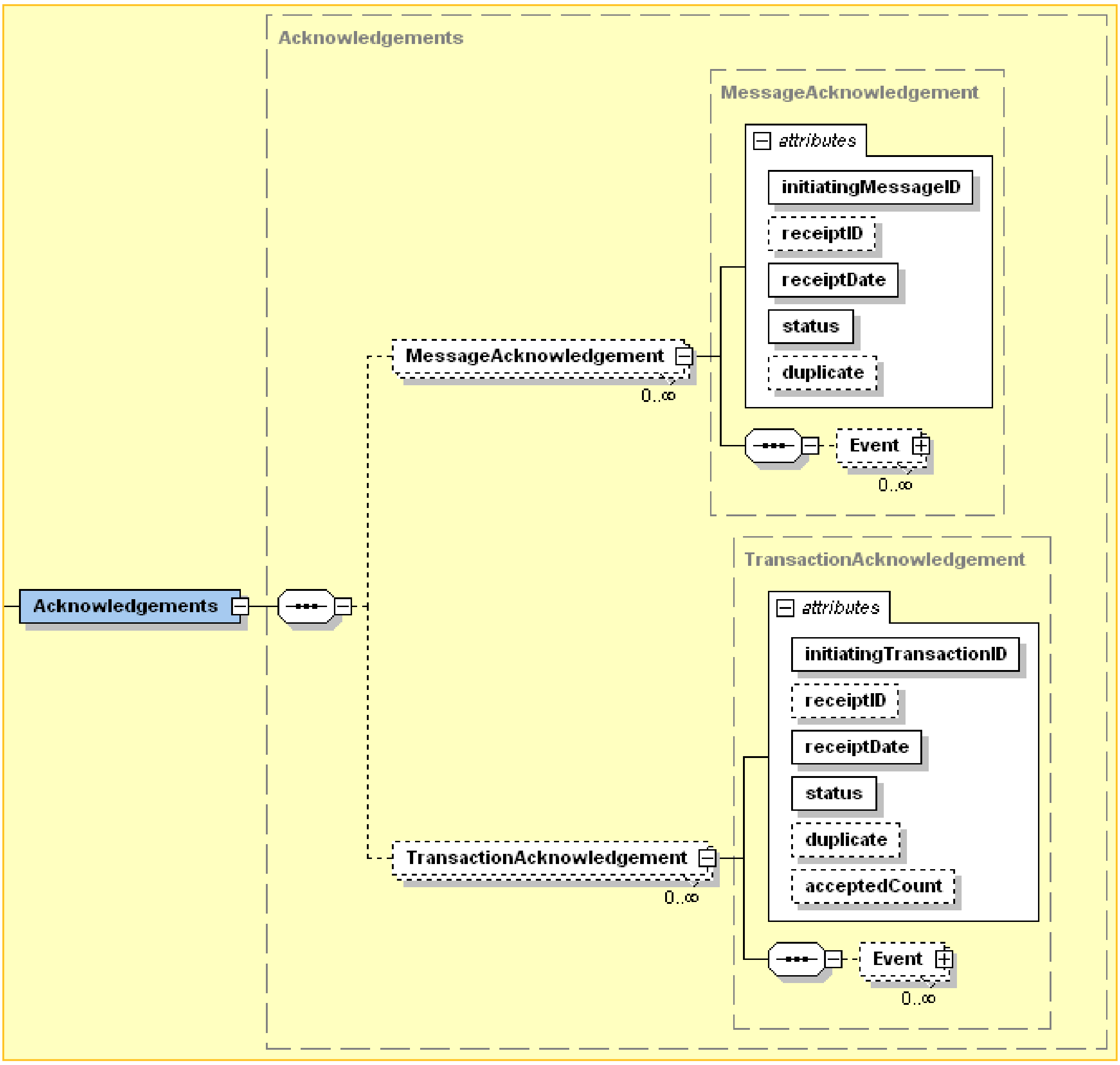


Figure 6: ase:Acknowledgements element down to Event element

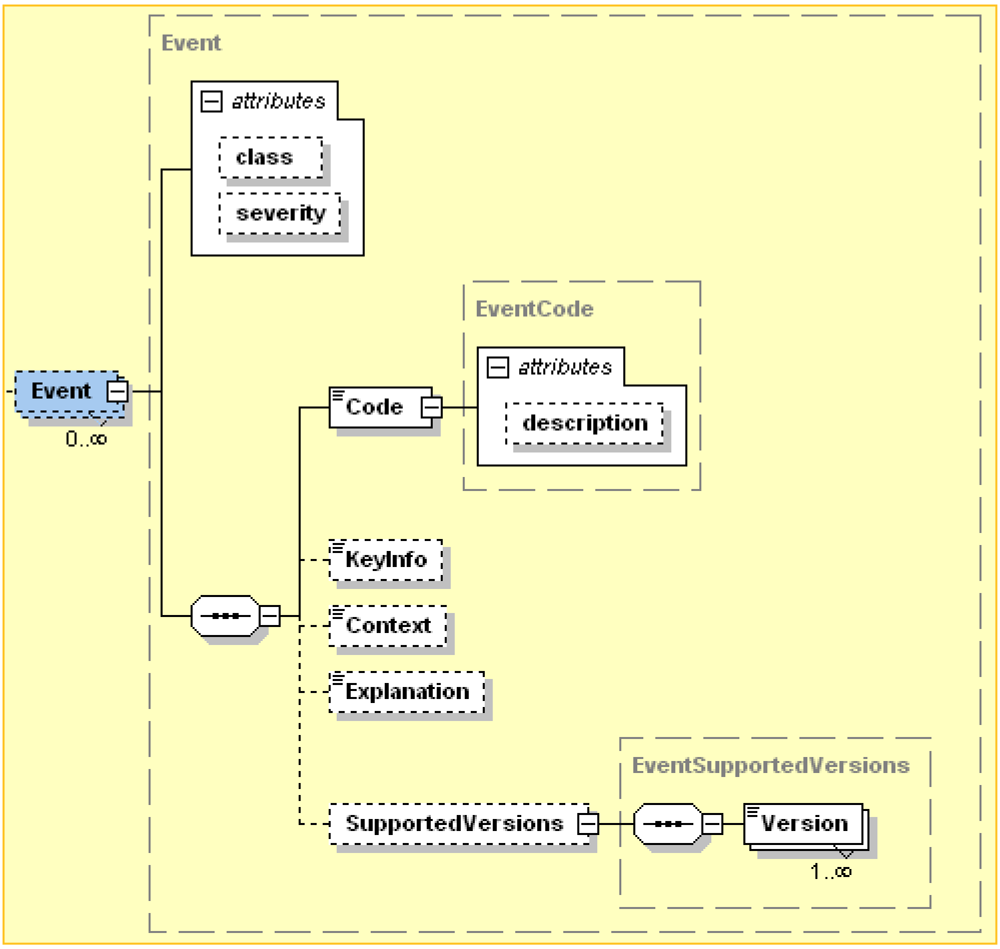


Figure 7: ase:Event element fully expanded

For the BDT acknowledgement, the following table lists some aseXML nodes in alphabetical sequence, with each XPath relative to ase:Acknowledgements.

Table 6: aseXML nodes in ase:Acknowledgements

XPath to aseXML node	aseXML type and restrictions
.	*Type ase:Acknowledgements (complex)* (§5.1.2 on page 17)
/aseXML	*Type ase:Envelope (complex)* (§5.1.53 on page 49)
/aseXML/Header	*Type ase:Header (complex)* (§5.1.63 on page 53)
/aseXML/Header/MessageDate	xsd:dateTime
/aseXML/Header/MessageID	*Type ase:MessageIdentifier* (string, 1-36 chars.) (§5.1.68 on page 56)
/aseXML/Header/Priority	*Type ase:TransactionPriority (enumerated)* (§5.1.135 on page 76), minOccurs="0"
/aseXML/Header /TransactionGroup	*Type ase:TransactionGroup (enumerated)* (§5.1.133 on page 74)
MessageAcknowledgement	*Type ase:MessageAcknowledgement (complex)* (§5.1.67 on page 55)
MessageAcknowledgement /Event	*Type ase:Event (complex)* (§5.1.54 on page 50), minOccurs="0" maxOccurs="unbounded"
MessageAcknowledgement /Event/Code	*Type ase:EventCode (complex)* (§5.1.56 on page 51)
MessageAcknowledgement /Event/Code@description	xsd:string, use="optional"

XPath to aseXML node	aseXML type and restrictions
MessageAcknowledgement /Event/Context	*Type ase:EventContext (string ≤ 80 chars.)* (§5.1.58 on page 52), minOccurs="0"
MessageAcknowledgement /Event/Explanation	xsd:string, minOccurs="0"
MessageAcknowledgement /Event/KeyInfo	*Type ase:EventKeyInfo (string ≤ 80 chars.)* (§5.1.59 on page 52), minOccurs="0"
MessageAcknowledgement /Event@class	*Type ase:EventClass (enumerated)* (§5.1.55 on page 51), use="optional" default="Application"
MessageAcknowledgement /Event@severity	*Type ase:EventSeverity (enumerated)* (§5.1.60 on page 52), use="optional" default="Fatal"
MessageAcknowledgement @duplicate	*Type ase:YesNo (enumerated list)* (§5.1.141 on page 77), default="No"
MessageAcknowledgement @initiatingMessageID	*Type ase:MessageIdentifier* (string, 1-36 chars.) (§5.1.68 on page 56), use="required"
MessageAcknowledgement @receiptDate	xsd:dateTime, use="required"
MessageAcknowledgement @receiptID	*Type ase:ReceiptIdentifier (string, 1-36 chars.)* (§5.1.124 on page 68), use="optional"
MessageAcknowledgement @status	*Type ase:MessageStatus (enumerated)* (§5.1.70 on page 57), use="required"
TransactionAcknowledgement /Event	*Type ase:Event (complex)* (§5.1.54 on page 50), minOccurs="0" maxOccurs="unbounded"
TransactionAcknowledgement /Event/Code	*Type ase:EventCode (complex)* (§5.1.56 on page 51)
TransactionAcknowledgement /Event/Code@description	xsd:string, use="optional"
TransactionAcknowledgement /Event/Context	*Type ase:EventContext (string ≤ 80 chars.)* (§5.1.58 on page 52), minOccurs="0"
TransactionAcknowledgement /Event/Explanation	xsd:string, minOccurs="0"
TransactionAcknowledgement /Event/KeyInfo	*Type ase:EventKeyInfo (string ≤ 80 chars.)* (§5.1.59 on page 52), minOccurs="0"
TransactionAcknowledgement /Event@class	*Type ase:EventClass (enumerated)* (§5.1.55 on page 51), use="optional" default="Application"
TransactionAcknowledgement /Event@severity	*Type ase:EventSeverity (enumerated)* (§5.1.60 on page 52), use="optional" default="Fatal"
TransactionAcknowledgement @acceptedCount	xsd:nonNegativeInteger, use="optional"
TransactionAcknowledgement @duplicate	*Type ase:YesNo (enumerated list)* (§5.1.141 on page 77), default="No"
TransactionAcknowledgement @initiatingTransactionID	*Type ase:TransactionIdentifier (string, 1-36 chars.)* (§5.1.134 on page 75), use="required"
TransactionAcknowledgement @receiptDate	xsd:dateTime, use="required"
TransactionAcknowledgement @receiptID	*Type ase:ReceiptIdentifier (string, 1-36 chars.)* (§5.1.124 on page 68), use="optional"
TransactionAcknowledgement @status	*Type ase:TransactionStatus (enumerated)* (§5.1.137 on page 76), use="required"

5 AseXML types

The aseXML types are in alphabetical sequence, to assist referencing. For an index, refer to the Contents.

These are the types used as the aseXML implementation of the Bulk Data Tool (BDT), including all parts down to fundamental XML types.

5.1.1 Type ase:AccessDetail (string ≤ 160 chars.)

The aseXML documentation for type ase:AccessDetail is:

Classification of High Voltage distribution line feeding property.

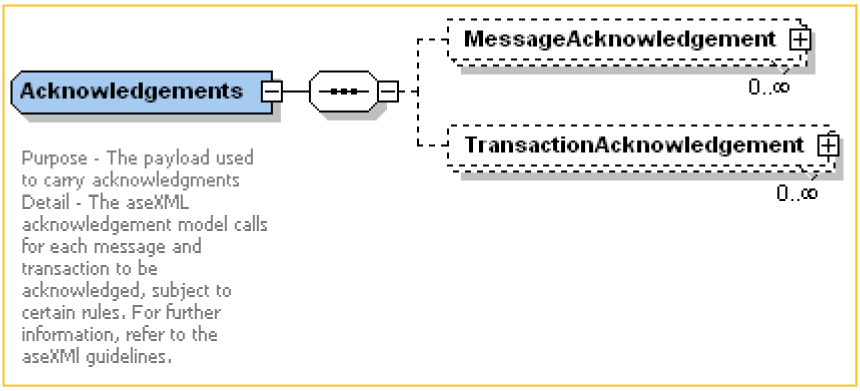
XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AccessDetail	maxLength value="160"	xsd:string

5.1.2 Type ase:Acknowledgements (complex)

The aseXML documentation for type ase:Acknowledgements is:

Purpose - The payload used to carry acknowledgments

Detail - The aseXML acknowledgement model calls for each message and transaction to be acknowledged, subject to certain rules. For further information, refer to the aseXMl guidelines.


Acknowledgements
MessageAcknowledgement
0..∞
TransactionAcknowledgement
0..∞
Purpose - The payload used to carry acknowledgments Detail - The aseXML acknowledgement model calls for each message and transaction to be acknowledged, subject to certain rules. For further information, refer to the aseXMl guidelines.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:Acknowledgements are:

XPath to aseXML node	aseXML node restrictions	aseXML type

MessageAcknowledgement	minOccurs="0"
	maxOccurs="unbounded"
		Type ase:MessageAcknowledgement (complex) (§5.1.67 on page 55)

TransactionAcknowledgement	minOccurs="0"
	maxOccurs="unbounded"
		Type ase:TransactionAcknowledgement (complex) (§5.1.132 on page 73)

5.1.3 Type ase:ActiveInactive (enumerated list)

The aseXML documentation for type ase:ActiveInactive is:

Purpose - Active/Inactive indication

Type ase:ActiveInactive has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- A

- l

5.1.4 Type ase:AustralianAddressLine (string, ≤ 80 chars.)

The aseXML documentation for type ase:AustralianAddressLine is:

Purpose - Define a line of unstructured Australian address information

Detail - An unstructured format allows for legacy or non-specific address/location information to be carried along with structured addresses within a common address container.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianAddressLine	maxLength value="80"	xsd:string

5.1.5 Type ase:AustralianBuildingOrPropertyName (string, 1-30 chars.)

The aseXML documentation for type ase:AustralianBuildingOrPropertyName is:

Purpose - Define building or property name as per Australian Standard AS4590

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianBuildingOrPropertyName	minLength value="1", maxLength value="30"	xsd:string

5.1.6 Type ase:AustralianDeliveryPointIdentifier (integer 10000000-99999999 incl.)

The aseXML documentation for type ase:AustralianDeliveryPointIdentifier is:

Purpose - Define Australian delivery point identifier as per Australian Standard AS4590/Australia Post AMAS standard

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianDeliveryPointIdentifier	minInclusive value="10000000", maxInclusive value="99999999"	xsd:nonNegativeInteger

5.1.7 Type ase:AustralianFlatOrUnitNumber (string with pattern)

The aseXML documentation for type ase:AustralianFlatOrUnitNumber is:

Purpose - Define flat or unit number as per Australian Standard AS4590

Detail - Note that alphabetic characters are permitted allowing for "100A" etc.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianFlatOrUnitNumber	pattern value="[\p{L}\p{N}\p{P}\s]{1,7}"	xsd:string

5.1.8 Type ase:AustralianFlatOrUnitType (enumerated list)

The aseXML documentation for type ase:AustralianFlatOrUnitType is:

Purpose - Define flat or unit types as per Australian Standard AS4590
Type ase:AustralianFlatOrUnitType has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- APT.
- CTGE.
- DUP.

- FY.
- F.
- HSE.
- KSK.
- MSNT.
- MB.
- OFF.
- PTHS.
- RM.
- SHED.
- SHOP.
- SITE.
- SL.
- STU.
- SE.
- TNHS.
- U.
- VLLA.
- WARD.
- WE

5.1.9 Type ase:AustralianFloorOrLevelNumber (string with pattern)

The aseXML documentation for type ase:AustralianFloorOrLevelNumber is:

Purpose - Define floor or level number as per Australian Standard AS4590

Detail - Note that alphabetic characters are permitted.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianFloorOrLevelNumber	pattern value="[\p{L}\p{N}\p{P}\s]{1,5}"	xsd:string

5.1.10 Type ase:AustralianFloorOrLevelType (enumerated list)

The aseXML documentation for type ase:AustralianFloorOrLevelType is:

Purpose - Define floor or level types as per Australian Standard AS4590

Type ase:AustralianFloorOrLevelType has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- B.
- FL.
- G.
- L.
- LG.
- M.
- UG.

5.1.11 Type ase:AustralianHouseNumber (integer, 0-99999 incl.)

The aseXML documentation for type ase:AustralianHouseNumber is:

Purpose - Define house number as per Australian Standard AS4590

Detail - Note that any alphabetic information should be included in the house number suffix

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianHouseNumber	maxInclusive value="99999"	xsd:nonNegativeIntege

5.1.12 Type ase:AustralianHouseNumberSuffix (string with pattern)

The aseXML documentation for type ase:AustralianHouseNumberSuffix is:

Purpose - Define house number suffix as per Australian Standard AS4590

Detail - Any non-numeric information should be entered here

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianHouseNumberSuffix	pattern value="[\p{L}\p{N}]{1}"	xsd:string

5.1.13 Type ase:AustralianLocationDescriptor (string with pattern)

The aseXML documentation for type ase:AustralianLocationDescriptor is:

Purpose - Define location descriptor as per Australian Standard AS4590

Detail - This is a "catch all" field for non-standard address information.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianLocationDescriptor	pattern value="[\p{L}\p{N}\p{P}\s]{1,30}"	xsd:string

5.1.14 Type ase:AustralianLotNumber (string with pattern)

The aseXML documentation for type ase:AustralianLotNumber is:

Purpose - Define lot number as per Australian Standard AS4590

XPath to aseXML node	aseXML node restrictions	aseXML type

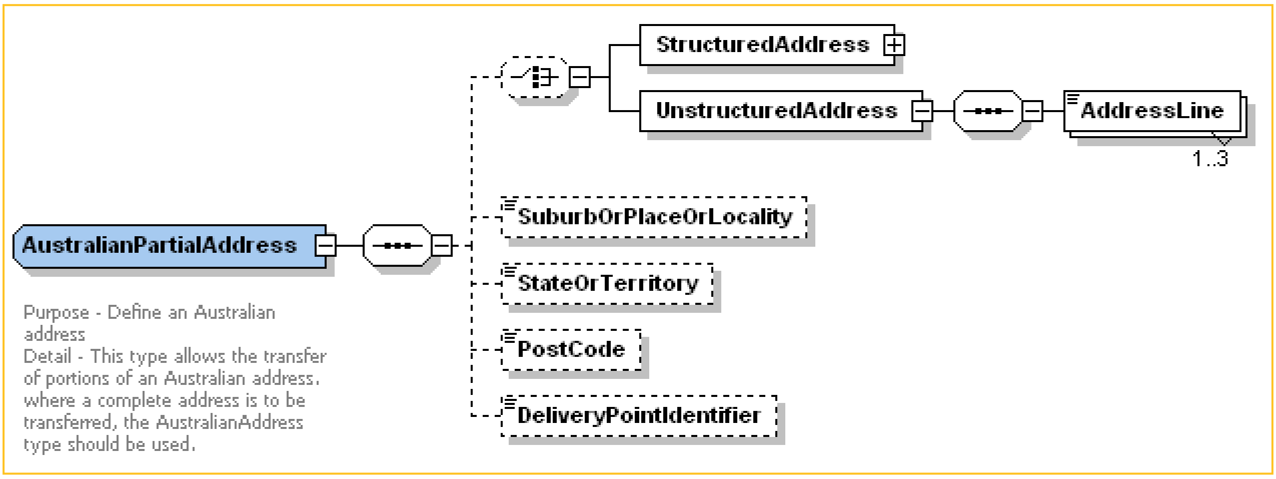
XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianLotNumber	pattern value="[\p{L}\p{N}\p{P}\s]{1,6}"	xsd:string

5.1.15 Type ase:AustralianPartialAddress (complex)

The aseXML documentation for type ase:AustralianPartialAddress is:

Purpose - Define an Australian address

Detail - This type allows the transfer of portions of an Australian address. where a complete address is to be transferred, the AustralianAddress type should be used.


StructuredAddress
UnstructuredAddress
AddressLine
1..3
SuburbOrPlaceOrLocality
AustralianPartialAddress
StateOrTerritory
PostCode
DeliveryPointIdentifier
Purpose - Define an Australian address
Detail - This type allows the transfer of portions of an Australian address. where a complete address is to be transferred, the AustralianAddress type should be used.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:AustralianPartialAddress are:

XPath to aseXML node	aseXML node restrictions	aseXML type

DeliveryPointIdentifier	nillable="true"
	minOccurs="0"
		Type ase:AustralianDeliveryPointIdentifier (integer 10000000-99999999 incl.) (§5.1.6 on page 18)

PostCode	nillable="true"
	minOccurs="0"
		Type ase:AustralianPostCode (string with pattern) (§5.1.20 on page 23)

StateOrTerritory	nillable="true"
	minOccurs="0"
		Type ase:AustralianStateOrTerritory (§5.1.21 on page 23)

StructuredAddress		*Type ase:AustralianStructuredAddressPartialComponents (complex)* (§5.1.25 on page 32)

SuburbOrPlaceOrLocality	nillable="true"
	minOccurs="0"
		Type ase:AustralianSuburbOrPlaceOrLocality (string ≤ 46 chars.) (§5.1.26 on page 34)

UnstructuredAddress		xsd:complexType, sequence; see Note 0 below

Notes:

The ase:UnstructuredAddress is a sequence of ase:AddressLine elements, where:

XPath to aseXML node	aseXML node restrictions	aseXML type

AddressLine	nillable="true"
	maxOccurs="3"
		Type ase:AustralianAddressLine (string, ≤ 80 chars.) (§5.1.4 on page 18)

5.1.16 Type ase:AustralianPostalDeliveryNumberPrefix (string with pattern)

The aseXML documentation for type ase:AustralianPostalDeliveryNumberPrefix is:

Purpose - Define postal delivery number prefix as per Australian Standard AS4590

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianPostalDeliveryNumberPrefix	pattern value="[\p{Lu}]{1,3}"	xsd:string

5.1.17 Type ase:AustralianPostalDeliveryNumberSuffix (string with pattern)

The aseXML documentation for type ase:AustralianPhoneNumber is:

Purpose - Define postal delivery number suffix as per Australian Standard AS4590

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianPostalDeliveryNumberSuffix	pattern value="[\p{Lu}]{1,3}"	xsd:string

5.1.18 Type ase:AustralianPostalDeliveryNumberValue (integer, 0-99999 incl.)

The aseXML documentation for type ase:AustralianPostalDeliveryNumberValue is:

Purpose - Define postal delivery number value as per Australian Standard AS4590

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianPostalDeliveryNumberValue	maxInclusive value="99999"	xsd:nonNegativeInteger

5.1.19 Type ase:AustralianPostalDeliveryType (enumerated list)

The aseXML documentation for type ase:AustralianPostalDeliveryType is:

Purpose - Define postal delivery types as per Australian Standard AS4590
Detail - There are several additions in this list over and above what is in AS4590. The extra types have been derived from the Australia Post AMAS standard.
Type ase:AustralianStateOrTerritory has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- CARE PO.
- CMA.
- CMB.
- CPA.
- GPO BOX.
- LOCKED BAG.
- MS.
- PO BOX.
- PRIVATE BAG.
- RSD.
- RMB.
- RMS.

5.1.20 Type ase:AustralianPostCode (string with pattern)

The aseXML documentation for type ase:AustralianPostCode is:

Purpose - Define Australian postcode as per Australian Standard AS4590

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianPostCode	pattern value="[\p{N}]{4}"	xsd:string

5.1.21 Type ase:AustralianStateOrTerritory (enumerated list)

The aseXML documentation for type ase:AustralianStateOrTerritory is:

Purpose - Define Australian states and territories as per Australian Standard AS4590

Type ase:AustralianStateOrTerritory has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- AAT.
- ACT.
- NSW.
- NT.
- QLD.
- SA.
- TAS.
- VIC.
- WA.

5.1.22 Type ase:AustralianStreetName (string with pattern)

The aseXML documentation for type ase:AustralianStreetName is:

Purpose - Define street name as per Australian Standard AS4590

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianStreetName	pattern value="[\p{L}\p{N}\s\-']{1,30}"	xsd:string

5.1.23 Type ase:AustralianStreetSuffix (enumerated list)

The aseXML documentation for type ase:AustralianStreetSuffix is:

Purpose - Define street suffixes as per Australian Standard AS4590

Type ase:AustralianStreetSuffix has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- CN.
- E.

- EX.
- LR.
- N.
- NE.
- NW.
- S.
- SE.
- SW.
- UP.
- W.

5.1.24 Type ase:AustralianStreetType (enumerated list)

The aseXML documentation for type ase:AustralianStreetType is:

Purpose - Define street types as per Australian Standard AS4590.

Detail - There are several additions in this list over and above those defined in AS4590. These additions are defined as follows:

-- BOWL - Bowl

-- CRSE - Course

-- GTWY - Gateway

-- HETH - Heath

-- HUB - Hub

-- ISLD - Island

-- PRST - Pursuit

-- PSGE - Passage

-- RTRN - Return

-- WOOD - Wood

-- WTRS - Waters

Version 2.0 addition

-- VLL - Villa

Type ase:AustralianStreetType has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- ACCS.
- ALLY.
- ALWY.
- AMBL.

- ANCG.
- APP.
- ARC.
- ART.
- AVE.
- BASN.
- BCH.
- BEND.
- BLK.
- BVD.
- BOWL.
- BRCE.
- BRAE.
- BRK.
- BDGE.
- BDWY.
- BROW.
- BYPA.
- BYWY.
- CAUS.
- CRSE.
- CTR.
- CNWY.
- CH.
- CIR.
- CLT.
- CCT.
- CRCS.
- CL.

- CLDE.
- CMMN.
- CON.
- CPS.
- CNR.
- CSO.
- CT.
- CTYD.
- COVE.
- CRES.
- CRST.
- CRSS.
- CRSG.
- CRD.
- COWY.
- CUWY.
- CDS.
- CTTG.
- DALE.
- DELL.
- DEVN.
- DIP.
- DSTR.
- DR.
- DRWY.
- EDGE.
- ELB.
- END.
- ENT.

- ESP.
- EST.
- EXP.
- EXTN.
- FAWY.
- FTRK.
- FITR.
- FLAT.
- FOLW.
- FTWY.
- FSHR.
- FORM.
- FWY.
- FRNT.
- FRTG.
- GAP.
- GDN.
- GDNS.
- GTE.
- GTES.
- GTWY.
- GLD.
- GLEN.
- GRA.
- GRN.
- GRND.
- GR.
- GLY.
- HETH.

- HTS.
- HRD.
- HUB.
- HWY.
- HILL.
- INTG.
- INTN.
- ISLD.
- JNC.
- KEY.
- LDG.
- LANE.
- LNWY.
- LEES.
- LINE.
- LINK.
- LT.
- LKT.
- LOOP.
- LWR.
- MALL.
- MNDR.
- MEW.
- MEWS.
- MWY.
- MT.
- NOOK.
- OTLK.
- PDE.

- PARK.
- PKLD.
- PKWY.
- PART.
- PASS.
- PATH.
- PHWY.
- PIAZ.
- PL.
- PLAT.
- PLZA.
- PKT.
- PNT.
- PORT.
- PROM.
- PRST.
- PSGE.
- QUAD.
- QDGL.
- QDRT.
- QY.
- QYS.
- RMBL.
- RAMP.
- RNGE.
- RCH.
- RES.
- REST.
- RTT.

- RIDE.
- RDGE.
- RGWY.
- ROWY.
- RING.
- RISE.
- RVR.
- RVWY.
- RVRA.
- RD.
- RDS.
- RDSD.
- RDWY.
- RNDE.
- RSBL.
- RTY.
- RND.
- RTE.
- RTRN.
- ROW.
- RUE.
- RUN.
- SWY.
- SDNG.
- SLPE.
- SND.
- SPUR.
- SQ.
- STRS.

- SHWY.
- STPS.
- STRA.
- ST.
- STRP.
- SBWY.
- TARN.
- TCE.
- THOR.
- TLWY.
- TOP.
- TOR.
- TWRS.
- TRK.
- TRL.
- TRLR.
- TRI.
- TKWY.
- TURN.
- UPAS.
- UPR.
- VALE.
- VDCT.
- VIEW.
- VLL.
- VLLS.
- VSTA.
- WADE.
- WALK.

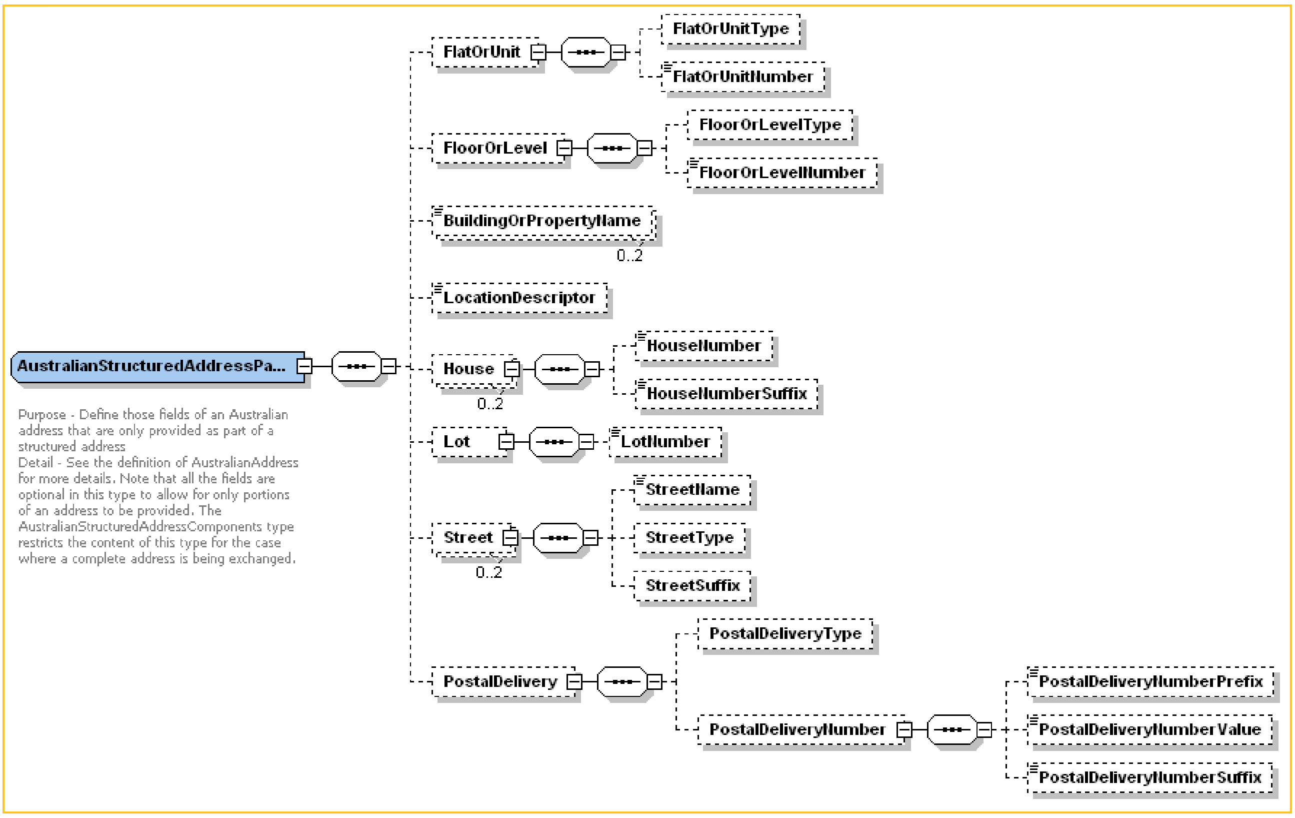
- WKWY.
- WAY.
- WHRF.
- WOOD.
- WTRS.
- WYND.
- YARD.

5.1.25 Type ase:AustralianStructuredAddressPartialComponents (complex)

The aseXML documentation for type ase:AustralianStructuredAddressPartialComponents is:

Purpose - Define those fields of an Australian address that are only provided as part of a structured address

Detail - See the definition of AustralianAddress for more details. Note that all the fields are optional in this type to allow for only portions of an address to be provided. The AustralianStructuredAddressComponents type restricts the content of this type for the case where a complete address is being exchanged.


FlatOrUnit
FlatOrUnitType
FlatOrUnitNumber
FloorOrLevel
FloorOrLevelType
FloorOrLevelNumber
BuildingOrPropertyName
0..2
LocationDescriptor
AustralianStructuredAddressPa...
Purpose - Define those fields of an Australian address that are only provided as part of a structured address
Detail - See the definition of AustralianAddress for more details. Note that all the fields are optional in this type to allow for only portions of an address to be provided. The AustralianStructuredAddressComponents type restricts the content of this type for the case where a complete address is being exchanged.
House
0..2
HouseNumber
HouseNumberSuffix
Lot
LotNumber
Street
0..2
StreetName
StreetType
StreetSuffix
PostalDelivery
PostalDeliveryType
PostalDeliveryNumber
PostalDeliveryNumberPrefix
PostalDeliveryNumberValue
PostalDeliveryNumberSuffix

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:AustralianStructuredAddressPartialComponents are:

XPath to aseXML node	aseXML node restrictions	aseXML type

BuildingOrPropertyName	nillable="true"
	minOccurs="0"
	maxOccurs="2"
		Type ase:AustralianBuildingOrPropertyName (§5.1.5 on page 18)

FlatOrUnit	minOccurs="0"	complexType, sequence; see Note 0 on page 33

XPath to aseXML node	aseXML node restrictions	aseXML type
FloorOrLevel	minOccurs="0"	complexType, sequence; see Note 0 below

House	minOccurs="0"
	maxOccurs="2"
		complexType, sequence; see Note 0 below

LocationDescriptor	nillable="true"
	minOccurs="0"
		Type ase:AustralianLocationDescriptor (§5.1.13 on page 20)

Lot	minOccurs="0"	complexType, sequence; see Note 0 below
PostalDelivery	minOccurs="0"	complexType, sequence; see Note 0 below

Street	minOccurs="0"
	maxOccurs="2"
		complexType, sequence; see Note 0 below

Notes:

Element ase:FlatOrUnit is a sequence containing:

XPath to aseXML node	aseXML node restrictions	aseXML type

FlatOrUnitNumber	nillable="true"
	minOccurs="0"
		Type ase:AustralianFlatOrUnitNumber (§5.1.7 on page 18)

FlatOrUnitType	nillable="true"
	minOccurs="0"
		Type ase:AustralianFlatOrUnitType (§5.1.8 on page 18)

Element ase:FloorOrLevel is a sequence containing:

XPath to aseXML node	aseXML node restrictions	aseXML type

FloorOrLevelNumber	nillable="true"
	minOccurs="0"
		Type ase:AustralianFloorOrLevelNumber (§5.1.9 on page 19)

FloorOrLevelType	nillable="true"
	minOccurs="0"
		Type ase:AustralianFloorOrLevelType (§5.1.10 on page 19)

Element ase:House is a sequence containing:

XPath to aseXML node	aseXML node restrictions	aseXML type

HouseNumber	nillable="true"
	minOccurs="0"
		Type ase:AustralianHouseNumber (§5.1.11 on page 20)

HouseNumberSuffix	nillable="true"
	minOccurs="0"
		Type ase:AustralianHouseNumberSuffix (§5.1.12 on page 20)

Element ase:Lot is a sequence containing:

XPath to aseXML node	aseXML node restrictions	aseXML type and restrictions

LotNumber	nillable="true"
	minOccurs="0"
		Type ase:AustralianLotNumber (§5.1.14 on page 20)

Element ase:Street is a sequence containing:

XPath to aseXML node	aseXML node restrictions	aseXML type

StreetName	nillable="true"
	minOccurs="0"
		Type ase:AustralianStreetName (§5.1.22 on page 23)

StreetSuffix	nillable="true"
	minOccurs="0"
		Type ase:AustralianStreetSuffix (§5.1.23 on page 23)

StreetType	nillable="true"
	minOccurs="0"
		Type ase:AustralianStreetType (§5.1.24 on page 24)

Element ase:PostalDelivery is a sequence containing:

XPath to aseXML node	aseXML node restrictions	aseXML type

XPath to aseXML node	aseXML node restrictions	aseXML type
PostalDeliveryNumber	nillable="true" minOccurs="0"	complexType, sequence; see Note 0 below
PostalDeliveryType	nillable="true" minOccurs="0"	*Type ase:AustralianPostalDeliveryType* (§5.1.19 on page 22)

Element ase:PostalDeliveryNumber is a sequence containing:

XPath to aseXML node	aseXML node restrictions	aseXML type
PostalDeliveryNumberPrefix	minOccurs="0"	*Type ase:AustralianPostalDeliveryNumberPrefix* (§5.1.16 on page 21)
PostalDeliveryNumberSuffix	minOccurs="0"	*Type ase:AustralianPostalDeliveryNumberSuffix* (§5.1.17 on page 22)
PostalDeliveryNumberValue	minOccurs="0"	*Type ase:AustralianPostalDeliveryNumberValue* (§5.1.18 on page 22)

5.1.26 Type ase:AustralianSuburbOrPlaceOrLocality (string ≤ 46 chars.)

The aseXML documentation for type ase:AustralianSuburbOrPlaceOrLocality is:

Purpose - Define suburb or locality as per Australian Standard AS4590

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AustralianSuburbOrPlaceOrLocality	maxLength value="46"	xsd:string

5.1.27 Type ase:AveragedDailyLoad (integer)

The aseXML documentation for type ase:AveragedDailyLoad is:

Purpose - The average daily load for a given data steam

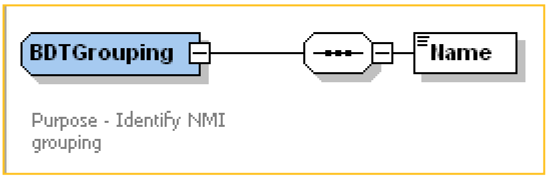
MSATS Data Model Column - AverageDailyLoad

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:AveragedDailyLoad		xsd:integer

5.1.28 Type ase:BDTGrouping (complex)

The aseXML documentation for type ase:ElectricityStandingDataBDT is:

Purpose - Identify NMI grouping



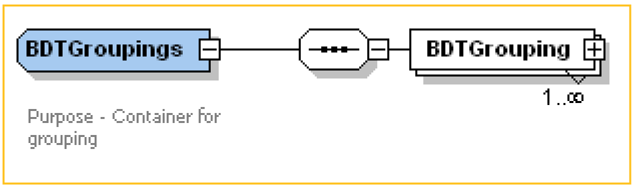
AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:BDTGrouping are:

XPath to aseXML node	aseXML node restrictions	aseXML type
Name		type="xsd:string"

5.1.29 Type ase:BDTGroupings (complex)

The aseXML documentation for type ase:ElectricityStandingDataBDT is:

Purpose - Container for grouping



AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:BDTGroupings are:

XPath to aseXML node	aseXML node restrictions	aseXML type
BDTGrouping	maxOccurs="unbounded"	*Type ase:BDTGroupings (complex)*, (§5.1.29 above)

5.1.30 Type ase:CATSBulkDataBlock (complex)

The aseXML documentation for type ase:CATSBulkDataBlock is:

Application - Bulk Data Tool

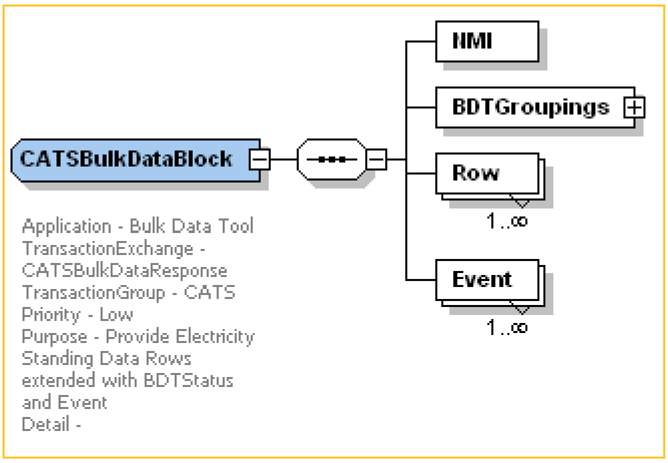
TransactionExchange - CATSBulkDataResponse

TransactionGroup - CATS

Priority - Low

Purpose - Provide Electricity Standing Data Rows extended with BDTStatus and Event

Detail -


CATSBulkDataBlock
NMI
BDTGroupings
Row
1..∞
Event
1..∞
Application - Bulk Data Tool
TransactionExchange - CATSBulkDataResponse
TransactionGroup - CATS
Priority - Low
Purpose - Provide Electricity Standing Data Rows extended with BDTStatus and Event
Detail -

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:CATSBulkDataBlock:

XPath to aseXML node	aseXML node restrictions	aseXML type
BDTGroupings		*Type ase:BDTGroupings (complex)*, (§5.1.29 above)
Event	maxOccurs="unbounded"	*Type ase:Event (complex)* (§5.1.54 on page 50)
NMI		*Type ase:NMI (complex)* (§5.1.98 on page 62)

XPath to aseXML node	aseXML node restrictions	aseXML type
Row	maxOccurs="unbounded"	*Type ase:ReplicationBaseRow (complex)*, (§5.1.126 on page 69)

5.1.31 Type ase:CATSBulkDataRequest (complex)

The aseXML documentation for type ase:CATSBulkDataRequest is:

Application - Bulk Data Tool

TransactionExchange - CATSBulkDataRequest

TransactionGroup - CATS

Priority - Low

Purpose - Provide the Electricity Standing Data for processing by Bulk Data Tool.

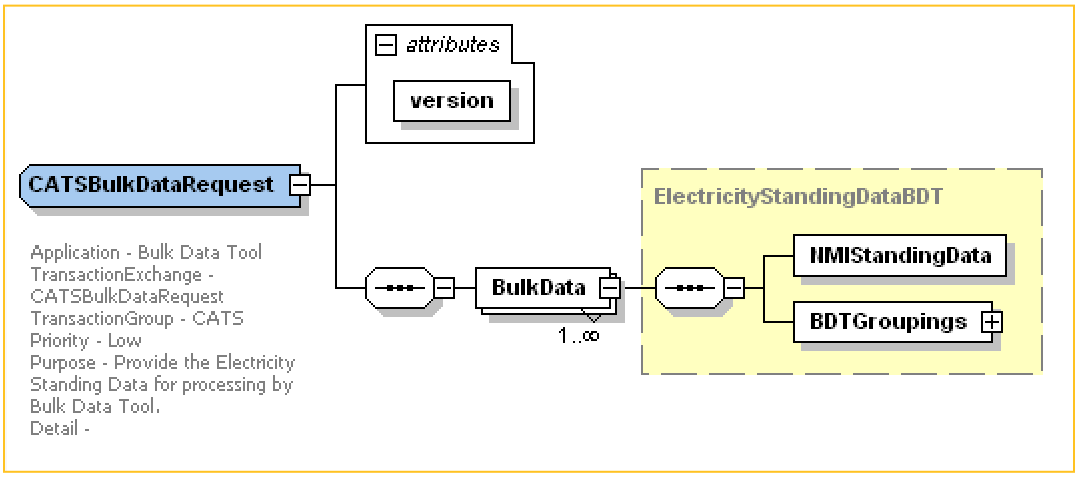
Detail –

Application - Separate application for Bulk Data Load - BDTLoadRequest

TransactionGroup - CATS

Priority - Medium

Purpose - Submit a standing data for multiple NMIs


attributes
version
CATSBulkDataRequest
Application - Bulk Data Tool
TransactionExchange - CATSBulkDataRequest
TransactionGroup - CATS
Priority - Low
Purpose - Provide the Electricity Standing Data for processing by Bulk Data Tool.
Detail -
BulkData
1..∞
ElectricityStandingDataBDT
NMIStandingData
BDTGroupings

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:CATSBulkDataRequest:

XPath to aseXML node	aseXML node restrictions	aseXML type
@version	use="required"	r9—*Type ase:ReleaseIdentifier (string with pattern)* (§5.1.125 on page 68)
BulkData	maxOccurs="unbounded"	*Type ase:ElectricityStandingDataBDT (complex)*, (§5.1.48on page 47)

5.1.32 Type ase:CATSBulkDataResponse (complex)

The aseXML documentation for type ase:CATSBulkDataResponse is:

Application - Bulk Data Tool

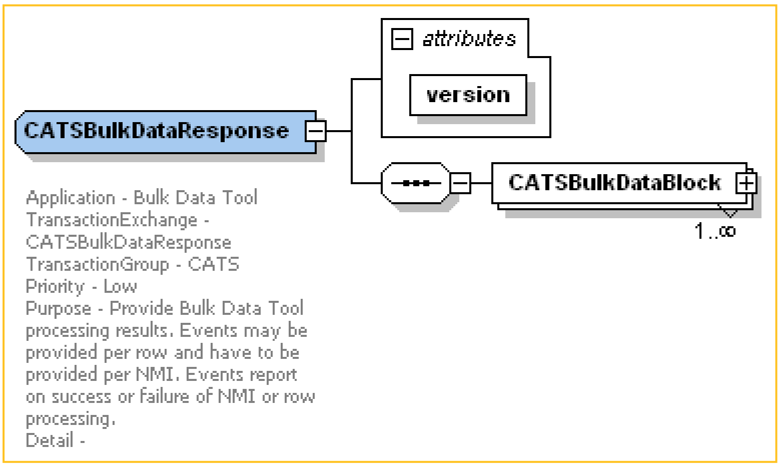
TransactionExchange - CATSBulkDataResponse

TransactionGroup - CATS

Priority - Low

Purpose - Provide Bulk Data Tool processing results. Events may be provided per row and have to be provided per NMI. Events report on success or failure of NMI or row processing.

Detail -


attributes
version
CATSBulkDataResponse
CATSBulkDataBlock
1..∞
Application - Bulk Data Tool
TransactionExchange - CATSBulkDataResponse
TransactionGroup - CATS
Priority - Low
Purpose - Provide Bulk Data Tool processing results. Events may be provided per row and have to be provided per NMI. Events report on success or failure of NMI or row processing.
Detail -

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:CATSBulkDataResponse:

XPath to aseXML node	aseXML node restrictions	aseXML type
@version	use="required"	r9—*Type ase:ReleaseIdentifier (string with pattern)* (§5.1.125 on page 68)
CATSBulkDataBlock	maxOccurs="unbounded"	*Type ase:CATSBulkDataBlock (complex)*, (§5.1.30 on page 35)

5.1.33 Type ase:CustomerFundedMeter (boolean)

The aseXML documentation for type ase:CustomerFundedMeter is:

Identifies that the customer has funded the purchase of the meter. This leads to a reduction in the meter charges allocated to the incumbent retailer, and so is needed for the network bill reconciliation. This also has an impact on contracts that can be offered to customers by prospective retailers.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:CustomerFundedMeter		xsd:boolean

5.1.34 Type ase:DataStreamType (enumerated list)

The aseXML documentation for type ase:DataStreamType is:

Purpose - Identify data stream type

MSATS Data Model Column - DataStreamType

Detail - Non-Interval has the same meaning as consumption.

Type ase:DataStreamType has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- Consumption.
- Interval.
- Non-Interval.

- Profile.

5.1.35 Type ase:DirectionIndicator (enumerated list)

The aseXML documentation for type ase:DirectionIndicator is:

This element may be handled using NetworkTariffCode, If so, the corresponding data element is to be removed.

Type ase:DirectionIndicator has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- Import.
- Export.

5.1.36 Type ase:DisplayType (string 1–20 chars.)

The aseXML documentation for type ase:DisplayType is:

In WA readings are collected from some customers by means of self-reader cards that have blank dials printed on them mimicking the layout of the meter display. This attribute defines the type of display on the meter and is used to define what style of self read card would be sent to the end-use customer when a retailer request them to be on a self read card arrangement. This also helps define the type of screen to display for internet captured readings.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:DisplayType	maxLength value="20" minLength value="1" whiteSpace value="collapse"	xsd:string

5.1.37 Type ase:DistanceFromSubstation (decimal, format = 9999.999)

The aseXML documentation for type ase:DistanceFromSubstation is:

For a number of network tariffs (generally larger customers), the distance to the zone substation is a factor in the calculation of the network access charges.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:DistanceFromSubstation	totalDigits value="7" fractionDigits value="3" maxInclusive value="9999.999"	xsd:decimal

5.1.38 Type ase:DistributionLossFactorCode (string ≤ 4 chars.)

The aseXML documentation for type ase:DistributionLossFactorCode is:

Purpose - Identify a distribution loss factor value

MSATS Data Model Column - DLFCode

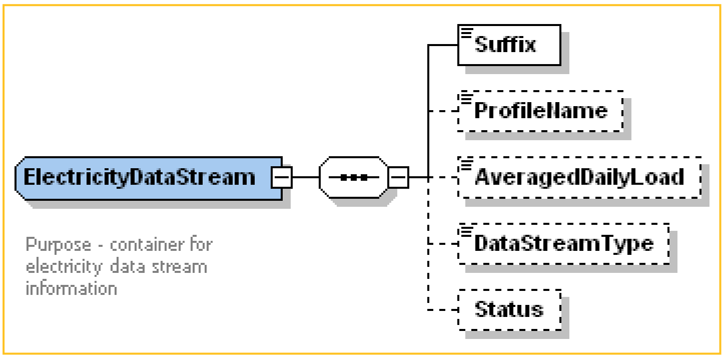
Detail - Whilst the current wholesale market identifies a loss factor value for each NMI, the number of NMIs in the retail market make this approach unwieldy. Multiple NMIs will have identical loss factors, and hence the need for a code to identify each value.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:DistributionLossFactorCode	maxLength value="4"	xsd:string

5.1.39 Type ase:ElectricityDataStream (complex)

The aseXML documentation for type ase:ElectricityDataStream is:

Purpose - container for electricity data stream information



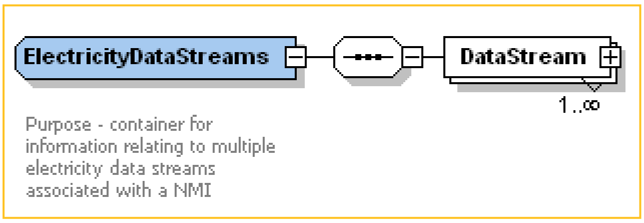
AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ElectricityDataStream are:

XPath to aseXML node	aseXML node restrictions	aseXML type
AveragedDailyLoad	nillable="true" minOccurs="0"	*Type ase:AveragedDailyLoad (integer)* (§5.1.27 on page 34)
DataStreamType	nillable="true" minOccurs="0"	*Type ase:DataStreamType* (§5.1.34 on page 37)
ProfileName	nillable="true" minOccurs="0"	*Type ase:ProfileName (string ≤ 10 chars.)* (§5.1.123 on page 68)
Status	nillable="true" minOccurs="0"	*Type ase:NMIStatusCode (string = 1 char.)* (§5.1.120 on page 67)
Suffix	nillable="true"	*Type ase:NMIDataStreamSuffix (string = 2 chars.)* (§5.1.118 on page 66)

5.1.40 Type ase:ElectricityDataStreams (complex)

The aseXML documentation for type ase:ElectricityDataStreams is:

Purpose - container for information relating to multiple electricity data streams associated with a NMI


ElectricityDataStreams
Purpose - container for information relating to multiple electricity data streams associated with a NMI
DataStream
1..∞

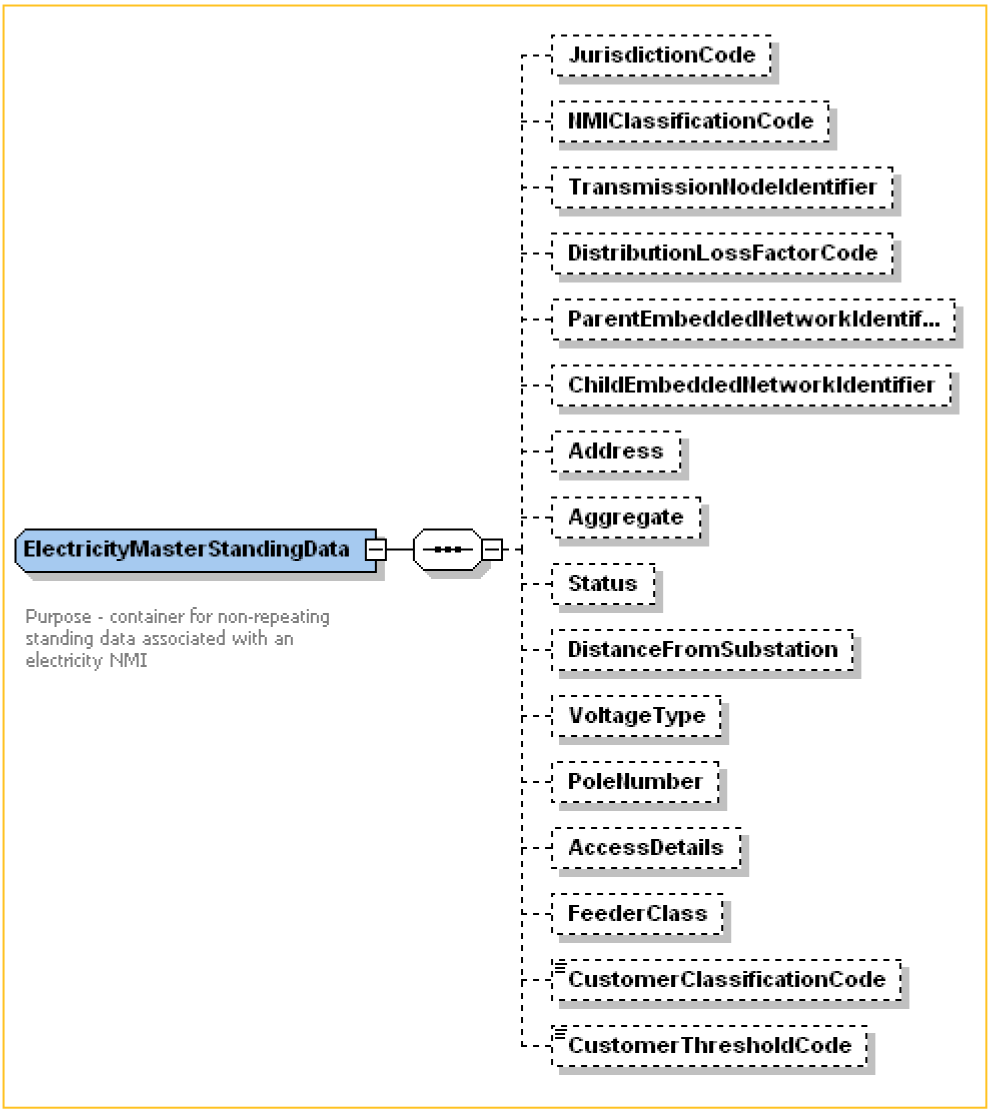
AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ElectricityDataStreams are:

XPath to aseXML node	aseXML node restrictions	aseXML type
DataStream	maxOccurs="unbounded"	*Type ase:ElectricityDataStream (complex)* (§5.1.39 on page 38)

5.1.41 Type ase:ElectricityMasterStandingData (complex)

The aseXML documentation for type ase:ElectricityMasterStandingData is:

Purpose - container for non-repeating standing data associated with an electricity NMI


ElectricityMasterStandingData
Purpose - container for non-repeating standing data associated with an electricity NMI
JurisdictionCode
NMIClassificationCode
TransmissionNodeIdentifier
DistributionLossFactorCode
ParentEmbeddedNetworkIdentif...
ChildEmbeddedNetworkIdentifier
Address
Aggregate
Status
DistanceFromSubstation
VoltageType
PoleNumber
AccessDetails
FeederClass
CustomerClassificationCode
CustomerThresholdCode

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ElectricityMasterStandingData are:

XPath to aseXML node	aseXML node restrictions	aseXML type

AccessDetails	nillable="true"
	minOccurs="0"
		Type ase:AccessDetail (string ≤ 160 chars.) (§5.1.1 on page 17)

Address	nillable="true"
	minOccurs="0"
		Type ase:AustralianPartialAddress (complex) (§5.1.15 on page 21)

Aggregate	nillable="true"
	minOccurs="0"
		Type ase:YesNo (enumerated list) (§5.1.141 on page 77)

ChildEmbeddedNetworkIdentifier	nillable="true"
	minOccurs="0"
		Type ase:EmbeddedNetworkIdentifier (string ≤ 10 chars.) (§5.1.49 on page 48)

CustomerClassificationCode	nillable="true"
	minOccurs="0"
		Type ase:EMSD_CustomerClassificationCode (string 1-20 chars.) (§5.1.50 on page 48)

CustomerThresholdCode	nillable="true"
	minOccurs="0"
		Type ase:EMSD_CustomerThresholdCode (string 1-20 chars.) (§5.1.51 on page 48)

DistanceFromSubstation	nillable="true"
	minOccurs="0"
		Type ase:DistanceFromSubstation (decimal, format = 9999.999) (§5.1.37 on page 38)

DistributionLossFactorCode	nillable="true"
	minOccurs="0"
		Type ase:DistributionLossFactorCode (string ≤ 4 chars.) (§5.1.38 on page 38)

XPath to aseXML node	aseXML node restrictions	aseXML type

FeederClass	nillable="true"
	minOccurs="0"
		Type ase:FeederClass (string 1-15 chars.) (§5.1.62 on page 53)

JurisdictionCode	nillable="true"
	minOccurs="0"
		Type ase:JurisdictionCode (string ≤ 3 chars.) (§5.1.65 on page 55)

NMIClassificationCode	nillable="true"
	minOccurs="0"
		Type ase:NMIClassificationCode (string ≤ 8 chars.) (§5.1.117 on page 66)

ParentEmbeddedNetworkIdentifier	nillable="true"
	minOccurs="0"
		Type ase:EmbeddedNetworkIdentifier (string ≤ 10 chars.) (§5.1.49 on page 48)

PoleNumber	nillable="true"
	minOccurs="0"
		Type ase:PoleNumber (string 1-40 chars.) (§5.1.122 on page 68)

Status	nillable="true"
	minOccurs="0"
		Type ase:NMIStatusCode (string = 1 char.) (§5.1.120 on page 67)

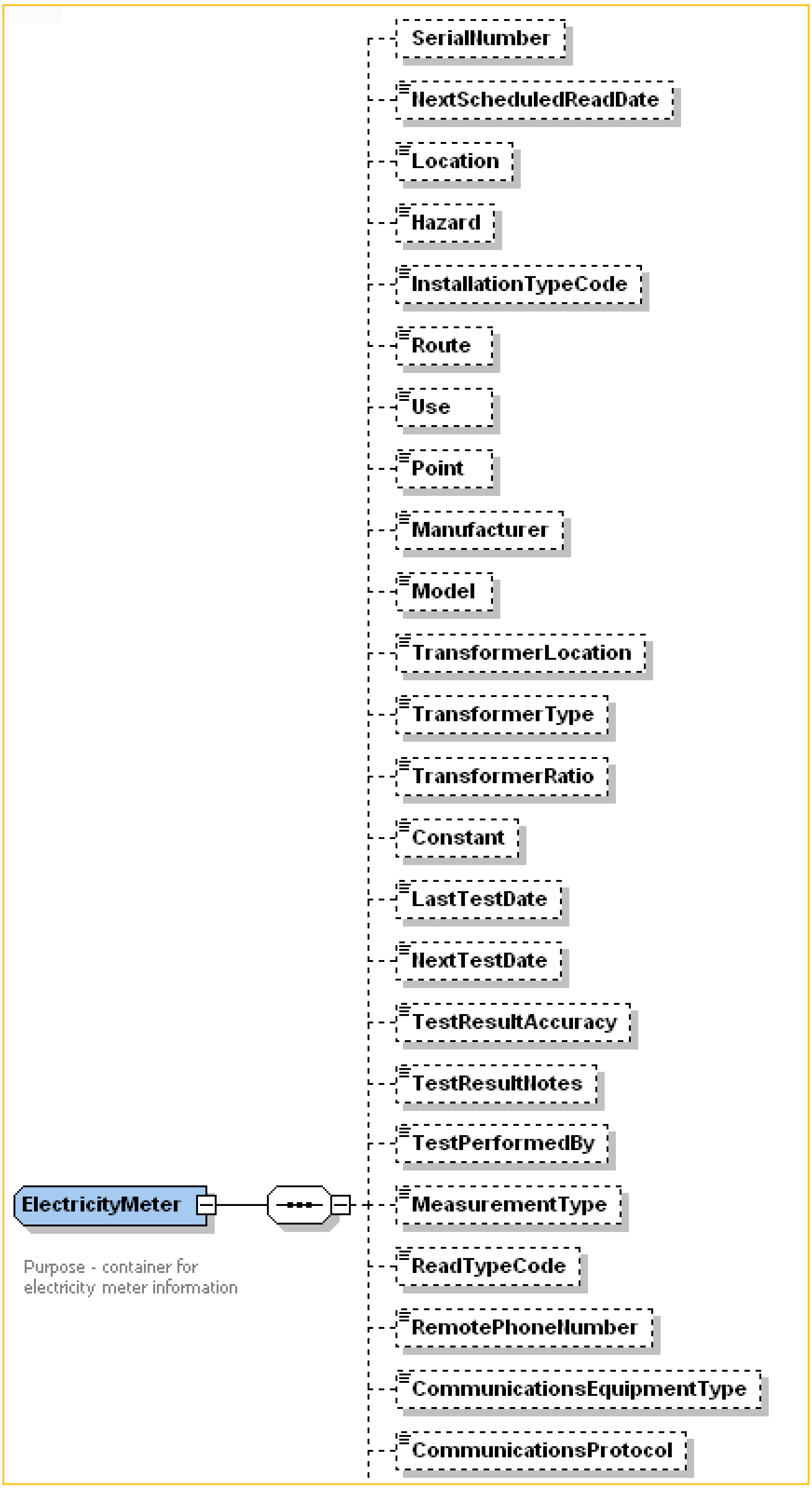
TransmissionNodeIdentifier	nillable="true"
	minOccurs="0"
		Type ase:TransmissionNodeIdentifier (string ≤ 4 chars.) (§5.1.138 on page 77)

VoltageType	nillable="true"
	minOccurs="0"
		Type ase:VoltageType (string 1-10 chars.) (§5.1.140 on page 77)

5.1.42 Type ase:ElectricityMeter (complex)

The aseXML documentation for type ase:ElectricityMeter is:

Purpose - container for electricity meter information



XML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ElectricityMeter are:

XPath to aseXML node	aseXML node restrictions	aseXML type

AdditionalSiteInformation	nillable="true"
	minOccurs="0"
		Type ase:MeterAdditionalSiteInformation (string ≤ 100 chars.) (§5.1.71 on page 57)

XPath to aseXML node	aseXML node restrictions	aseXML type
AssetManagementPlan	nillable="true" minOccurs="0"	*Type ase:MeterAssetManagementPlan (string ≤ 50 chars.)* (§5.1.72 on page 57)
CalibrationTables	nillable="true" minOccurs="0"	*Type ase:MeterCalibrationTables (string ≤ 50 chars.)* (§5.1.73 on page 57)
CommunicationsEquipmentType	nillable="true" minOccurs="0"	*Type ase:MeterCommunicationsEquipmentType (string ≤ 4 chars.)* (§5.1.74 on page 57)
CommunicationsProtocol	nillable="true" minOccurs="0"	*Type ase:MeterCommunicationsProtocol (string ≤ 50 chars.)* (§5.1.75 on page 58)
Constant	nillable="true" minOccurs="0"	*Type ase:MeterCommunicationsEquipmentType (string ≤ 4 chars.)* (§5.1.74 on page 57)
CustomerFundedMeter	nillable="true" minOccurs="0"	*Type ase:CustomerFundedMeter (boolean)* (§5.1.33 on page 37)
DataConversion	minOccurs="0"	*Type ase:MeterDataConversion (string ≤ 50 chars.)* (§5.1.78 on page 58)
DataValidations	nillable="true" minOccurs="0"	*Type ase:MeterDataValidations (string ≤ 50 chars.)* (§5.1.79 on page 58)
DisplayType	nillable="true" minOccurs="0"	*Type ase:DisplayType (string 1–20 chars.)* (§5.1.36 on page 38)
EstimationInstructions	nillable="true" minOccurs="0"	*Type ase:MeterEstimationInstructions (string ≤ 50 chars.)* (§5.1.82 on page 59)
Hazard	nillable="true" minOccurs="0"	*Type ase:MeterHazard (string ≤ 12 chars.)* (§5.1.83 on page 59)
InstallationTypeCode	nillable="true" minOccurs="0"	*Type ase:MeterInstallationTypeCode (string ≤ 8 chars.)* (§5.1.84 on page 59)
KeyCode	nillable="true" minOccurs="0"	*Type ase:KeyCode (string 1–8 chars.)* (§5.1.66 on page 55)
LastTestDate	nillable="true" minOccurs="0"	xsd:date
Location	nillable="true" minOccurs="0"	*Type ase:MeterLocation (string ≤ 50 chars.)* (§5.1.86 on page 60)
Manufacturer	nillable="true" minOccurs="0"	*Type ase:MeterManufacturer (string ≤ 15 chars.)* (§5.1.87 on page 60)
MeasurementType	nillable="true" minOccurs="0"	*Type ase:MeterMeasurementType (string ≤ 4 chars.)* (§5.1.88 on page 60)
Model	nillable="true" minOccurs="0"	*Type ase:MeterModel (string ≤ 12 chars.)* (§5.1.89 on page 60)
NextScheduledReadDate	nillable="true" minOccurs="0"	xsd:date
NextTestDate	nillable="true" minOccurs="0"	xsd:date
Password	nillable="true" minOccurs="0"	Type ase:MeterPassword (string ≤ 20 chars.)5.1.92on page 61
Point	nillable="true" minOccurs="0"	*Type ase:MeterPoint (string ≤ 2 chars.)* (§5.1.93 on page 61)
Program	nillable="true" minOccurs="0"	*Type ase:MeterProgram (string ≤ 30 chars.)* (§5.1.94 on page 61)
ReadTypeCode	nillable="true" minOccurs="0"	*Type ase:MeterReadTypeCode (string ≤ 4 chars.)* (§5.1.95 on page 61)
RegisterConfiguration	nillable="true" minOccurs="0"	*Type ase:ElectricityMeterRegisterConfiguration (complex)* (§5.1.44 on page 44)
RemotePhoneNumber	nillable="true" minOccurs="0"	*Type ase:MeterRemotePhoneNumber (string ≤ 12 chars.)* (§5.1.99 on page 62)
Route	nillable="true" minOccurs="0"	*Type ase:MeterRoute (string ≤ 12 chars.)* (§5.1.100 on page 62)
SerialNumber	nillable="true" minOccurs="0"	*Type ase:MeterSerialNumber (string ≤ 12 chars.)* (§5.1.101 on page 63)

XPath to aseXML node	aseXML node restrictions	aseXML type

Status	nillable="true"
	minOccurs="0"
		Type ase:MeterStatusCode (enumerated list) (§5.1.102 on page 63)

TestCalibrationProgram	nillable="true"
	minOccurs="0"
		Type ase:MeterTestCalibrationProgram (string ≤ 50 chars.) (§5.1.103 on page 63)

TestPerformedBy	nillable="true"
	minOccurs="0"
		Type ase:MeterTestPerformedBy (string ≤ 20 chars.) (§5.1.104 on page 63)

TestResultAccuracy	nillable="true"
	minOccurs="0"
		Type ase:MeterTestResultAccuracy (decimal, format 999.99999) (§5.1.105 on page 63)

TestResultNotes	nillable="true"
	minOccurs="0"
		Type ase:MeterTestResultNotes (string ≤ 50 chars.) (§5.1.106 on page 64)

TransformerLocation	nillable="true"
	minOccurs="0"
		Type ase:MeterTransformerLocation (string ≤ 30 chars.) (§5.1.108 on page 64)

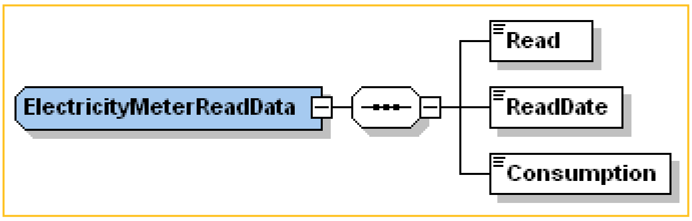
TransformerRatio	nillable="true"
	minOccurs="0"
		Type ase:MeterTransformerRatio (string ≤ 20 chars.) (§5.1.109 on page 64)

TransformerType	nillable="true"
	minOccurs="0"
		Type ase:MeterTransformerType (string ≤ 20 chars.) (§5.1.110 on page 64)

Use	nillable="true"
	minOccurs="0"
		5.1.112*Type ase:MeterUse (string ≤ 10 chars.)* (§5.1.112 on page 65)

UserAccessRights	nillable="true"
	minOccurs="0"
		Type ase:MeterUserAccessRights (string ≤ 50 chars.) (§5.1.113 on page 65)

5.1.43 Type ase:ElectricityMeterReadData (complex)



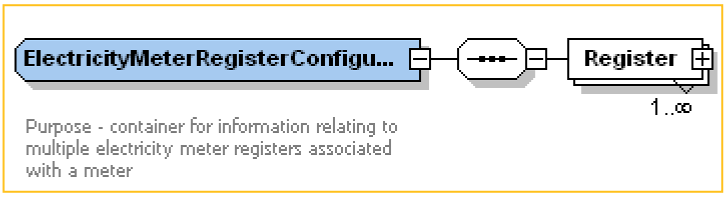
AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ElectricityMeterReadData are:

XPath to aseXML node	aseXML node restrictions	aseXML type
Consumption	totalDigits value="15" fractionDigits value="3"	xsd:decimal
Read	maxLength value="15"	xsd:string
ReadDate		xsd:date

5.1.44 Type ase:ElectricityMeterRegisterConfiguration (complex)

The aseXML documentation for type ase:ElectricityMeterRegisterConfiguration is:

Purpose - container for information relating to multiple electricity meter registers associated with a meter


ElectricityMeterRegisterConfigu...
Register
1..∞
Purpose - container for information relating to multiple electricity meter registers associated with a meter

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ElectricityMeterRegisterConfiguration are:

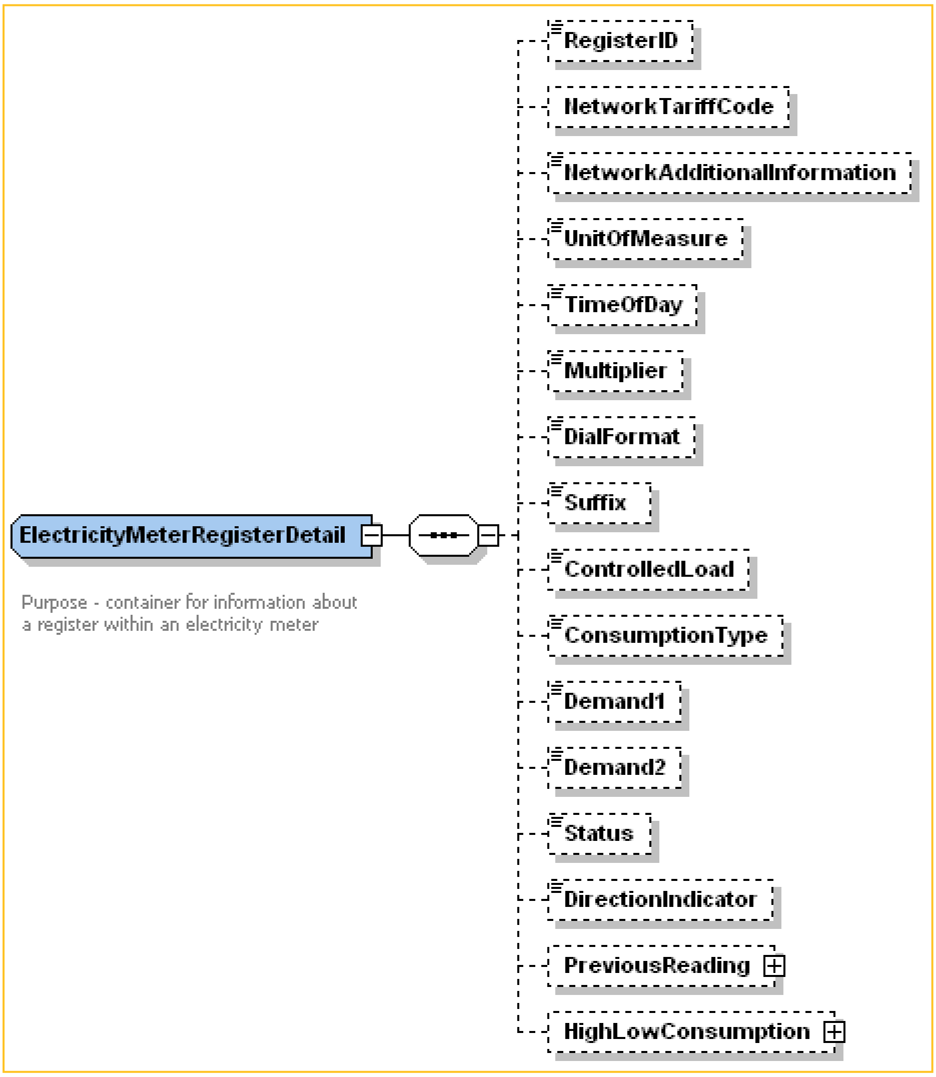
XPath to aseXML node	aseXML node restrictions	aseXML type

XPath to aseXML node	aseXML node restrictions	aseXML type
Register	maxOccurs="unbounded"	*Type ase:ElectricityMeterRegisterDetail (complex)* (§5.1.45 below)

5.1.45 Type ase:ElectricityMeterRegisterDetail (complex)

The aseXML documentation for type ase:ElectricityMeterRegisterDetail is:

Purpose - container for information about a register within an electricity meter



AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ElectricityMeterRegisterDetail are:

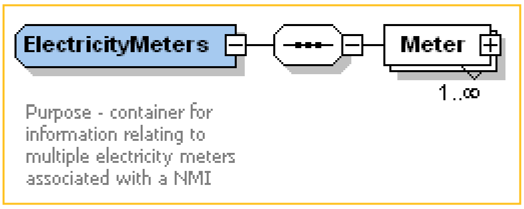
XPath to aseXML node	aseXML node restrictions	aseXML type
ConsumptionType	nillable="true" minOccurs="0"	*Type ase:MeterConsumptionType (enumerated)* (§5.1.76 on page 58)
ControlledLoad	nillable="true" minOccurs="0"	*Type ase:MeterControlledLoad (string ≤ 100 chars.)* (§5.1.77 on page 58)
Demand1	nillable="true" minOccurs="0"	*Type ase:MeterDemand (integer, 8 digits)* (§5.1.80 on page 59)
Demand2	nillable="true" minOccurs="0"	*Type ase:MeterDemand (integer, 8 digits)* (§5.1.80 on page 59)
DialFormat	nillable="true" minOccurs="0"	*Type ase:MeterDialFormat (string ≤ 50 chars.)* (§5.1.81 on page 59)

XPath to aseXML node	aseXML node restrictions	aseXML type
DirectionIndicator	nillable="true" minOccurs="0"	*Type ase:DirectionIndicator (enumerated list)* (§5.1.35 on page 38)
HighLowConsumption	nillable="true" minOccurs="0"	*Type ase:HighLowConsumption (complex)* (§5.1.64 on page 54)
Multiplier	nillable="true" minOccurs="0"	*Type ase:MeterMultiplier (decimal)* (§5.1.90 on page 60)
NetworkAdditionalInformation	nillable="true" minOccurs="0"	*Type ase:MeterNetworkAdditionalInformation (string)* (§5.1.91 on page 61)
NetworkTariffCode	nillable="true" minOccurs="0"	*Type ase:NetworkTariffCode (string ≤ 10 chars.)* (§5.1.114 on page 65)
PreviousReading	nillable="true" minOccurs="0"	*Type ase:ElectricityMeterReadData (complex)* (§5.1.43 on page 44)
RegisterID	minOccurs="0"	*Type ase:MeterRegisterIdentifier (string ≤ 10 chars.)* (§5.1.96 on page 61)
Status	nillable="true" minOccurs="0"	*Type ase:MeterRegisterStatusCode (enumerated)* (§5.1.97 on page 62)
Suffix	nillable="true" minOccurs="0"	*Type ase:NMIDataStreamSuffix (string = 2 chars.)* (§5.1.118 on page 66)
TimeOfDay	nillable="true" minOccurs="0"	*Type ase:MeterTimeOfDay (string ≤ 10 chars.)* (§5.1.107 on page 64)
UnitOfMeasure	nillable="true" minOccurs="0"	*Type ase:MeterUnitOfMeasure (string ≤ 5 chars.)* (§5.1.111 on page 65)

5.1.46 Type ase:ElectricityMeters (complex)

The aseXML documentation for type ase:ElectricityMeters is:

Purpose - container for information relating to multiple electricity meters associated with a NMI


ElectricityMeters
Meter
1..∞
Purpose - container for information relating to multiple electricity meters associated with a NMI

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ElectricityMeters are:

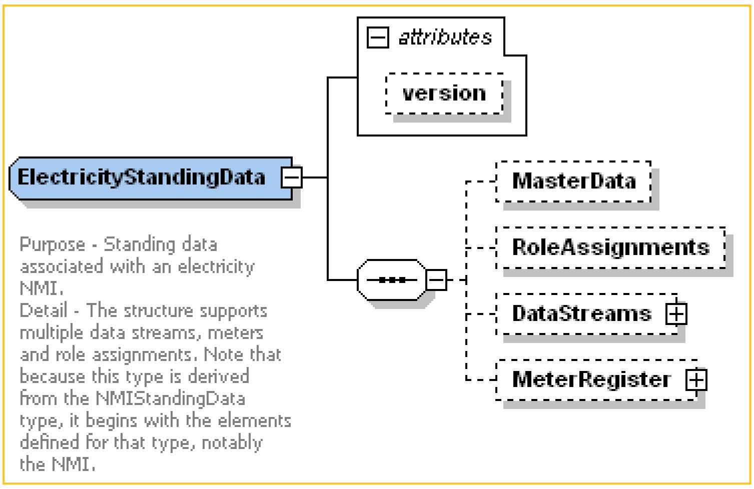
XPath to aseXML node	aseXML node restrictions	aseXML type
Meter	maxOccurs="unbounded"	*Type ase:ElectricityMeter (complex)* (§5.1.42 on page 41)

5.1.47 Type ase:ElectricityStandingData (complex)

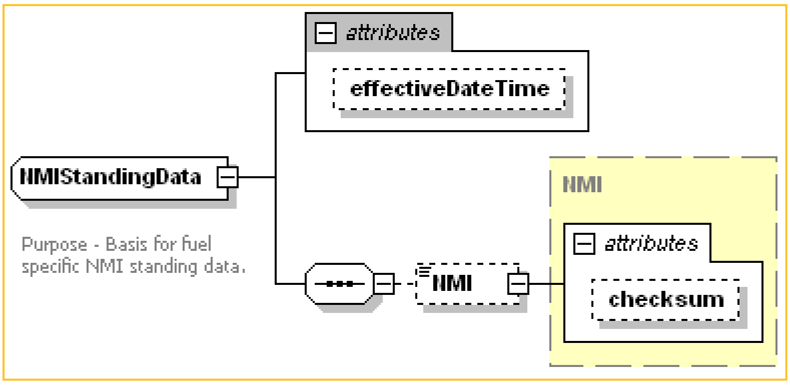
The aseXML documentation for type ase:ElectricityStandingData is:

Purpose - Standing data associated with an electricity NMI.

Detail - The structure supports multiple data streams, meters and role assignments. Note that because this type is derived from the NMIStandingData type, it begins with the elements defined for that type, notably the NMI.


attributes
version
ElectricityStandingData
MasterData
RoleAssignments
DataStreams
MeterRegister
Purpose - Standing data associated with an electricity NMI.
Detail - The structure supports multiple data streams, meters and role assignments. Note that because this type is derived from the NMIStandingData type, it begins with the elements defined for that type, notably the NMI.

Type ase:ElectricityStandingData extends *(abstract) Type ase:NMIStandingData* (complex) (§5.1.119 on page 66).



AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ElectricityStandingData are:

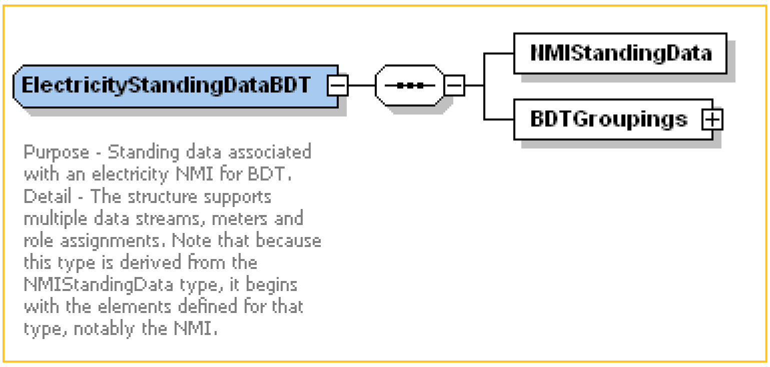
XPath to aseXML node	aseXML node restrictions	aseXML type
@effectiveDateTime	use="optional"	xsd:dateTime
@version	use="optional"	r25—*Type ase:ReleaseIdentifier (string with pattern)* (§5.1.125 on page 68)
DataStreams	minOccurs="0"	*Type ase:ElectricityDataStreams (complex)*, (§5.1.40 on page 39)
MasterData	minOccurs="0"	*Type ase:ElectricityMasterStandingData (complex)* (§5.1.41 on page 39)
MeterRegister	minOccurs="0"	*Type ase:ElectricityMeters (complex)* (§5.1.46 on page 46)
NMI	minOccurs="0"	*Type ase:NMI (complex)* (§5.1.98 on page 62)
RoleAssignments	minOccurs="0"	*Type ase:RoleAssignments (complex)* (§5.1.129 on page 70)

5.1.48 Type ase:ElectricityStandingDataBDT (complex)

The aseXML documentation for type ase:ElectricityStandingDataBDT is:

Purpose - Standing data associated with an electricity NMI for BDT.

Detail - The structure supports multiple data streams, meters and role assignments. Note that because this type is derived from the NMIStandingData type, it begins with the elements defined for that type, notably the NMI.


ElectricityStandingDataBDT
NMIStandingData
BDTGroupings
Purpose - Standing data associated with an electricity NMI for BDT. Detail - The structure supports multiple data streams, meters and role assignments. Note that because this type is derived from the NMIStandingData type, it begins with the elements defined for that type, notably the NMI.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ElectricityStandingDataBDT:

XPath to aseXML node	aseXML node restrictions	aseXML type
NMIStandingData		*(abstract) Type ase:NMIStandingData (complex)*, (§5.1.119 on page 66)
BDTGroupings		BDTGroupings

5.1.49 Type ase:EmbeddedNetworkIdentifier (string ≤ 10 chars.)

The aseXML documentation for type ase:EmbeddedNetworkIdentifier is:

Purpose - Identify an embedded network

MSATS Data Model Column - EmbNetIdCode, EmbNetParent, EmbNetChild

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:EmbeddedNetworkIdentifier	maxLength value="10"	xsd:string

5.1.50 Type ase:EMSD_CustomerClassificationCode (string 1-20 chars.)

The aseXML documentation for type ase:EMSD_CustomerClassificationCode is:

Classification of customer type, value assigned by FRMP.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:EMSD_CustomerClassificationCode	maxLength value="20" minLength value="1" whiteSpace value="collapse"	xsd:string

5.1.51 Type ase:EMSD_CustomerThresholdCode (string 1-20 chars.)

The aseXML documentation for type ase:EMSD_CustomerThresholdCode is:

Classification of customer consumption threshold, value assigned by DNSP.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:EMSD_CustomerThresholdCode	maxLength value="20" minLength value="1" whiteSpace value="collapse"	xsd:string

5.1.52 Type ase:EnergyMarket (enumerated list)

The aseXML documentation for type ase:EnergyMarket is:

Purpose -Indicate the energy market to which the message belongs

Detail - Markets are currently defined for the National Electricity Market, the NSW Gas Market and the Victorian Gas Market.

Type ase:EnergyMarket has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- AATELEC.
- ACTELEC.
- NEM.
- NSWELEC.
- NTELEC.
- QLDELEC.
- SAELEC.
- TASELEC.
- VICELEC.
- WAELEC.
- AATGAS.
- ACTGAS.
- NSWGAS.
- NTGAS.
- QLDGAS.
- SAGAS.
- TASGAS.
- VICGAS.
- WAGAS.

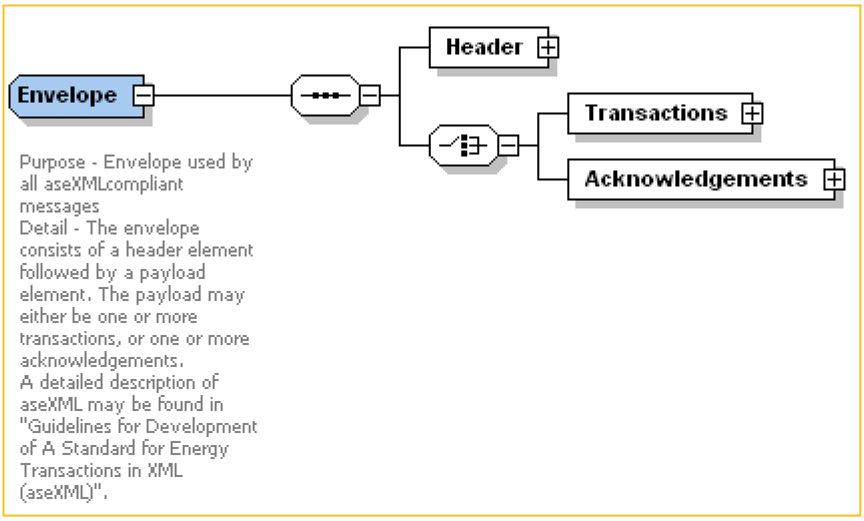
5.1.53 Type ase:Envelope (complex)

The aseXML documentation for type ase:Envelope is:

Purpose - Envelope used by all aseXMLcompliant messages

Detail - The envelope consists of a header element followed by a payload element. The payload may either be one or more transactions, or one or more acknowledgements.

A detailed description of aseXML may be found in "Guidelines for Development of A Standard for Energy Transactions in XML (aseXML)".


Envelope
Header
Transactions
Acknowledgements
Purpose - Envelope used by all aseXMLcompliant messages
Detail - The envelope consists of a header element followed by a payload element. The payload may either be one or more transactions, or one or more acknowledgements.
A detailed description of aseXML may be found in "Guidelines for Development of A Standard for Energy Transactions in XML (aseXML)".

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:Envelope are:

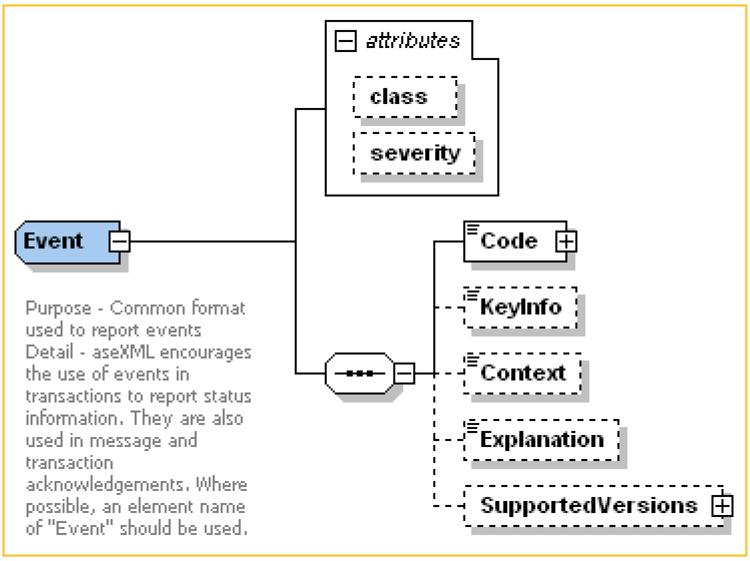
XPath to aseXML node	aseXML node restrictions	aseXML type
Acknowledgements		*Type ase:Acknowledgements (complex)* (§5.1.2 on page 17)
Header		*Type ase:Header (complex)* (§5.1.63 on page 53)
Transactions		*Type ase:Transactions (complex)* (§5.1.136 on page 76)

5.1.54 Type ase:Event (complex)

The aseXML documentation for type ase:Event is:

Purpose - Common format used to report events

Detail - aseXML encourages the use of events in transactions to report status information. They are also used in message and transaction acknowledgements. Where possible, an element name of "Event" should be used.


attributes
class
severity
Event
Code
KeyInfo
Context
Explanation
SupportedVersions
Purpose - Common format used to report events
Detail - aseXML encourages the use of events in transactions to report status information. They are also used in message and transaction acknowledgements. Where possible, an element name of "Event" should be used.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:Event are:

XPath to aseXML node	aseXML node restrictions	aseXML type

@class	use="optional"
	default="Application"
		Type ase:EventClass (enumerated) (§5.1.55 below)

@severity	use="optional"
	default="Fatal"
		Type ase:EventSeverity (enumerated) (§5.1.60 on page 52)

Code		*Type ase:EventCode (complex)* (§5.1.56 below)
Context	minOccurs="0"	*Type ase:EventContext (string ≤ 80 chars.)* (§5.1.58 on page 52)
Explanation	minOccurs="0"	xsd:string
KeyInfo	minOccurs="0"	*Type ase:EventKeyInfo (string ≤ 80 chars.)* (§5.1.59 on page 52)
SupportedVersions	minOccurs="0"	*Type ase:EventSupportedVersions (complex)* (§5.1.61 on page 53)

5.1.55 Type ase:EventClass (enumerated)

The aseXML documentation for type ase:EventClass is:

Purpose - Provide a high level categorisation of events

Detail - Message events pertain to the structure and delivery of aseXML messages.

Application events are generated by the applications communicating via aseXML.

Processing events reflect environmental issues associated with running applications.

Type ase:EventClass has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

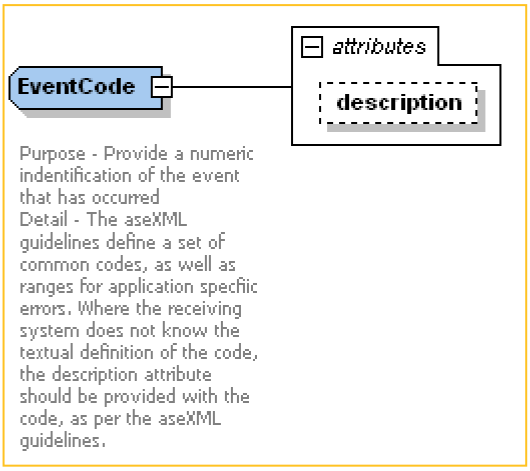
- Message.
- Application.
- Processing.

5.1.56 Type ase:EventCode (complex)

The aseXML documentation for type ase:EventCode is:

Purpose - Provide a numeric indentification of the event that has occurred

Detail - The aseXML guidelines define a set of common codes, as well as ranges for application specfiic errors. Where the receiving system does not know the textual definition of the code, the description attribute should be provided with the code, as per the aseXML guidelines.


EventCode
attributes
description
Purpose - Provide a numeric indentification of the event that has occurred
Detail - The aseXML guidelines define a set of common codes, as well as ranges for application specfiic errors. Where the receiving system does not know the textual definition of the code, the description attribute should be provided with the code, as per the aseXML guidelines.

Type ase:EventCode extends base="EventCodeBase". AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:EventCode are:

XPath to aseXML node	aseXML node restrictions	aseXML type
@description	use="optional"	xsd:string
element using the type ase:EventCode		*Type ase:EventCodeBase (nonNegativeInteger)* (§5.1.57 below)

5.1.57 Type ase:EventCodeBase (nonNegativeInteger)

The aseXML documentation for type ase:EventCodeBase is:

Purpose - Helper type in defining EventCode

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:EventCodeBase		xsd:nonNegativeInteger

5.1.58 Type ase:EventContext (string ≤ 80 chars.)

The aseXML documentation for type ase:EventContext is:

Purpose - Identify the specific data/condition that caused the event

Detail - The portion of the input to which the event applies should be provided.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:EventContext	maxLength value="80"	xsd:string

5.1.59 Type ase:EventKeyInfo (string ≤ 80 chars.)

The aseXML documentation for type ase:EventKeyInfo is:

Purpose - Provide information to allow identification of the data that triggered the event

Detail - Where the combination of class and code are insufficient to completely describe an event, KeyInfo may be used to provide further detail as to the information needed to locate the source of the event within the original transaction.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:EventKeyInfo	maxLength value="80"	xsd:string

5.1.60 Type ase:EventSeverity (enumerated)

The aseXML documentation for type ase:EventSeverity is:

Purpose - Provide an indication as to the action required as a result of receiving an event

Detail - The severity dictates the further processing possible after receiving the event.

Type ase:EventSeverity has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- Information.
- Warning.
- Error.

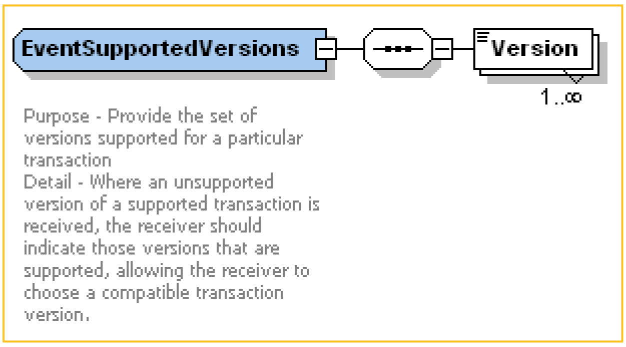
- Fatal.

5.1.61 Type ase:EventSupportedVersions (complex)

The aseXML documentation for type ase:EventSupportedVersions is:

Purpose - Provide the set of versions supported for a particular transaction

Detail - Where an unsupported version of a supported transaction is received, the receiver should indicate those versions that are supported, allowing the receiver to choose a compatible transaction version.


EventSupportedVersions
Version
1..∞
Purpose - Provide the set of versions supported for a particular transaction
Detail - Where an unsupported version of a supported transaction is received, the receiver should indicate those versions that are supported, allowing the receiver to choose a compatible transaction version.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:EventSupportedVersions are:

XPath to aseXML node	aseXML node restrictions	aseXML type
Version	maxOccurs="unbounded"	*Type ase:ReleaseIdentifier (string with pattern)* (§5.1.125 on page 68)

5.1.62 Type ase:FeederClass (string 1-15 chars.)

The aseXML documentation for type ase:FeederClass is:

Classification of High Voltage distribution line feeding property.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:FeederClass	maxLength value="15" minLength value="1" whiteSpace value="collapse"	xsd:string

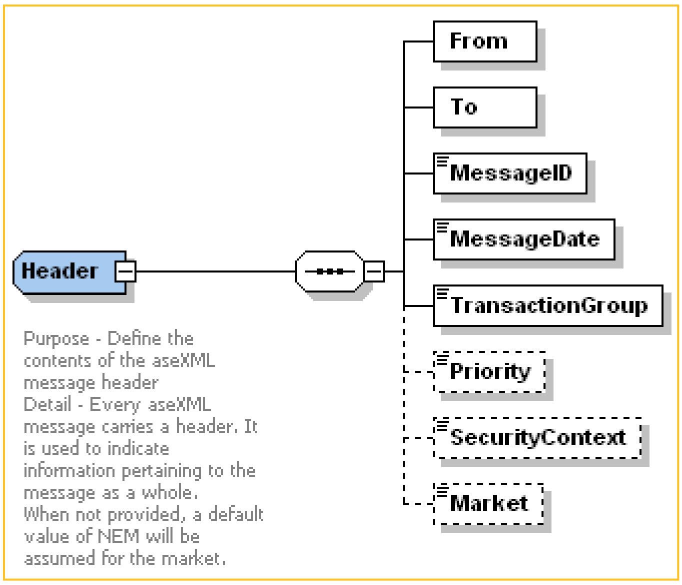
5.1.63 Type ase:Header (complex)

The aseXML documentation for type ase:Header is:

Purpose - Define the contents of the aseXML message header

Detail - Every aseXML message carries a header. It is used to indicate information pertaining to the message as a whole.

When not provided, a default value of NEM will be assumed for the market.


Header
From
To
MessageID
MessageDate
TransactionGroup
Priority
SecurityContext
Market
Purpose - Define the contents of the aseXML message header
Detail - Every aseXML message carries a header. It is used to indicate information pertaining to the message as a whole.
When not provided, a default value of NEM will be assumed for the market.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:Header are:

XPath to aseXML node	aseXML node restrictions	aseXML type
From		*Type ase:PartyIdentifier (complex)* (§5.1.121 on page 67)

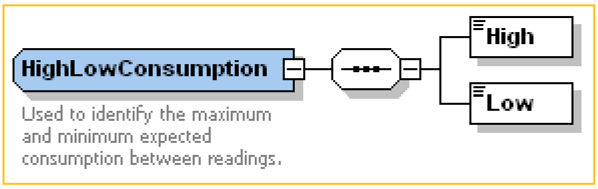
Market	minOccurs="0"
	default="NEM"
		Type ase:EnergyMarket (enumerated list) (§5.1.52 on page 48)

MessageDate		xsd:dateTime
MessageID		*Type ase:MessageIdentifier* (string, 1-36 chars.) (§5.1.68 on page 56)
Priority	minOccurs="0"	*Type ase:TransactionPriority (enumerated)* (§5.1.135 on page 76)
SecurityContext	minOccurs="0"	*Type ase:MessageSecurityContext (string, ≤ 15 chars.)* (§5.1.69 on page 56)
To		*Type ase:PartyIdentifier (complex)* (§5.1.121 on page 67)
TransactionGroup		*Type ase:TransactionGroup (enumerated)* (§5.1.133 on page 74)

5.1.64 Type ase:HighLowConsumption (complex)

The aseXML documentation for type ase:HighLowConsumption is:

Used to identify the maximum and minimum expected consumption between readings.


HighLowConsumption
High
Low
Used to identify the maximum and minimum expected consumption between readings.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:HighLowConsumption are:

XPath to aseXML node	aseXML node restrictions	aseXML type

XPath to aseXML node	aseXML node restrictions	aseXML type
High		xsd:integer
Low		xsd:integer

5.1.65 Type ase:JurisdictionCode (string ≤ 3 chars.)

The aseXML documentation for type ase:JurisdictionCode is:

Purpose - Identify the jurisdictions of relevance to aseXML

MSATS Data Model Column - JurisdictionCode

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:JurisdictionCode	maxLength value="3"	xsd:string

5.1.66 Type ase:KeyCode (string 1–8 chars.)

The aseXML documentation for type ase:KeyCode is:

Identifies the type of key and/or location of the key required to access a meter at a site. This is required in standing data as the Franchise market Retailer still carries responsibility for performing site activities for some areas of the network.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:KeyCode	maxLength value="8" minLength value="1" whiteSpace value="collapse"	xsd:string

5.1.67 Type ase:MessageAcknowledgement (complex)

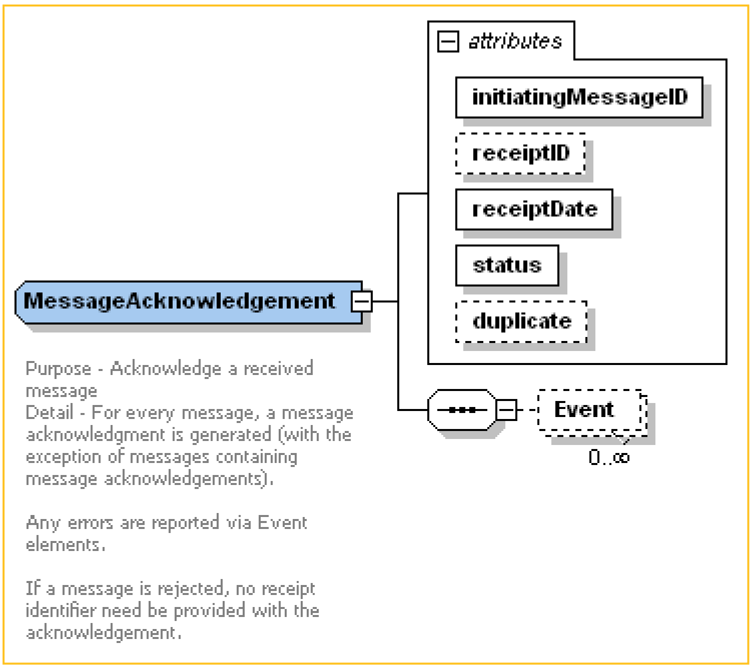
The aseXML documentation for type ase:MessageAcknowledgement is:

Purpose - Acknowledge a received message

Detail - For every message, a message acknowledgment is generated (with the exception of messages containing message acknowledgements).

Any errors are reported via Event elements.

If a message is rejected, no receipt identifier need be provided with the acknowledgement.


attributes
initiatingMessageID
receiptID
receiptDate
status
duplicate
MessageAcknowledgement
Event
0..∞
Purpose - Acknowledge a received message
Detail - For every message, a message acknowledgment is generated (with the exception of messages containing message acknowledgements).

Any errors are reported via Event elements.

If a message is rejected, no receipt identifier need be provided with the acknowledgement.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:MessageAcknowledgement are:

XPath to aseXML node	aseXML node restrictions	aseXML type
@initiatingMessageID	use="required"	*Type ase:MessageIdentifier* (string, 1-36 chars.) (§5.1.68 below)
@receiptDate	use="required"	xsd:dateTime
@receiptID	use="optional"	*Type ase:ReceiptIdentifier (string, 1-36 chars.)* (§5.1.124 on page 68)
@status	use="required"	*Type ase:MessageStatus (enumerated)* (§5.1.70 on page 57)
duplicate	default="No"	*Type ase:YesNo (enumerated list)* (§5.1.141 on page 77)

Event	minOccurs="0"
	maxOccurs="unbounded"
		Type ase:Event (complex) (§5.1.54 on page 50)

5.1.68 Type ase:MessageIdentifier (string, 1-36 chars.)

The aseXML documentation for type ase:MessageIdentifier is:

Purpose - Uniquely identify every message generated by the message sender

Detail - Note that message identifiers do not have to be globally unique, only unique to a particular sender. However, the length has been chosen such that UUIDs can be used as message identifiers if considered appropriate, guaranteeing global uniqueness.

A message acknowledgement identifies which message it is acknowledging by providing the message identifier as an attribute.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MessageIdentifier	minLength value="1", maxLength value="36"	xsd:string

5.1.69 Type ase:MessageSecurityContext (string, ≤ 15 chars.)

The aseXML documentation for type ase:MessageSecurityContext is:

Purpose - Provide the processing priority desired by the sender

Detail - The sender can indicate information needed by the receiver to determine whether or not the sender is authorised to submit the transactions within the message.

For the Market Settlement And Transfer System (MSATS), this will be used to hold the participant userid from which the context for transaction processing is determined.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MessageSecurityContext	maxLength value="15"	xsd:string

5.1.70 Type ase:MessageStatus (enumerated)

The aseXML documentation for type ase:MessageStatus is:

Purpose - Indicate the acceptance or rejection of the message.

Type ase:MessageStatus has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- Accept.
- Reject.

5.1.71 Type ase:MeterAdditionalSiteInformation (string ≤ 100 chars.)

The aseXML documentation for type ase:MeterAdditionalSiteInformation is:

MSATS Data Model Column - AddlSiteInfo

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterAdditionalSiteInformation	maxLength value="100"	xsd:string

5.1.72 Type ase:MeterAssetManagementPlan (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterAssetManagementPlan is:

MSATS Data Model Column - AssetMgmtPlan

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterAssetManagementPlan	maxLength value="50"	xsd:string

5.1.73 Type ase:MeterCalibrationTables (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterCalibrationTables is:

MSATS Data Model Column - CalibrationTables

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterCalibrationTables	maxLength value="50"	xsd:string

5.1.74 Type ase:MeterCommunicationsEquipmentType (string ≤ 4 chars.)

The aseXML documentation for type ase:MeterCommunicationsEquipmentType is:

MSATS Data Model Column - CommEquipType

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterCommunicationsEquipmentType	maxLength value="4"	xsd:string

5.1.75 Type ase:MeterCommunicationsProtocol (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterCommunicationsProtocol is:

MSATS Data Model Column - CommProtocol

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterCommunicationsProtocol	maxLength value="50"	xsd:string

5.1.76 Type ase:MeterConsumptionType (enumerated)

The aseXML documentation for type ase:MeterConsumptionType is:

MSATS Data Model Column -

Type ase:MeterStatusCode has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- Actual.
- Cumulative.

5.1.77 Type ase:MeterControlledLoad (string ≤ 100 chars.)

The aseXML documentation for type ase:MeterControlledLoad is:

MSATS Data Model Column - ControlledLoad

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterControlledLoad	length value="100"	xsd:string

5.1.78 Type ase:MeterDataConversion (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterDataConversion is:

MSATS Data Model Column - DataConv

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterDataConversion	maxLength value="50"	xsd:string

5.1.79 Type ase:MeterDataValidations (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterDataValidations is:

MSATS Data Model Column - DataValidations

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterDataValidations	maxLength value="50"	xsd:string

5.1.80 Type ase:MeterDemand (integer, 8 digits)

The aseXML documentation for type ase:MeterDemand is:

MSATS Data Model Column - Demand1, Demand2

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterDemand	totalDigits value="8"	xsd:integer

5.1.81 Type ase:MeterDialFormat (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterDialFormat is:

MSATS Data Model Column - DialFormat

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterDialFormat	totalDigits value="4" ractionDigits value="2" minInclusive value="0" maxInclusive value="99.99"	xsd:decimal

5.1.82 Type ase:MeterEstimationInstructions (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterEstimationInstructions is:

MSATS Data Model Column - EstInstruct

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterEstimationInstructions	maxLength value="50"	xsd:string

5.1.83 Type ase:MeterHazard (string ≤ 12 chars.)

The aseXML documentation for type ase:MeterHazard is:

MSATS Data Model Column - MeterHazard

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterHazard	maxLength value="12"	xsd:string

5.1.84 Type ase:MeterInstallationTypeCode (string ≤ 8 chars.)

The aseXML documentation for type ase:MeterInstallationTypeCode is:

MSATS Data Model Column - MeterInstallCode

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterInstallationTypeCode	maxLength value="8"	xsd:string

See also *Type ase:MeterInstallationTypeDescription (string ≤ 50 chars.)* (§5.1.85 below).

5.1.85 Type ase:MeterInstallationTypeDescription (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterInstallationTypeDescription is:

Purpose - Describe installation type codes

MSATS Data Model Column - MeterInstallDesc

Detail - The description is intended to assist in identification of the correct code. The mapping between codes and descriptions will be provided to participants via the Table Replication transaction exchanges.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterInstallationTypeDescription	maxLength value="50"	xsd:string

5.1.86 Type ase:MeterLocation (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterLocation is:

MSATS Data Model Column - MeterLocation

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterLocation	maxLength value="50"	xsd:string

5.1.87 Type ase:MeterManufacturer (string ≤ 15 chars.)

The aseXML documentation for type ase:MeterManufacturer is:

MSATS Data Model Column - MeterManufacturer

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterManufacturer	maxLength value="15"	xsd:string

5.1.88 Type ase:MeterMeasurementType (string ≤ 4 chars.)

The aseXML documentation for type ase:MeterMeasurementType is:

MSATS Data Model Column - MeasurementType

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterMeasurementType	maxLength value="4"	xsd:string

5.1.89 Type ase:MeterModel (string ≤ 12 chars.)

The aseXML documentation for type ase:MeterModel is:

MSATS Data Model Column - MeterModel

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterModel	maxLength value="12"	xsd:string

5.1.90 Type ase:MeterMultiplier (decimal)

The aseXML documentation for type ase:MeterMultiplier is:

MSATS Data Model Column - Multiplier

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterMultiplier		xsd:decimal

5.1.91 Type ase:MeterNetworkAdditionalInformation (string)

The aseXML documentation for type ase:MeterNetworkAdditionalInformation is:

MSATS Data Model Column - NTAddInfo

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterNetworkAdditionalInformation		xsd:string

5.1.92 Type ase:MeterPassword (string ≤ 20 chars.)

The aseXML documentation for type ase:MeterPassword is:

MSATS Data Model Column - Password

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterPassword	maxLength value="20"	xsd:string

5.1.93 Type ase:MeterPoint (string ≤ 2 chars.)

The aseXML documentation for type ase:MeterPoint is:

MSATS Data Model Column - MeterPoint

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterPoint	maxLength value="2"	xsd:string

5.1.94 Type ase:MeterProgram (string ≤ 30 chars.)

The aseXML documentation for type ase:MeterProgram is:

MSATS Data Model Column - MeterProgram

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterProgram	maxLength value="30"	xsd:string

5.1.95 Type ase:MeterReadTypeCode (string ≤ 4 chars.)

The aseXML documentation for type ase:MeterReadTypeCode is:

MSATS Data Model Column - MeterReadType

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterReadTypeCode	maxLength value="4"	xsd:string

5.1.96 Type ase:MeterRegisterIdentifier (string ≤ 10 chars.)

The aseXML documentation for type ase:MeterRegisterIdentifier is:

Purpose - Identifies a register within a meter

MSATS Data Model Column - RegisterId

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterRegisterIdentifier	maxLength value="10"	xsd:string

5.1.97 Type ase:MeterRegisterStatusCode (enumerated)

The aseXML documentation for type ase:MeterRegisterStatusCode is:

MSATS Data Model Column - RegisterIdStatus

Type ase:MeterRegisterStatusCode has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

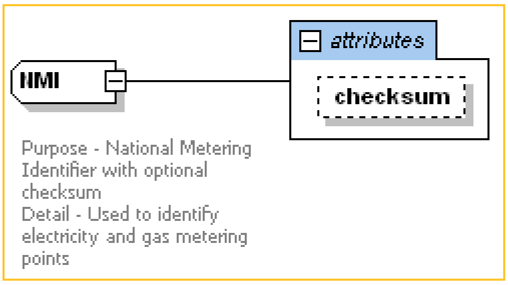
- C.
- R.

5.1.98 Type ase:NMI (complex)

The aseXML documentation for type ase:NMI is:

Purpose - National Metering Identifier with optional checksum

Detail - Used to identify electricity and gas metering points


NMI
attributes
checksum
Purpose - National Metering Identifier with optional checksum
Detail - Used to identify electricity and gas metering points

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:NMI:

XPath to aseXML node	aseXML node restrictions	aseXML type
@checksum	use="optional"	*Type ase:NMIChecksum (integer, 0–9)* (§5.1.116 on page 65)
element using the type ase:NMI		*Type ase:NMIBase (string ≤ 10 chars.)*, (§5.1.115 on page 65)

5.1.99 Type ase:MeterRemotePhoneNumber (string ≤ 12 chars.)

The aseXML documentation for type ase:MeterRemotePhoneNumber is:

MSATS Data Model Column - RemotePhoneNumber

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterRemotePhoneNumber	maxLength value="12"	xsd:string

5.1.100 Type ase:MeterRoute (string ≤ 12 chars.)

The aseXML documentation for type ase:MeterRoute is:

MSATS Data Model Column - MeterRoute

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterRoute	maxLength value="12"	xsd:string

5.1.101 Type ase:MeterSerialNumber (string ≤ 12 chars.)

The aseXML documentation for type ase:MeterSerialNumber is:

MSATS Data Model Column - MeterSerial

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterSerialNumber	maxLength value="12"	xsd:string

5.1.102 Type ase:MeterStatusCode (enumerated list)

The aseXML documentation for type ase:MeterStatusCode is:

Purpose - Status code of Meter Register Data

MSATS Data Model Column - MeterStatus

Type ase:MeterStatusCode has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- C.
- R.

5.1.103 Type ase:MeterTestCalibrationProgram (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterTestCalibrationProgram is:

MSATS Data Model Column - TestCalibProgrm

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterTestCalibrationProgram	maxLength value="50"	xsd:string

5.1.104 Type ase:MeterTestPerformedBy (string ≤ 20 chars.)

The aseXML documentation for type ase:MeterTestPerformedBy is:

MSATS Data Model Column - TestPerformedBy

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterTestPerformedBy	maxLength value="20"	xsd:string

5.1.105 Type ase:MeterTestResultAccuracy (decimal, format 999.99999)

The aseXML documentation for type ase:MeterTestResultAccuracy is:

MSATS Data Model Column - TestResultAccuracy

XPath to aseXML node	aseXML node restrictions	aseXML type

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterTestResultAccuracy	totalDigits value="8" fractionDigits value="5"	xsd:decimal

5.1.106 Type ase:MeterTestResultNotes (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterTestResultNotes is:

MSATS Data Model Column - TestResultNotes

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterTestResultNotes	maxLength value="50"	xsd:string

5.1.107 Type ase:MeterTimeOfDay (string ≤ 10 chars.)

The aseXML documentation for type ase:MeterTimeOfDay is:

MSATS Data Model Column - TestResultNotes

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterTimeOfDay	maxLength value="10"	xsd:string

5.1.108 Type ase:MeterTransformerLocation (string ≤ 30 chars.)

The aseXML documentation for type ase:MeterTransformerLocation is:

MSATS Data Model Column - TransformerLocation

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterTransformerLocation	maxLength value="30"	xsd:string

5.1.109 Type ase:MeterTransformerRatio (string ≤ 20 chars.)

The aseXML documentation for type ase:MeterTransformerRatio is:

MSATS Data Model Column - TransformerRatio

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterTransformerRatio	maxLength value="20"	xsd:string

5.1.110 Type ase:MeterTransformerType (string ≤ 20 chars.)

The aseXML documentation for type ase:MeterTransformerType is:

MSATS Data Model Column - TransformerType

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterTransformerType	maxLength value="20"	xsd:string

5.1.111 Type ase:MeterUnitOfMeasure (string ≤ 5 chars.)

The aseXML documentation for type ase:MeterUnitOfMeasure is:

MSATS Data Model Column - UnitOfMeasure

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterUnitOfMeasure	maxLength value="5"	xsd:string

5.1.112 Type ase:MeterUse (string ≤ 10 chars.)

The aseXML documentation for type ase:MeterUse is:

MSATS Data Model Column - MeterUse

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterUse	maxLength value="10"	xsd:string

5.1.113 Type ase:MeterUserAccessRights (string ≤ 50 chars.)

The aseXML documentation for type ase:MeterUserAccessRights is:

MSATS Data Model Column - UserAccessRights

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:MeterUserAccessRights	maxLength value="50"	xsd:string

5.1.114 Type ase:NetworkTariffCode (string ≤ 10 chars.)

The aseXML documentation for type ase:NetworkTariffCode is:

Purpose - Identifies an LNSP specific network tariff

MSATS Data Model Column - NetworkTariffCode

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:NetworkTariffCode	maxLength value="10"	xsd:string

5.1.115 Type ase:NMIBase (string ≤ 10 chars.)

The aseXML documentation for type ase:NMIBase is:

Purpose - Helper type to assist in the definition of the NMI type

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:NMIBase	maxLength value="10"	xsd:string

5.1.116 Type ase:NMIChecksum (integer, 0–9)

The aseXML documentation for type ase:NMIChecksum is:

Purpose - Helper type to assist in the definition of the NMI type

Detail - The checksum is a single numeric digit.

XPath to aseXML node	aseXML node restrictions	aseXML type

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:NMIChecksum	maxLength value="10"	xsd:string

5.1.117 Type ase:NMIClassificationCode (string ≤ 8 chars.)

The aseXML documentation for type ase:NMIClassificationCode is:

Purpose - Identify the broad class to which the NMI belongs

MSATS Data Model Column - NMIClassCode

Detail - NMI classification often forms the basis for jurisdictional rules.

MSATS Data Model Column - JurisdictionCode

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:NMIClassificationCode	maxLength value="8"	xsd:string

5.1.118 Type ase:NMIDataStreamSuffix (string = 2 chars.)

The aseXML documentation for type ase:NMIDataStreamSuffix is:

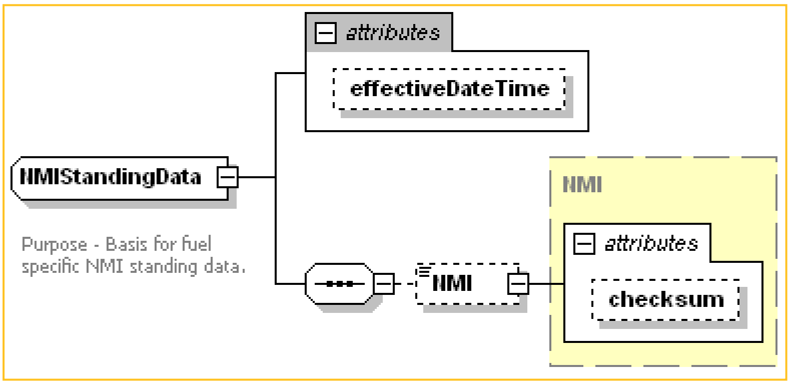
Purpose - Identify a data stream associated with a NMI

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:NMIDataStreamSuffix	length value="2"	xsd:string

5.1.119 (abstract) Type ase:NMIStandingData (complex)

The aseXML documentation for type ase:NMIStandingData is:

Purpose - Basis for fuel specific NMI standing data.



AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:NMIStandingData:

XPath to aseXML node	aseXML node restrictions	aseXML type
@effectiveDateTime	use="optional"	xsd:dateTime
NMI	minOccurs="0"	*Type ase:NMI (complex)*, (§5.1.98 on page 62)

The type ase:NMIStandingData is abstract, so elements using this type have another aseXML type (either within the aseXML schema or explicitly in the aseXML file using an xsi:type attribute).

5.1.120 Type ase:NMIStatusCode (string = 1 char.)

The aseXML documentation for type ase:NMIStatusCode is:

Purpose - Status code of portions of NMI Standing Data

MSATS Data Model Column - NMIStatusCode

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:NMIStatusCode	length value="1"	xsd:string

5.1.121 Type ase:PartyIdentifier (complex)

The aseXML documentation for type ase:PartyIdentifier is:

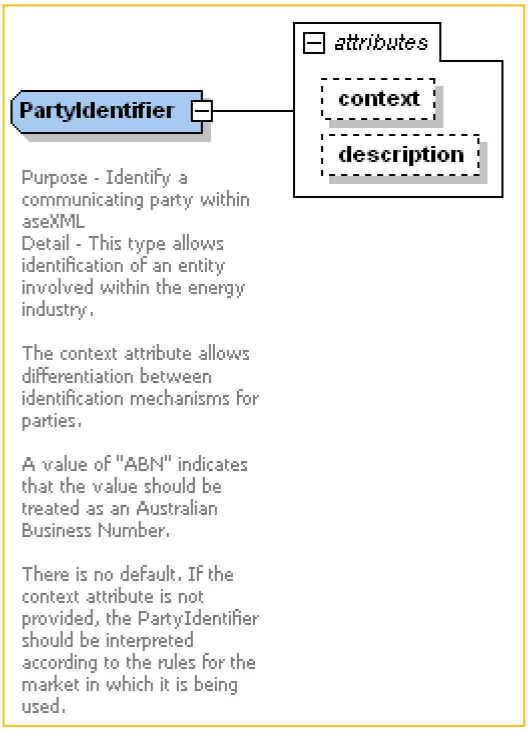
Purpose - Identify a communicating party within aseXML

Detail - This type allows identification of an entity involved within the energy industry.

The context attribute allows differentiation between identification mechanisms for parties.

A value of "ABN" indicates that the value should be treated as an Australian Business Number.

There is no default. If the context attribute is not provided, the PartyIdentifier should be interpreted according to the rules for the market in which it is being used.


attributes
PartyIdentifier
context
description
Purpose - Identify a communicating party within aseXML
Detail - This type allows identification of an entity involved within the energy industry.
The context attribute allows differentiation between identification mechanisms for parties.
A value of "ABN" indicates that the value should be treated as an Australian Business Number.
There is no default. If the context attribute is not provided, the PartyIdentifier should be interpreted according to the rules for the market in which it is being used.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:PartyIdentifier are:

XPath to aseXML node	aseXML node restrictions	aseXML type
@context	enumeration value="ABN"	xsd:string
@description		xsd:string

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:PartyIdentifier		xsd:string

5.1.122 Type ase:PoleNumber (string 1-40 chars.)

The aseXML documentation for type ase:PoleNumber is:

This information is used to help find remote meters that may not be easily identified by address. Often it is easier to find the site via reference to a network pole number (generally in sequence on a feeder). This is required in standing data as the Franchise market Retailer still carries responsibility for performing site activities for some areas of the network.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:PoleNumber	maxLength value="40" minLength value="1" whiteSpace value="collapse"	xsd:string

5.1.123 Type ase:ProfileName (string ≤ 10 chars.)

The aseXML documentation for type ase:ProfileName is:

Purpose - Identify a profile

MSATS Data Model Column - ProfileNameID

Detail - Profiles are applied to consumption meter reads to massage them into a form suitable for settlement of a market based on shorter intervals than provided by the meter read cycle.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:ProfileName	maxLength value="10"	xsd:string

5.1.124 Type ase:ReceiptIdentifier (string, 1-36 chars.)

The aseXML documentation for type ase:ReceiptIdentifier is:

Purpose - Provide a unique identifier to acknowledge a message or transaction.

Detail - A receipt identifier is generated by the receiver of an aseXML message or transaction. Its purpose is to provide the sender with a string that can be quoted when inquiring with regard to the progress of processing for a message or transaction. As implied by the name, it is the sender's guarantee that the reciever will process the message or transaction.

Note that receipt identifiers do not have to be globally unique, only unique to a particular receiver. However, the length has been chosen such that UUIDs can be used as receipt identifiers if considered appropriate, guaranteeing global uniqueness.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:ReceiptIdentifier		*Type ase:UniqueIdentifier (string, 1-36 chars.)* (§5.1.139 on page 77).

5.1.125 Type ase:ReleaseIdentifier (string with pattern)

The aseXML documentation for type ase:ReleaseIdentifier is:

Purpose - Identify a release of aseXML

Details - Release identifiers are used extensively in aseXML to enable versioning of transactions.

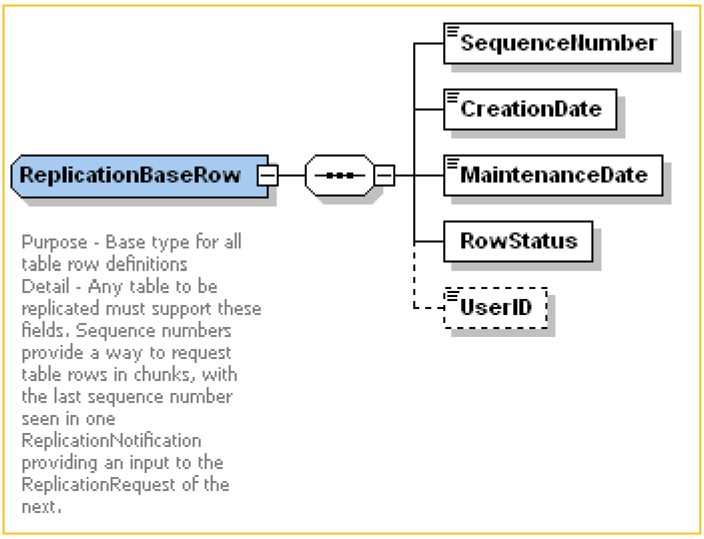
Type ase:ReleaseIdentifier has an aseXML base of xsd:string, restricted to a pattern (specifically xsd:pattern value="r[0-9]*|r[0-9]*_[a-z][0-9]*").

5.1.126 Type ase:ReplicationBaseRow (complex)

The aseXML documentation for type ase:ReplicationBaseRow is:

Purpose - Base type for all table row definitions

Detail - Any table to be replicated must support these fields. Sequence numbers provide a way to request table rows in chunks, with the last sequence number seen in one ReplicationNotification providing an input to the ReplicationRequest of the next.


SequenceNumber
CreationDate
ReplicationBaseRow
MaintenanceDate
RowStatus
UserID
Purpose - Base type for all table row definitions
Detail - Any table to be replicated must support these fields. Sequence numbers provide a way to request table rows in chunks, with the last sequence number seen in one ReplicationNotification providing an input to the ReplicationRequest of the next.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:ReplicationBaseRow, are:

XPath to aseXML node	aseXML node restrictions	aseXML type
SequenceNumber		*Type ase:ReplicationSequenceNumber* (§5.1.127 below)
CreationDate		xsd:dateTime
MaintenanceDate		xsd:dateTime
RowStatus		*Type ase:ActiveInactive (enumerated list)*, (§5.1.3 on page 17)
UserID	minOccurs="0"	xsd:string

5.1.127 Type ase:ReplicationSequenceNumber (

The aseXML documentation for type ase:ReplicationSequenceNumber is:

Purpose - Uniquely identifies each row within a table

MSATS Data Model Column - ID_*

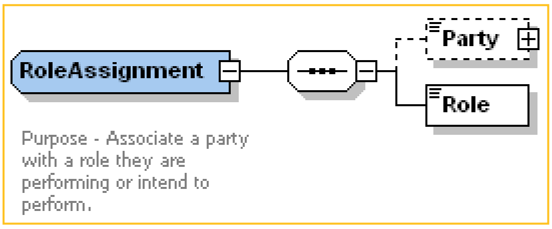
Detail - Sequence numbers on each row allow delivery of large amounts of data via a series of related requests.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:ReplicationSequenceNumber		xsd:nonNegativeInteger

5.1.128 Type ase:RoleAssignment (complex)

The aseXML documentation for type ase:RoleAssignment is:

Purpose - Associate a party with a role they are performing or intend to perform.


RoleAssignment
Party
Role
Purpose - Associate a party with a role they are performing or intend to perform.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:RoleAssignment, are:

XPath to aseXML node	aseXML node restrictions	aseXML type

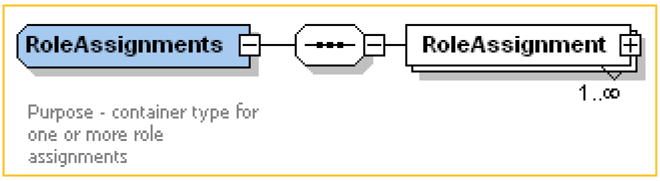
Party	nillable="true"
	minOccurs="0"
		Type ase:PartyIdentifier (complex) (§5.1.121 on page 67)

Role		*Type ase:RoleIdentifier (string, ≤ 4 chars.)* (§5.1.130 below)

5.1.129 Type ase:RoleAssignments (complex)

The aseXML documentation for type ase:RoleAssignments is:

Purpose - container type for one or more role assignments



AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:RoleAssignments, are:

XPath to aseXML node	aseXML node restrictions	aseXML type
RoleAssignment	maxOccurs="unbounded"	*Type ase:RoleAssignment (complex)* (§5.1.128 on page 69)

5.1.130 Type ase:RoleIdentifier (string, ≤ 4 chars.)

The aseXML documentation for type ase:RoleIdentifier is:

Purpose - Identify the roles a party might perform.

MSATS Data Model Column - RoleID

Detail - Roles often form the basis for jurisdictional rules.

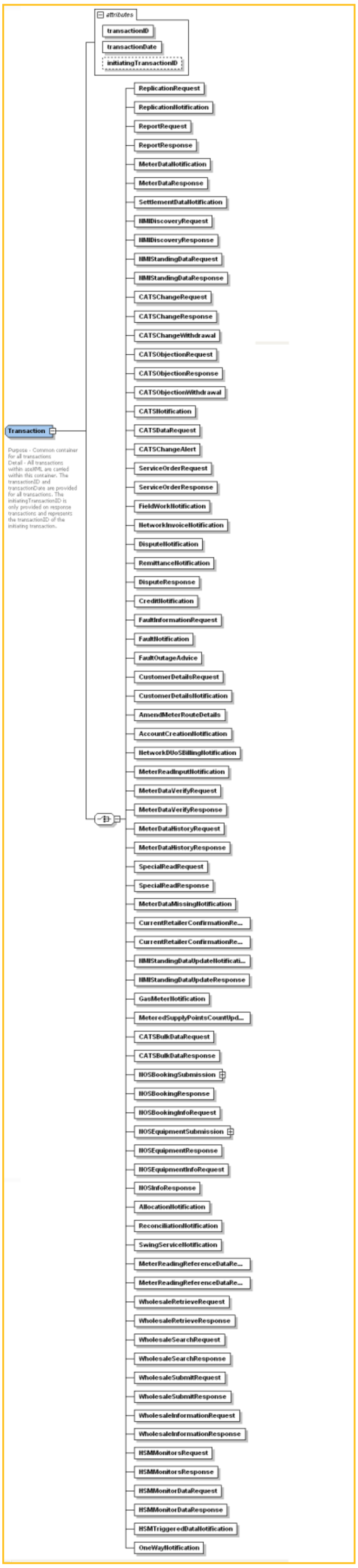
XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:RoleIdentifier	maxLength value="4"	xsd:string

5.1.131 Type ase:Transaction (complex)

The aseXML documentation for type ase:Transaction is:

Purpose - Common container for all transactions

Detail - All transactions within aseXML are carried within this container. The transactionID and transactionDate are provided for all transactions. The initiatingTransactionID is only provided on response transactions and represents the transactionID of the initiating transaction.


Purpose - Common container for all transactions Detail - All transactions within aseXML are carried within this container. The transactionID and transactionDate are provided for all transactions. The initiatingTransactionID is only provided on response transactions and represents the transactionID of the initiating transaction.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:Transaction:

XPath to aseXML node	aseXML node restrictions	aseXML type
@initiatingTransactionID	use="optional"	*Type ase:TransactionIdentifier (string, 1-36 chars.)* (§5.1.134 on page 75)
@transactionDate	use="required"	xsd:dateTime
@transactionID	use="required"	*Type ase:TransactionIdentifier (string, 1-36 chars.)* (§5.1.134 on page 75)
AccountCreationNotification		Unreferenced; see Note 1 below.
AllocationNotification		Unreferenced; see Note 1 below.
AmendMeterRouteDetails		Unreferenced; see Note 1 below
CATSBulkDataRequest		*Type ase:CATSBulkDataRequest (complex)*, (§5.1.31 on page 36)
CATSBulkDataResponse		*Type ase:CATSBulkDataResponse (complex)*, (§5.1.32 on page 36)
CATSChangeAlert		Unreferenced; see Note 1 below.
CATSChangeRequest		Unreferenced; see Note 1 below.
CATSChangeResponse		Unreferenced; see Note 1 below.
CATSChangeWithdrawal		Unreferenced; see Note 1 below.
CATSDataRequest		Unreferenced; see Note 1 below.
CATSNotification		Unreferenced; see Note 1 below.
CATSObjectionRequest		Unreferenced; see Note 1 below.
CATSObjectionResponse		Unreferenced; see Note 1 below.
CATSObjectionWithdrawal		Unreferenced; see Note 1 below.
CreditNotification		Unreferenced; see Note 1 below.
CurrentRetailerConfirmationRequest		Unreferenced; see Note 1 below.
CurrentRetailerConfirmationResponse		Unreferenced; see Note 1 below.
CustomerDetailsNotification		Unreferenced; see Note 1 below
CustomerDetailsRequest)		Unreferenced; see Note 1 below
DisputeNotification		Unreferenced; see Note 1 below.
DisputeResponse		Unreferenced; see Note 1 below.
FaultInformationRequest		Unreferenced; see Note 1 below.
FaultNotification		Unreferenced; see Note 1 below.
FaultOutageAdvice		Unreferenced; see Note 1 below.
FieldWorkNotification		Unreferenced; see Note 1 below.
GasMeterNotification		Unreferenced; see Note 1 below.
HSMMonitorDataRequest		Unreferenced; see Note 1 below.
HSMMonitorDataResponse		Unreferenced; see Note 1 below.
HSMMonitorsRequest		Unreferenced; see Note 1 below.
HSMMonitorsResponse		Unreferenced; see Note 1 below.
HSMTriggeredDataNotification		Unreferenced; see Note 1 below.
MeterDataHistoryRequest		Unreferenced; see Note 1 below.
MeterDataHistoryResponse		Unreferenced; see Note 1 below.
MeterDataMissingNotification		Unreferenced; see Note 1 below
MeterDataNotification		Unreferenced; see Note 1 below
MeterDataResponse		Unreferenced; see Note 1 below.
MeterDataVerifyRequest		Unreferenced; see Note 1 below
MeterDataVerifyResponse		Unreferenced; see Note 1 below.
MeteredSupplyPointsCountUpdate		Unreferenced; see Note 1 below.
MeterReadingReferenceDataRequest		Unreferenced; see Note 1 below.

XPath to aseXML node	aseXML node restrictions	aseXML type
MeterReadingReferenceDataResponse		Unreferenced; see Note 1 below.
MeterReadInputNotification		Unreferenced; see Note 1 below.
NetworkDUoSBillingNotification		Unreferenced; see Note 1 below.
NetworkInvoiceNotification		Unreferenced; see Note 1 below.
NMIDiscoveryRequest		Unreferenced; see Note 1 below.
NMIDiscoveryResponse		Unreferenced; see Note 1 below.
NMIStandingDataRequest		Unreferenced; see Note 1 below.
NMIStandingDataResponse		Unreferenced; see Note 1 below.
NMIStandingDataUpdateNotification		Unreferenced; see Note 1 below.
NMIStandingDataUpdateResponse		Unreferenced; see Note 1 below.
NOSBookingInfoRequest		Unreferenced; see Note 1 below.
NOSBookingResponse		Unreferenced; see Note 1 below.
NOSBookingSubmission		Unreferenced; see Note 1 below.
NOSEquipmentInfoRequest		Unreferenced; see Note 1 below.
NOSEquipmentResponse		Unreferenced; see Note 1 below.
NOSEquipmentSubmission		Unreferenced; see Note 1 below.
NOSInfoResponse		Unreferenced; see Note 1 below.
OneWayNotification		Unreferenced; see Note 1 below
ReconciliationNotification		Unreferenced; see Note 1 below.
RemittanceNotification		Unreferenced; see Note 1 below.
ReplicationNotification		Unreferenced; see Note 1 below.
ReplicationRequest		Unreferenced; see Note 1 below.
ReportRequest		Unreferenced; see Note 1 below.
ReportResponse		Unreferenced; see Note 1 below.
ServiceOrderRequest		Unreferenced; see Note 1 below
ServiceOrderResponse		Unreferenced; see Note 1 below
SettlementDataNotification		Unreferenced; see Note 1 below.
SpecialReadRequest		Unreferenced; see Note 1 below
SpecialReadResponse		Unreferenced; see Note 1 below
SwingServiceNotification		Unreferenced; see Note 1 below.
WholesaleInformationRequest		Unreferenced; see Note 1 below.
WholesaleInformationResponse		Unreferenced; see Note 1 below.
WholesaleRetrieveRequest		Unreferenced; see Note 1 below.
WholesaleRetrieveResponse		Unreferenced; see Note 1 below.
WholesaleSearchRequest		Unreferenced; see Note 1 below.
WholesaleSearchResponse		Unreferenced; see Note 1 below.
WholesaleSubmitRequest		Unreferenced; see Note 1 below.
WholesaleSubmitResponse		Unreferenced; see Note 1 below.

Note:

1. No data objects for BDT refer to the transactions noted as “unreferenced”.

5.1.132 Type ase:TransactionAcknowledgement (complex)

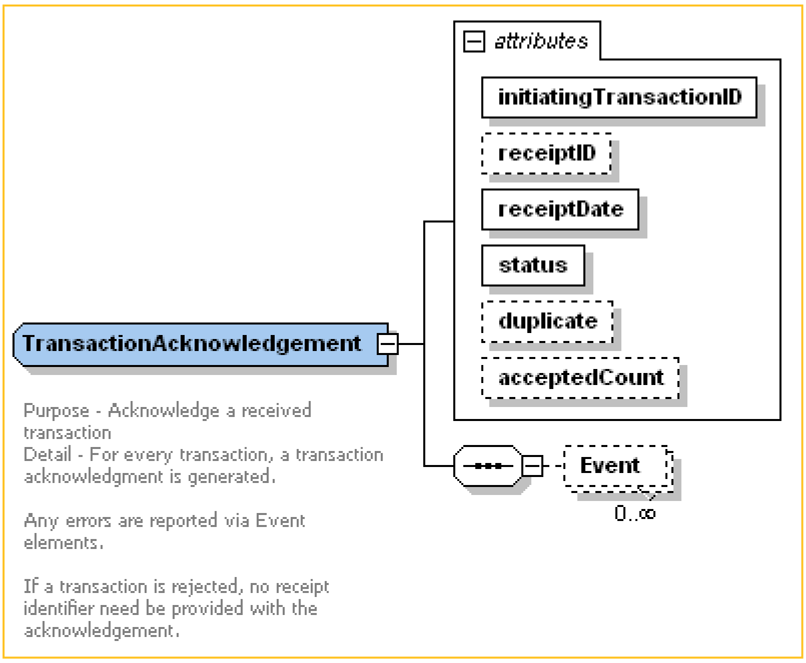
The aseXML documentation for type ase:TransactionAcknowledgement is:

Purpose - Acknowledge a received transaction

Detail - For every transaction, a transaction acknowledgment is generated.

Any errors are reported via Event elements.

If a transaction is rejected, no receipt identifier need be provided with the acknowledgement.


attributes
initiatingTransactionID
receiptID
receiptDate
status
duplicate
acceptedCount
TransactionAcknowledgement
Event
0..∞
Purpose - Acknowledge a received transaction
Detail - For every transaction, a transaction acknowledgment is generated.
Any errors are reported via Event elements.
If a transaction is rejected, no receipt identifier need be provided with the acknowledgement.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:TransactionAcknowledgement:

XPath to aseXML node	aseXML node restrictions	aseXML type
@acceptedCount	use="optional"	xsd:nonNegativeInteger
@duplicate	default="No"	*Type ase:YesNo (enumerated list)* (§5.1.141 on page 77)
@initiatingTransactionID	use="required"	*Type ase:TransactionIdentifier (string, 1-36 chars.)* (§5.1.134 on page 75)
@receiptDate	use="required"	xsd:dateTime
@receiptID	use="optional"	*Type ase:ReceiptIdentifier (string, 1-36 chars.)* (§5.1.124 on page 68)
@status	use="required"	*Type ase:TransactionStatus (enumerated)* (§5.1.137 on page 76)

Event	minOccurs="0"
	maxOccurs="unbounded"
		Type ase:Event (complex) (§5.1.54 on page 50)

5.1.133 Type ase:TransactionGroup (enumerated)

The aseXML documentation for type ase:TransactionGroup is:

Purpose - Provide the transaction group to which all the contained transactions or transaction acknowledgments belong

Detail - The target application is at liberty to reject any transactions within the message that do not belong to the stated TransactionGroup. Where only message acknowledgements are carried, a transaction group of "MSGS" should be used.

The following groups are B2B process related:

FLTS - Faults and Outages

SORD - Service order, planned works

NETB - Network billing

MTRD - Meter data

CUST - Customer related

NOTF - Notifications (broadcast)

MKTW - Wholesale market operations

HSMD - High Speed Monitoring system

OWNP - Own Way Notification

Type ase:TransactionGroup has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- CATS.
- MDMT.
- MSGS.
- NMID.
- FLTS.
- SORD.
- NETB.
- MTRD.
- CUST.
- NOTF.
- SITE.
- FLDW.
- OUTG.
- BAR.
- NMIF.
- MKTW.
- HSMD.
- OWNP.

5.1.134 Type ase:TransactionIdentifier (string, 1-36 chars.)

The aseXML documentation for type ase:TransactionIdentifier is:

Purpose - Uniquely identify every transaction generated by the transaction sender

Detail - Note that transaction identifiers do not have to be globally unique, only unique to a particular sender. However, the length has been chosen such that UUIDs can be used as transaction identifiers if considered appropriate, guaranteeing global uniqueness.

A transaction acknowledgement identifies which transaction it is acknowledging by providing the transaction identifier as an attribute.

In addition to carrying its own unique identifier, a transaction generated in response to the initial transaction of a transaction exchange also carries the transaction identifier of the initial transaction. This allows the sender to determine the context in which the response transaction should be interpreted.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:TransactionIdentifier		*Type ase:UniqueIdentifier (string, 1-36 chars.)* (§5.1.139 on page 77)

5.1.135 Type ase:TransactionPriority (enumerated)

The aseXML documentation for type ase:TransactionPriority is:

Purpose - Provide the processing priority desired by the sender

Detail - The sender can indicate their preference in terms of timeliness of processing for the payload. It is left to the discretion of the receiver to determine whether and how to honour the requested priority.

Type ase:TransactionPriority has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

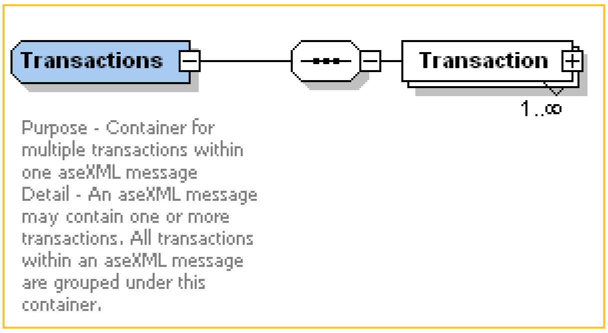
- High.
- Medium.
- Low.

5.1.136 Type ase:Transactions (complex)

The aseXML documentation for type ase:Transactions is:

Purpose - Container for multiple transactions within one aseXML message

Detail - An aseXML message may contain one or more transactions. All transactions within an aseXML message are grouped under this container.



Purpose - Container for multiple transactions within one aseXML message
Detail - An aseXML message may contain one or more transactions. All transactions within an aseXML message are grouped under this container.

AseXML items in alphabetical sequence of XPath, with each XPath relative to the element using the type ase:Transactions are:

XPath to aseXML node	aseXML node restrictions	aseXML type
Transactions	maxOccurs="unbounded"	type="Transaction"

5.1.137 Type ase:TransactionStatus (enumerated)

The aseXML documentation for type ase:TransactionStatus is:

Purpose - Indicate the acceptance, partial acceptance or rejection of the transaction.

Type ase:TransactionStatus has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- Accept.
- Partial.
- Reject.

5.1.138 Type ase:TransmissionNodeIdentifier (string ≤ 4 chars.)

The aseXML documentation for type ase:TransmissionNodeIdentifier is:

Purpose - Transmission node identfiers

MSATS Data Model Column - TNICode

Detail - Transmission nodes identify the points at which the intra-regional loss factors apply

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:TransmissionNodeIdentifier	maxLength value="4"	xsd:string

5.1.139 Type ase:UniqueIdentifier (string, 1-36 chars.)

The aseXML documentation for type ase:UniqueIdentifier is:

Purpose - Uniquely identify the associated data element or attribute.

Detail - Note that unique identifiers do not have to be globally unique, only unique to a particular sender. However, the length has been chosen such that UUIDs can be used as identifiers if considered appropriate, guaranteeing global uniqueness.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:UniqueIdentifier	minLength value="1", maxLength value="36"	xsd:string

5.1.140 Type ase:VoltageType (string 1-10 chars.)

The aseXML documentation for type ase:VoltageType is:

Indicates whether the site is connected at High or Low voltage.

XPath to aseXML node	aseXML node restrictions	aseXML type
element using the type ase:VoltageType	maxLength value="10" minLength value="1" whiteSpace value="collapse"	xsd:string

5.1.141 Type ase:YesNo (enumerated list)

The aseXML documentation for type ase:YesNo is:

Purpose - Yes/No indication

Detail - Note that these values are case sensitive.

Type ase:YesNo has an aseXML base of xsd:string and is restricted to one of the following enumerated values:

- Yes.
- No.

6 References

The resources listed in this section contain additional related information that may assist you.

- AEMO Help Desk: phone: 1300 236 600 (1300 AEMO 00), and follow the prompts; e-mail: helpdesk@aemo.com.au.
- National Gas Rules (NGR) and National Electricity Rules (NER): see the Australian Energy Market Commission (AEMC) website http://www.aemc.gov.au.

6.1 AEMO’s website

The following are available on AEMO’s website:

- *Electricity Market Management Systems Access Policy and Procedure* on web page “Understanding Energy Market Information Systems”. Available http://www.aemo.com.au/registration/infosystems.html (Home > Energy Market Registration > Understanding Energy Market Information Systems). Viewed 29 March 2012).
- “MSATS Procedures Consultation v3.6” web page is at http://www.aemo.com.au/electricityops/msats3-6.html, (Home > Electricity Retail & Metering > MSATS > MSATS Procedures Consultation v3.6). Viewed 29 February 2012.
- “CATS and WIGS Procedures” web page is at http://www.aemo.com.au/electricityops/cats_wigs.html, (Home > Electricity Retail & Metering > MSATS > CATS and WIGS Procedures). Viewed 29 February 2012.
- “aseXML Standards” is at http://www.asexml.com. Viewed 29 March 2012).
- “EITS Publications” web page is a secured part of AEMO’s website, at http://www.aemo.com.au/eits/eits.html. Viewed 29 March 2012.
- *Guide to Market Systems - Gaining Access* is on http://www.aemo.com.au/registration/nemnet.html, (Home > Energy Market Registration > Understanding Energy Market Information Systems). Viewed 29 March 2012.
- *EMMS Data Interchange Guide* is on the “EITS Publications” secured web page, http://www.aemo.com.au/eits/eits.html, under the link to EMMS - Data Interchange. Viewed 29 March 2012).
- “MSATS Participant User Interface Guides” web page on AEMO’s public website (at http://www.aemo.com.au/electricityops/userguide.html, Home > Electricity Retail & Metering > MSATS > MSATS Participant User Interface Guides, viewed 29 March 2012).
- “Participant Batcher” on secured part of AEMO’s website (at http://www.aemo.com.au/eits/eits.html, under the link EMMS - Participant Data Replication (PDR), viewed 29 March 2012).
- *Web Portal Login User Guide* on secured part of AEMO’s website (http://www.aemo.com.au/eits/eits.html under the link AEMO General, viewed 29 March 2012).

6.2 ParticipantID’s IT security contacts

The participantID’s IT security contacts are AEMO’s primary contact point for all issues affecting the shared responsibility of security of data and access to AEMO’s systems. All participantID’s users need to be aware of the contents of *Electricity Market Management Systems Access Policy and Procedure* (see “References” on page 78).

6.3 AEMO Help Desk

The AEMO Help Desk offers assistance for IT-related issues of AEMO-supported products, such as installation, and systems administration support.

Telephone: 1300 AEMO 00 (1300 2366 00), option 1 then 2.

E-mail: helpdesk@aemo.com.au.