B2B PROCEDURE: TECHNICAL DELIVERY SPECIFICATION

PREPARED BY: AEMO Markets

VERSION: 3.~~7~~8

EFFECTIVE DATE: ~~7 November 2022~~May 2023

STATUS: ~~Final~~Initial

Approved for distribution and use by:

APPROVED BY: Information Exchange Committee

DATE: ~~14 September 2021~~TBA

VERSION RELEASE HISTORY

Version	Effective Date	Summary of Changes
1.6.1	18/08/2009	Update to reflect change of governance from NEMMCO to AEMO. Update the version number and release date to retain version numbering with the other B2B Procedures. Minor updates to sections 1.2, to align clauses with the other B2B Procedures. Issued as Determination – Effective 25 November 2009
1.7	17/03/2010	Updated version numbers and release date to retain version numbering with other B2B Procedures. Graphical updates to diagrams. Update Clause 1.6. Updates to all clauses Section 6 Issued as Final Determination – effective 26 May 2010
1.7a	15/07/2011	Updated version number to 1.7a and release date to retain version numbering with other B2B Procedures. Updated procedure to facilitate further extension of contestability to small business customers in Tasmania.
1.8	15/08/2011	Updated version numbers and release date to retain version numbering with other B2B Procedures. Updates to Clause 3.5, 5.8 and Section 6.
1.9	06/11/2012	Updated version numbers and release date to retain version numbering with other B2B Procedures. Update to clause 1.6 a. for Meter Data Providers.
2.0	13/11/2013	Updates to capture QC 776 CSND Project Changes
2.1	15/05/2014	Update reference for Service Paperwork Reference Table
2.2	21/11/2014	Updated version numbers and release date to retain version numbering with other B2B Procedures.
3.0	06/03/2017	Updates as part of the following: • National Electricity Amendment (Expanding Competition in Metering and Related Services) Rule 2015 No. 12; • National Electricity Amendment (Embedded Networks) Rule 2015 No. 15; and • National Electricity Amendment (Updating the Electricity B2B Framework) Rule 2016 No. 6.
3.1	01/12/2017	Updated for IEC B2B Errata
3.2	01/02/2019	Updated version numbers and release date to retain version numbering with other B2B Procedures.
3.3	10/12/2019	Updated version numbers and release date to retain version numbering with other B2B Procedures.
3.31	1/1/2020	Updated version numbers and release date to retain version numbering with other B2B Procedures.
3.4	1/10/2021	Updated internal document references, file size and transaction number limits in clause 5.8(a). Megabyte defined in section 1.6 for clarity
3.4.1	1/10/2021	Updated version numbers and release date to retain version numbering with other B2B Procedures.
3.5	10/11/2021	Updated to reflect NCONUML changes

Version	Effective Date	Summary of Changes
3.6	10/11/2021	Updated version numbers and release date to retain version numbering with other B2B Procedures.
3.6.1	10/11/2021	Updated version numbers and release date to retain version numbering with other B2B Procedures.
3.6.2	10/11/2021	Placeholder for v3.6.2
3.7	7/11/2022	Updated to include the following changes: - Update requirements for NameTitleField ICF - Update requirements for PersonNameGiven Field ICF
3.8	May 2023	Updated to include shared fuse arrangement transaction

CONTENTS

1.	INTRODUCTION	**7**
1.1.	Purpose and Scope	7
1.2.	Definitions and Interpretation	7
1.3.	Related Documents	7
1.4.	Background	8
1.5.	Availability	8
1.6.	Terminology	9
2.	MESSAGE FORMAT REQUIREMENTS	**14**
2.1.	Overview	14
2.2.	General aseXML Conventions	15
2.3.	aseXML Acknowledgements	15
2.4.	Clearly Identifying aseXML transactions and acknowledgements	15
2.5.	Naming aseXML Transactions	16
2.6.	Identifying field names in an aseXML transaction or acknowledgement	16
2.7.	Naming fields in an aseXML transaction or acknowledgement	16
2.8.	Referring to an aseXML "sub-transaction"	16
2.9.	aseXML Error Reporting and Handling	16
2.10.	aseXML Events	16
2.11.	Mapping Business Documents to aseXML Transactions	17
2.12.	Mapping Business Signals to aseXML Acknowledgements	18
2.13.	Message Format	18
3.	FIELD FORMAT CONVENTIONS	**19**
3.1.	Use of standardised format conventions for fields	19
3.2.	Basic field formats	19
3.3.	User-defined field formats	21
3.4.	ADDRESS definition	21
3.5.	PERSONNAME definition	24
3.6.	TELEPHONE definition	24
3.7.	Fields that contain codes or enumerated lists	25
4.	PAYLOAD DEFINITIONS	**25**
4.1.	CSV Notification Detail	25
5.	TRANSACTION DELIVERY REQUIREMENTS	**27**
5.1.	Delivery Mechanisms	27
5.2.	Participant Addressing	27
5.3.	Message Equivalents	28
5.4.	Overview of MSATS B2B Handler Functionality (FTP)	28
5.5.	Overview of SMP Hub Functionality (Webservices)	32
5.6.	Authentication and Non-repudiation	36

5.7. Priority of aseXML Messages 37
5.8. Size of aseXML Messages 37
5.9. Timing Requirements 38
5.10. Transaction Logging 39
5.11. Handling of duplicate or resent Transactions and Messages 40
5.12. Timestamps 41

6. TRANSACTION MODELS **42**
6.1. Background 42
6.2. Delivery Protocols 42
6.3. Changing Protocols 42
6.4. MSATS B2B Handler Transaction Flow Model 42
6.5. SMP Hub Transaction Flow Model 58
6.6. A Summary of Transaction Model Exception Points 64

7. INTEROPERABILITY **65**
7.1. Webservices to FTP Hokey Pokey 65
7.2. FTP Hokey Pokey to Webservices 66

8. NOTIFIED PARTIES **67**
8.1. Overview 67
8.2. Sequence Diagrams – e-Hub generated notifications 67
8.3. Trigger Points for Notifications 77
8.4. Notification Format 77
8.5. Notification Bundling 77
8.6. Error Handling 77

9. CONTINGENCY RECOVERY REQUIREMENTS **78**
9.1. Overview of National B2B Infrastructure 78
9.2. Need for Contingency Arrangements 78
9.3. Basic Principles for Contingency Arrangements 79
9.4. Overview of Major Contingency Requirements 79
9.5. Major Failure Events and Contingency Steps 80
9.6. Contingency Messages 81
9.7. Use of the B2B Browser Application as a Contingency Solution 81
9.8. Use of Email as a Contingency Solution 82
9.9. Use of Telephone and Fax 82
9.10. Notification and Activation Requirements 82
9.11. Prioritisation of Transactions 83
9.12. Handling of contingency Transactions once normal operations resume 83

TABLES

Table 1 Business Document to aseXML mapping 17
Table 2 Business Signal to aseXML mapping 18
Table 3 Basic field formats 19
Table 4 Address field definition 22
Table 5 Person Name field definition 24
Table 6 Telephone field definition 25
Table 7 CSV field definition 26
Table 8 Message terminology by protocol 28
Table 9 Timing Requirements 39
Table 10 Timing Requirements (webservices only) 39
Table 11 Data Logging Requirements 40
Table 12 Transaction Action Matrix for negative acknowledgements 40

FIGURES

Figure 1 Summary of Timing Requirements 38
Figure 2 FTP Activity Diagram - Acknowledgement Model for normal processing 44
Figure 3 FTP Sequence Diagram - Acknowledgement Model for normal processing 45
Figure 4 FTP Activity Diagram - Acknowledgement Model for invalid aseXML Message 48
Figure 5 FTP Sequence Diagram - Acknowledgement Model for invalid aseXML Message 49
Figure 6 FTP Activity Diagram - Acknowledgement Model when Recipient Outbox full 51
Figure 7 FTP Sequence Diagram - Acknowledgement Model when Recipient Outbox full 52
Figure 8 FTP Activity Diagram - Acknowledgement Model for MACK validation failure 54
Figure 9 FTP Sequence Diagram - Acknowledgement Model for MACK validation failure 55
Figure 10 Example of FTP Transaction Model for Service Order process 57
Figure 11 Webservices Sequence Diagram - Acknowledgement Model normal processing 59
Figure 12 Webservices Sequence Diagram – Acknowledgement Model for invalid aseXML Message 60
Figure 13 Webervices Sequence Diagram – Acknowledgement Model when Recipient unavailable 61
Figure 14 Webservices sequence diagram – message acknowledgement validation failure 63
Figure 15 Example of Webservices Transaction Model for Service Order process 64
Figure 16 Interoperability – Webservices to FTP sequence diagram 66
Figure 17 Interoperability – FTP to Webservices sequence diagram 66
Figure 18 e-Hub generated notifications – normal processing 68
Figure 19 Notified Parties – Recipient sends negative message acknowledgement 70
Figure 20 Notified Parties – Recipient sends a Business Rejection 71
Figure 21 Notified Parties – Incorrect Notified Party 72
Figure 22 Notified Parties – Notified Party Stopped 75
Figure 23 National B2B Infrastructure 78

1. INTRODUCTION

1.1. Purpose and Scope

(a) This B2B Procedure: Technical Delivery Specification (Procedure) is *published* by AEMO in accordance with clause 7.17.3 of the NER and specifies the technical requirements for the delivery of B2B Transactions using the e-Hub.

(b) This Procedure defines Participant interactions with the e-Hub.

(c) This Procedure also defines baseline configuration settings applicable to the e-Hub for the delivery of (national) B2B Transactions (i.e. the configuration of the e-Hub that is required by the industry to support National B2B Standards).

(d) This Procedure also considers contingency arrangements relevant to Participants and the National B2B Infrastructure. Refer to Figure 23 for a diagram illustrating the National B2B Infrastructure.

(e) This Procedure only applies to the B2B Transactions identified in Section 2.11 of this Procedure;

(f) This Procedure does not apply to internal processes or technical infrastructure requirements, specific to the DNSP, Retailer or other relevant Participants, except where there are prescribed connectivity or contingency requirements.

(g) This Procedure does not describe free-form messaging or additional functionality supported by the SMP Hub. Refer to the SMP Technical Guide for further optional functionality.

1.2. Definitions and Interpretation

(a) The Retail Electricity Market Procedures – Glossary and Framework:

(i) is incorporated into and forms part of this Procedure; and

(ii) should be read with this Procedure.

(b) Grey boxes are used throughout this document to indicate that the information contained within are for guidance and informational purposes only.

(c) Technical terminology used throughout the B2B Procedures have been defined in section 1.6 of this document.

1.3. Related Documents

Title	Location
Retail Electricity Market Procedures – Glossary and Framework	http://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Glossary-and-Framework
B2B Procedure Service Order Process	http://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Business-to-business-procedures
B2B Procedure Meter Data Process	http://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Business-to-business-procedures
B2B Procedure Customer and Site Details Notification Process	http://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Business-to-business-procedures
B2B Procedure One Way Notification Process	http://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Business-to-business-procedures
Metrology Procedure: Part A	http://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering

Title	Location
Guidelines for Development of A Standard for Energy Transactions in XML (aseXML), also known as the 'aseXML Guidelines'.	http://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/IT-systems-and-change/aseXML_standards/aseXML-Guidelines
B2B Mapping to aseXML	http://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering/Business-to-business-procedures
SMP Technical Guide	http://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering
B2B Guide	http://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Retail-and-metering

1.4. Background

(a) AEMO must provide and operate the e-Hub for the delivery of B2B Transactions. As required by this Procedure and the B2B Procedures, Participants must use the MSATS B2B Handler or SMP Hub for B2B Transactions.

(b) The MSATS B2B Handler is essentially an extension of the MSATS Batch Handler (refer to 010905 - Technical Architecture Design Report v4.4, Appendix B - Batch Handler BI007 TSD for details).

(c) The MSATS B2B Handler supports the transfer of compressed ("zipped") aseXML files directly between market Participants.

(d) The SMP Hub supports the transfer of aseXML B2B messages between market Participants.

(e) The functionality available via the e-Hub includes:

(i) the ability for B2B files to be sent to Participant directories, as specified;

(ii) header and schema validation of files;

(iii) support for specific B2B Transaction types; and

(iv) logging of handler activity in an activity log.

1.5. Availability

(a) Each Participant and AEMO agrees to use reasonable endeavours to make that portion of the National B2B Infrastructure over which they have control and for which they are responsible available at all times. However each Participant and AEMO are not able to guarantee the provision of a continuous and fault free National B2B Infrastructure for various reasons, including:

(i) the conduct of a user of the National B2B Infrastructure;

(ii) an electrical or telecommunications fault or failure;

(iii) an emergency or fault rectification procedure;

(iv) scheduled maintenance;

(v) a fault, virus, security breach or breakdown;

(vi) an event of force majeure.

(b) All obligations imposed on a Participant and/or AEMO in this Procedure must be read subject to clause (a) above.

1.6. Terminology

Term	Definition
Accept	As a general term, this means the Recipient of the Message or Transaction has agreed to process the Message or Transaction further. When used in the context of a Transaction, indicates that the Recipient of the Transaction has accepted the Transaction using a *BusinessAcceptance/Rejection* with an *ase:Status* of "Accept".
Acknowledgement	See B2B Acknowledgement.
Acknowledgement File	A file containing a Message Acknowledgement.
Approved Version of the Schema	A version of the aseXML schema approved by the aseXML Working Group, or its successors.
aseXML	A Standard for Energy Transactions in XML. A set of schemas and usage guidelines that define how data should be exchanged under FRC in the gas and electricity industries in Australia.
aseXML Document	See aseXML Message.
aseXML Message	A Message compliant with an aseXML Schema.
aseXML Message Handler	Software that manages aseXML Message interactions.
aseXML Schema	Specification used to describe the structure of an aseXML Message.
aseXML Transaction	See Transaction.
aseXML Wrapped CSV Transaction	An aseXML Transaction that includes CSV formatted data.
B2B	Business-to-Business. Generic term used to refer to defined business-to-business interactions between Participants; excludes interactions between a Participant and market systems such as MSATS.
B2B Acknowledgement	A generic term used to refer to an aseXML-format Message or Transaction Acknowledgement, specifically within the context of a B2B interaction. A B2B Acknowledgement is the physical interpretation of a Business Signal. Often referred to as being positive (indicating correctness of the associated file) or negative (indicating an error with the associated file).

Term	Definition
B2B Browser Application	An application supplied by AEMO for Participants to manage their MSATS B2B Handler Inbox(es) and Outbox(es), and also support the creation of a specified set of B2B Transactions.
B2B File	See B2B Message.
B2B Infrastructure	See National B2B Infrastructure.
B2B Interaction	A complete set of related exchanges of B2B Messages between two Participants involving: • Business Document; • Business Receipt; and • Business Acceptance/Rejection.
B2B Message	A B2B Transaction or Acknowledgement sent between a B2B Initiator and a B2B Recipient.
B2B Process	A defined business process of which a B2B Interaction is a key component. Identified B2B Processes are: • Customer and Site Details • Meter Data • One Way Notification; and • Service Orders.
B2B Standard	A collection of B2B Procedures and supporting documentation that collectively form a coherent set of requirements (an industry "B2B Standard"). The components of a B2B Standard are described by the B2B Standards Framework.
B2B Standards Framework	Describes the components of a B2B Standard.
B2B Transaction	See Transaction.
B2B Transaction Types	The Transactions defined in the B2B Procedures.
Build Pack	A document that details the specific aseXML interfaces to be used in the implementation of B2B transactions.
Business Acceptance	Specific instance of a Business Acceptance/Rejection Business Signal indicating acceptance.

Term	Definition
Business Acceptance/Rejection	A Business Signal indicating whether a Business Document has been accepted or rejected based on the application of business rules. Refer to each B2B Procedure for further details regarding the use of this Transaction.
Business Receipt	A Business Receipt is a Business Signal that indicates that a Business Document has been received and its contents indicates if it is readable by the recipient.
Business Rejection	Specific instance of a Business Acceptance/Rejection Business Signal indicating a rejection.
CSV Notification Detail	The description for the CSV Notification Pay load described in the B2B Procedure One Way Notification Process.
ebXML	Electronic Business XML. An internationally developed set of protocols for using XML. ebXML is a wide ranging standard that covers all layers of messaging. VicGas FRC project has adopted only the transport, routing and packaging (TRP) recommendations of ebXML, and is using aseXML to define standards for message content.
Event Code	A specific code used to refer to a Business Event defined in a B2B Procedure.
e-Hub	Collective term used to refer to both the MSATS B2B Handler (FTP) and SMP Hub (Webservices).
FAQs	Frequently Asked Questions. Used to provide supplementary answers to questions raised regarding the interpretation of the B2B Procedures.
File Limit	Refers to the number of files in an Inbox or Outbox at which point the B2B Handler will generate a flow control file.
File Transfer and Acknowledgement Protocol	See MSATS File Exchange Protocol.
FRC	Full Retail Contestability. The state of the energy market when all customers can be transferred between Retailers.
FTP	File Transfer Protocol.
Fully Tagged aseXML Transactions	An aseXML Transaction not containing a .CSV payload.
Hokey-Pokey	See Hokey-Pokey Protocol.

Term	Definition
Hokey-Pokey Protocol	See MSATS File Exchange Protocol.
Hub Acknowledgement	A Message Acknowledgement generated by the e-Hub.
Inbox	See MSATS B2B Handler Inbox.
Initiating Message	The first Message in a series of related Messages.
Interoperability	The term used to describe the operation and interaction between multiple delivery protocols.
MB	Megabyte, which consists of 1024 kilobytes
Message	See B2B Message.
Message Acknowledgement	An aseXML realisation of a Business Receipt.
MSATS B2B Handler	An extension of the MSATS batch handler to manage B2B transactions.
MSATS B2B Handler Inbox	The file directory where Participants publish B2B messages and acknowledgements for other Participants.
MSATS B2B Handler Outbox	The file directory where Participants receive B2B messages and acknowledgements from other Participants.
MSATS File Exchange Protocol	The file exchange protocol used by MSATS, as described in 010905 – Technical Architecture Design Report v4.4 (as amended from time to time).
MSATS Notification	As defined in MSATS Procedure CATS Procedure.
National B2B Infrastructure	Infrastructure (software and hardware) that physically enables B2B communication between Participants. This includes, but is not necessarily limited to: • MSATS B2B Handler (software and hardware); • MSATS B2B Gateways; • SMP Hub; • Communications between Participants and MSATS B2B Gateways; and • Participant Gateways.
NCONUML	Non-contestable *Unmetered Load*

Term	Definition
Notification Business Transaction Pattern	A B2B Interaction characterised by one Participant sending a Notification transaction (eg CustomerDetailsNotification) to another Participant without a corresponding reply Transaction.
Outbox	See MSATS B2B Handler Outbox.
Participant B2B System	The computer hardware and software used by a Participant to create, send, receive and process B2B Messages.
Participant Directories	Participant Inbox and Outbox used by the B2B Handler.
Participant Gateways	Hardware and software used by a Participant to send and receive B2B files.
Put process	The FTP 'Put" command. Used to copy files between Participant In- and Outboxes.
Reject	When used in the context of a Transaction, indicates that the Recipient of the Transaction has rejected the Transaction using a *BusinessAcceptance/Rejection* with an *ase:Status* of "Reject".
Request and Response Transactions	See Request/Response Business Transaction Pattern.
Request/Response Business Transaction Pattern	A B2B Interaction characterised by one Participant sending a Request transaction (eg ServiceOrderRequest) to another Participant and the other Participant responding with a corresponding Response transaction (eg ServiceOrderResponse). In some cases a Notification may be sent in response to a Request transaction (eg a MeterDataNotification providing the metering data requested in a ProvideMeterDataRequest).
Schema	See aseXML Schema.
SMP	Shared Market Protocol.
SMP Hub	The new platform implemented as part of the B2B Framework changes, which uses the webservices protocol.
Stop File	A file that is generated by the e-Hub when a Participant's FTP Outbox or Websevices message queue exceeds the Water Mark – Warn/High.
Stopbox	The file directory where Participants receive stop files related to the unavailability of other Participants in the Market.

Term	Definition
Stopped	Used to describe the state of a Participant that has a Stop File in place.
Transaction	An aseXML realisation of a Business Document.
Transaction Acknowledgement	An aseXML realisation of a Business Acceptance/Rejection.
Transaction Group	The Transaction Group field in aseXML Message.
Transaction Model	The physical exchange of B2B messages to complete a B2B interaction.
Transaction Priority	An element on an aseXML message that allows the sender to indicate their preference in terms of timeliness of processing for the message contents. The three allowable values are "High", "Medium" and "Low". As used in terms such as 'Medium Priority' or 'Low Priority'.
UML	Unified Modelling Language. A convention adopted for drawing process flow diagrams (activity diagrams) and sequence diagrams.
Water Mark – High	A high water mark is an upper limit of a message queue scale. When a message queue reaches this limit a Stop File is generated in the Outbox of the Participant.
Water Mark – Low	A low water mark is a lower limit of a message queue scale. When a participant message queue reaches this limit a Stop File would be removed if present.
Water Mark – Warn	A warn water mark is a warning limit of a message queue scale. When a message queue reaches this limit a Stop File is placed in the Stopbox of all the Participants stating that the impacted Participant is having issues in processing the files/messages.
XML	eXtensible Markup Language

2. MESSAGE FORMAT REQUIREMENTS

2.1. Overview

(a) B2B Procedures define a series of B2B Interactions as Business Documents or Business Signals.

(b) Business Documents are Notifications, Requests or Responses between Participants and contain important relevant business information.

(c) Business Signals are used to indicate the receipt, acceptance/rejection of a Business Document.

(d) Business Documents and Business Signals are mapped onto aseXML Transactions and Acknowledgements, respectively.

2.2. General aseXML Conventions

(a) Participants must ensure that all B2B Interactions comply with the requirements for the aseXML protocol as defined in the aseXML Guidelines subject to the provisions of this Procedure.

(b) A Participant must ensure that their aseXML Message Handler implements the Acknowledgement model as defined in the aseXML Guidelines, and subject to the provisions of this Procedure.

2.3. aseXML Acknowledgements

(a) A Participant receiving a Message must ensure that an *ase:MessageAcknowledgement* is generated for every aseXML Message received.

(b) A Participant receiving a Transaction must ensure that an *ase:TransactionAcknowledgement* is generated for every Business Document that has passed validations associated with generating an *ase:MessageAcknowledgement*.

2.4. Clearly Identifying aseXML transactions and acknowledgements

(a) The prefix "ase:" (without quotes) is used to differentiate between (logical) Business Documents/Business Signals/fields and (physical) aseXML equivalents.

(b) A prefix of "ase:" (without quotes) followed by the name of the aseXML Transaction is used to identify an aseXML Transaction and provide differentiation from the related Business Document. The full name of the aseXML Transaction, including the prefix, is underlined. For example: ase:ServiceOrderRequest is a valid identifier for an aseXML Transaction representing a ServiceOrderRequest Business Document.

(c) The term *ase:MessageAcknowledgement* is used to represent the aseXML equivalent of a *BusinessReceipt*. As shown, the term is italicised, underlined and prefixed by "ase:".

(d) As an abbreviation, the term *MsgAck* may be used to indicate an *ase:MessageAcknowledgement* with a value of attribute "status" *ase:MessageAcknowledgement/@status*="Accept" (see convention in section 2.6~~2.6~~).

(e) As an abbreviation, the term *MsgNack* may be used to indicate an *ase:MessageAcknowledgement* with a value of attribute "status" *ase:MessageAcknowledgement/@status*="Reject" OR a standalone *ase:Event* (i.e. a "negative acknowledgement"). See convention in section 2.6~~2.6~~.

(f) The phrase "positive Acknowledgement" refers to an *ase:TransactionAcknowledgement* with a *Status* of either "Accept" or "Partial" or *ase:MessageAcknowledgement* with a *Status* of "Accept.

(g) The phrase "negative Acknowledgement" refers to either an *ase:TransactionAcknowledgement* or *ase:MessageAcknowledgement* with a *Status* of "Reject", or an *ase:Event*.

(h) The phrase "positive ase:MessageAcknowledgement" refers to an ase:MessageAcknowledgement with a Status of "Accept".

(i) The phrase "positive ase:TransactionAcknowledgement" refers to an ase:TransactionAcknowledgement with a *Status* of either "Accept" or "Partial".

2.5. Naming aseXML Transactions

(a) Generally, the name of an aseXML Transaction is derived from the name of the originating Business Document, except where an alternative "mapping" has been specified in the appropriate process-related "Business Document Mapping to aseXML".

(b) aseXML transaction names use upper camel case (excepting the "ase:" prefix).

2.6. Identifying field names in an aseXML transaction or acknowledgement

(a) A prefix of "ase:" and the use of italics are used to identify fields within an aseXML transaction or acknowledgement.

(b) XML is case sensitive; Participants must ensure that the aseXML "field" names must match exactly the definitions with the aseXML schema.

(c) Fields may be implemented as "Elements" and "Attributes". The XML specification defines Element field names as upper camel case, and Attribute field names as lower camel case.

(d) As a minimum, every Element name must be prefixed with "ase:", for example *ase:ServiceOrderType*.

(e) As a minimum, every Attribute name must be prefixed with "ase:" and the name of the parent element, followed by the literals "/@", for example ase:ServiceOrderRequest/@actionType.

(f) The full path to the data field may also be used as per the XPath specification, for example:

- *ase:aseXML/Transactions/Transaction/ServiceOrderRequest/ServiceOrder/ServiceOrderType*
- *ase:aseXML/Transactions/Transaction/ServiceOrderRequest/@actionType*.

2.7. Naming fields in an aseXML transaction or acknowledgement

(a) The actual field names used in an aseXML Transaction or Acknowledgement (as distinct from the field names proposed in the definition of a Business Document or Business Signal) are established in the appropriate "Business Document Mapping to aseXML" for the process area. These names are as implemented in the aseXML Schema.

2.8. Referring to an aseXML "sub-transaction"

(a) A Business Document may be physically implemented as an aseXML Transaction or sub-transaction.

(b) An aseXML sub-transaction is referred to using the standard "ase:" prefix, the name of the "parent" transaction, the literal "/" followed by the name of the sub-transaction. For example: *ase:AmendMeterRouteDetails/AmendSiteAccessDetails*.

2.9. aseXML Error Reporting and Handling

(a) Participants must ensure that error reporting and handling complies with aseXML Guidelines subject to the provisions of this Procedure.

2.10. aseXML Events

(a) All aseXML Transaction Acknowledgements and some response Transactions may contain the aseXML Event element; usage must be as defined in the aseXML Guidelines.

(b) Participants must use reasonable endeavours to ensure that a generic Event Code is only used where a specific Event Code does not apply.

2.11. Mapping Business Documents to aseXML Transactions

(a) Participants must ensure that Business Documents are physically realised in aseXML as Transactions, in accordance with the following table:

Table 1 Business Document to aseXML mapping

Process Area	Business Document	aseXML Transaction	Transaction Group	Description
Meter Data	MeterDataNotification	ase:MeterDataNotification	MTRD	Meter Readings (includes CSV component in a valid MDFF).
	ProvideMeterDataRequest	ase:MeterDataMissingNotification	MTRD	Request for meter data
	VerifyMeterDataRequest	ase:MeterDataVerifyRequest	MTRD	Request for meter data to be verified
	RemoteServiceRequest	ase:RemoteServiceRequest	MRSR	Remote Meter service request
	RemoteServiceResponse	ase:RemoteServiceResponse	MRSR	Response to remote meter service request
Service Orders	ServiceOrderRequest	ase:ServiceOrderRequest	SORD	Service Order Request
	ServiceOrderResponse	ase:ServiceOrderResponse	SORD	Service Order Response
Customer Data	CustomerDetailsNotification	ase:CustomerDetailsNotification	CUST	Customer Details Notification
	CustomerDetailsRequest	ase:CustomerDetailsRequest	CUST	Request for a customer details notification
	LifeSupportNotification	ase:LifeSupportNotiication	CUST	Life Support Notification
	LifeSupportRequest	ase:LifeSupportRequest	CUST	Request for a life support notification
	SiteAccessRequest	ase:SiteAccessRequest	SITE	Request for a site access details notification
	SiteAccessNotification	ase:AmendMeterRouteDetails/AmendSiteAccessDetails	SITE	Updated site access details notification
One Way Notification	OneWayNotification	ase:CSVNotificationDetail	OWNP	The payload for the One Way Notification Process.
	PlannedInterruptionNotification	ase:PlannedInterruptionNotification	OWNX	Notification of a planned interruption
	MeterFaultandIssueNoification	ase:MeterFaultandIssueNotification	OWNX	Notification of a meter fault or issue
	NoticeOfMeteringWorks	ase:NoticeOfMeteringWorks	OWNX	Notice of metering works
	NotifiedParty	ase:NotifiedPartyNotification	NPNX	Used to facilitate interaction with Notified Parties
	SharedFuseArrangement	ase:SharedFuseArrangement	OWNX	Notification of shared fuse arrangement at a NMI

2.12. Mapping Business Signals to aseXML Acknowledgements

(a) Business Signals are physically realised in aseXML as Message and Transaction Acknowledgements (or negative Acknowledgements), in accordance with the following table:

Table 2 Business Signal to aseXML mapping

Process Area	Business Signal	aseXML Equivalent	Description
All	*BusinessReceipt*	*ase:MessageAcknowledgement* Or *ase:Event*	A *BusinessReceipt* may be communicated as a *MessageAcknowledgement* or as an *Event*
	BusinessAcceptance/ Rejection	*ase:TransactionAcknowledgement*	

2.13. Message Format

(a) An aseXML Message may contain one or more aseXML Transactions. A Participant must:

(i) use reasonable endeavours to bundle Transactions in order to support efficient Message handling;

(ii) ensure that bundling of Transactions does not reduce their ability to meet the Timing Requirements for the delivery of Transactions;

(iii) ensure that only Transactions of the same Transaction Group are included in the same Message; and

(iv) use reasonable endeavours to ensure that only Transactions of the same Transaction Priority (as defined in each B2B Procedure) are included in the same Message.

(b) If sent using FTP, a Participant must ensure that the Message *Priority* must match the priority character in the file name (refer 5.4.5 (a)(i)).

(c) A Participant must ensure that an aseXML Message complies with the restrictions set out in Section 5 of this Procedure.

(d) Only one aseXML version (as defined in the aseXML Guidelines) of a B2B Transaction will be implemented by Industry at any given time.

(e) AEMO must ensure that the e-Hub generates a Hub Acknowledgement or negative Acknowledgement (.ac1 file in the case of the MSATS B2B Handler) in the same version of the schema as the received Message. Where the schema version of the file cannot be determined by the e-Hub, AEMO must ensure that the e-Hub will generate the Hub Acknowledgement in a default schema version.

(f) Participants must generate the Message Acknowledgment in the current and approved version of the aseXML Schema.

(g) Participants must ensure that their B2B System is capable of receiving aseXML Messages in any version of the aseXML Schema that is approved and effective for the applicable B2B Procedure pursuant to the aseXML Guidelines.

(h) Participants must ensure that they generate aseXML Messages in a version of the aseXML Schema that is approved and effective for the applicable B2B Procedure pursuant to the aseXML Guidelines.

(i) An aseXML Message may contain one or more *BusinessAcceptance/Rejections*.

(j) Participants may include *BusinessReceipts* and *BusinessAcceptance/Rejections* in the same aseXML Message. If *BusinessReceipts* and *BusinessAcceptance/Rejections* are included in the same aseXML message, then that message must be in the format of a Message Acknowledgement (see section 5.4.2 (b)).

(k) AEMO is not required to ensure that items (a), (b), (h), and (j) listed above are not validated by the e-Hub.

3. FIELD FORMAT CONVENTIONS

3.1. Use of standardised format conventions for fields

(a) A Business Document or Business Signal contains a number of fields (items of information/data). A Participant must ensure that each field has a defined format and that the format conforms to the definitions and requirements of this Procedure. The field format indicates the basic contents for the field and imposes length and/or content restrictions.

(b) Note that the format of a field in a Business Document or Business Signal does not describe how the field is implemented in aseXML – the relationship between fields and aseXML Schema elements is defined in the appropriate Business Document/Signal Mapping to aseXML.

3.2. Basic field formats

(a) The field formats in the B2B Procedures are defined in the following table. The value of "x" must be positive and cannot be zero.

Table 3 Basic field formats

	Format	Definition
1.	CHAR(x)	Indicates a field that can only contain alphanumeric characters and must contain exactly "x" characters. The B2B Procedure may add further details to constrain the types of characters allowed. Note that leading and trailing "spaces" are considered significant (i.e. form part of the "x" characters for the field).
2.	VARCHAR(x)	Indicates a character field containing up to "x" characters.
3.	DATE(8)	Indicates a reverse notation date field (i.e. ccyymmdd) with no separators between years, months or days. Years must include the century, whilst months and days must be given as double digits. The "8" indicates that the total field length is always 8 characters. For example: "20030401" represents the 1st April 2003.
4.	DATE(10)	Indicates a reverse notation date field with a hyphen used to separate the years, months and days (i.e. ccyy-mm-dd). Years must include the century whilst months and days must be given as double digits. The "10" indicates that the total field length is always 10 characters. For example: "2003-04-01" represents the 1st April 2003. This is the preferred format for Date fields. This is the format used where the format DATE is used in a B2B Procedure.
5.	DATE(10+hh:mm)	Indicates a 10 character reverse notation date with a time zone indicator, with the "hh" indicating hours and "mm" indicating minutes for the time zone. Note that the "+" (or "-") and ":" characters must be included and so the total field length is 16 characters.

	Format	Definition
6.	DATETIME	Indicates a date time field which is always structured as: ccyy-mm-ddThh:mm:ss.sss+hh:mm This field must include a time zone indicator ("+" or "-" hh:mm). The fractional seconds component (".sss") is optional with any number of digits after the decimal point supported.
7.	TIME	Indicates a time only field and is structured as per the time component of the DATETIME format, ie the DATETIME format left truncated to remove "ccyy-mm-ddT".
8.	NUMERIC(x)	Indicates a positive integer (zero or above) up to "x" significant digits long; any leading zeroes are not significant and hence "050" is equivalent to "50".
9.	NUMERIC(sx)	Indicates a signed integer (positive or negative) up to "x" digits long with an optional leading character for the sign. The sign can be "+" or "-" or a "space" character, with space being interpreted as positive, by convention. If the sign is not provided, the default is positive. The maximum length of the field as a whole is "x"+1 character (reserving space for leading sign).
10.	NUMERIC(x.y)	Indicates a positive number with up to "x" significant characters to the left of the decimal point and "y" decimal places after the decimal point (trailing zeros are optional). In other words, the maximum length of the field as a whole is "x"+"y"+1 characters (the +1 reserving space for the decimal point).
11.	NUMERIC(sx.y)	Indicates a signed number (positive or negative) with up to "x" significant characters to the left of the decimal point and "y" decimal places after the decimal point (trailing zeros are optional). There is a single leading character for the optional sign ("+", "-" or "space"). If the leading sign is "space" or is not provided, the number is interpreted as positive. The maximum length of the field as a whole is "x"+"y"+2 characters (reserving space for the decimal point and leading sign).
12.	ADDRESS	Indicates that a structured or unstructured address needs to be provided. The supplied address will include a number of distinct data elements and must conform to the requirements detailed in Section 3.4~~3.4~~.
13.	ADDRESS (Structured)	Indicates that only a structured ADDRESS will be provided. See Section 3.4~~3.4~~.
14.	ADDRESS (Unstructured)	Indicates that only an unstructured ADDRESS will be provided. See Section 3.4~~3.4~~.
15.	EVENTCODE	Indicates that the field must only be populated with a valid industry agreed code to indicate the reason for "rejecting" a Business Document or indicate explicit acceptance of a Business Document. The explicit events must be detailed in the B2B Procedures whilst the codes are summarised in a National List of Event Codes.
16.	PERSONNAME	Defines a person's legal name as per AS4590-2017- AMD1 2020. See Section 3.5~~3.5~~.
17.	BUSINESSNAME	Defines a business name as per AS4590-2006. This is a 200 character alpha-numeric field.
18.	YESNO	Indicates that a field must contain either "Yes" or "No".

Formatt

Formatt

Formatt

Formatt

	Format	Definition
19.	JURISDICTIONCODE	Indicates that a field must contain a valid jurisdiction code. Valid codes are "ACT", "NSW", "QLD", "SA","VIC", "TAS", "NT", and "WA".
20.	TELEPHONE	Defines a person's Australian telephone service number as per AS4590-1999. See Section 3.6~~3.6~~.
21.	EVENTCONTEXT	Contain the portion of the Transaction or Message to which the Event applies. Format is VARCHAR(80).

Formatt

(b) Participants must use reasonable endeavours to ensure that field formats are fully capitalised (as above). Where a Participant does not adopt this convention, the above conventions still apply. In other words, a field format of "Char(10)" is to be read as equivalent to "CHAR(10)".

3.3. User-defined field formats

(a) Where none of the above basic field formats apply, a user-defined field format can be introduced, provided that it is defined clearly in the relevant B2B Procedure.

(b) Participants must ensure that user-defined field formats are named in a way that avoids confusion with basic field formats (as defined in Section 3.2).

(c) Participants must use reasonable endeavours to ensure that the type identifier is surrounded by double quotation marks to indicate that a user-defined field format has been used. For example, "ddmmccyy" could be used to indicate a forward notation date without separators.

(d) Complex user-defined field formats can be defined as a short form for multiple fields of well-defined types. For example, a field named *SiteAddress* could be defined with a field format of "*AUSTRALIANADDRESS*" provided that the B2B Procedure appropriately defines "AUSTRALIANADDRESS" and what fields it contains. Note that when referring to any given field within a complex format, such as referring to the Postcode within an ADDRESS (Structured) field it should be written as ADDRESS(Structured).*Postcode*; the "full-stop" being used to separate the field names.

3.4. ADDRESS definition

(a) The use of a field format of ADDRESS indicates that a supplied address may be structured or unstructured. Participants must use reasonable endeavours to provide a structured address, unless otherwise required by a B2B Procedure.

(b) A field format of ADDRESS(Structured) indicates those circumstances when a Participant must use a structured address.

(c) A field format of ADDRESS(Unstructured) indicates those circumstance when a Participant must use an unstructured address.

(d) A structured address can comprise all fields (in the following table) except the three *UnstructuredAddress* fields.

(e) An unstructured address comprises *Locality*, *SiteAddressState*, *SiteAddressPostcode* fields, and at least one *UnstructuredAddress* field.

(f) Participants must ensure that the ADDRESS fields *FlatOrUnitNumber*, *FloorOrLevelNumber*, *LocationDescriptor*, and *LotNumber* do not contain the following characters:

` Grave accent

~ Tilde

$ Dollar sign

^ Circumflex

& Ampersand

+ Plus sign

= EQUALS SIGN

| Vertical line

< Less-than sign

> Greater-than sign

/ Forward slash

(g) The following table summarises the information that Participants may provide as part of an ADDRESS:

Table 4 Address field definition

Field name	Field Format	Optional/ Mandatory or Required	Comments
FlatOrUnitType	VARCHAR(4)	R	Code that defines the type of flat or unit as per Australian Standard AS4590-1999. Allowable codes include: APT, CTGE, DUP, FY, F, HSE, KSK, MSNT, MB, OFF, PTHS, RM, SHED, SHOP, SITE, SL, STU, SE, TNHS, U, VLLA, WARD, WE
FlatOrUnitNumber	VARCHAR(7)	R	Defines the flat or unit number as per Australian Standard AS4590-1999.
FloorOrLevelType	VARCHAR(2)	R	Code that defines the floor or level type as per Australian Standard AS4590-1999. Allowable codes include: B, FL, G, LG, M, UG.
FloorOrLevelNumber	VARCHAR(5)	R	Defines the floor or level number as per Australian Standard AS4590-1999.
BuildingOrPropertyName	VARCHAR(30)	R	Defines the building or property name as per Australian Standard AS4590-1999. This field may occur up to two times.
LocationDescriptor	VARCHAR(30)	R	Defines the location descriptor as per Australian Standard AS4590-1999. This is a catch-all field for non-standard address information.
HouseNumber	NUMERIC(5) IN RANGE: 0-99999	R	Defines the house number as per Australian Standard AS4590-1999. The combination of House Number and House Number Suffix may occur up to two times.
HouseNumberSuffix	VARCHAR(1)	R	Defines the house number suffix as per Australian Standard AS4590-1999. The combination of House Number and House Number Suffix may occur up to two times. This field may only contain alphanumeric characters.

Field name	Field Format	Optional/ Mandatory or Required	Comments
LotNumber	VARCHAR(6)	R	Defines the lot number as per Australian Standard AS4590-1999.
StreetName	VARCHAR(30)	R	Defines the street name as per Australian Standard AS4590-1999. The combination of Street Name, Street Type and Street Suffix may occur up to two times. This field may only contain letters, numbers, hyphens ('-') and spaces.
StreetType	VARCHAR(4)	R	A code that defines the street type as allowed for use in MSATS.
StreetSuffix	VARCHAR(2)	R	A code that defines the street suffix as per Australian Standard AS4590-1999. Allowable codes include: CN, E, EX, LR, N, NE, NW, S, SE, SW, UP, W
PostalDeliveryType	VARCHAR(11)	R	A code that defines the postal delivery type as per Australian Standard AS4590-1999. E.g. CARE PO, CMA, CMB, CPA, GPO BOX, LOCKED BAG, MS, PO BOX, PRIVATE BAG, RSD, RMB, RMS
PostalDeliveryNumberPrefix	VARCHAR(3)	R	Defines the postal delivery number prefix as per Australian Standard AS4590-1999. This field may only contain a maximum of 3 capital letters.
PostalDeliveryNumberValue	NUMERIC(5) IN RANGE: 0-99999	R	Defines the postal delivery number value as per Australian Standard AS4590-1999
PostalDeliveryNumberSuffix	VARCHAR(3)	R	Defines the postal delivery number suffix as per Australian Standard AS4590-1999. This field may only contain a maximum of 3 capital letters.
Locality (SiteAddressCity)	VARCHAR(46)	M	Defines the suburb or locality as per Australian Standard AS4590-1999.
SiteAddressState	VARCHAR(3)	M	A code that defines the state as per Australian Standard AS4590-1999. E.g. AAT, ACT, NSW, NT, QLD, SA, TAS, VIC, WA.
SiteAddressPostcode	CHAR(4)	M	Defines the postcode as per Australian Standard AS4590-1999. This field may only contain 4 numbers.
SiteAddressDPID	NUMERIC(8) IN RANGE: 10000000 - 99999999	R	Defines the delivery point identifier as per Australian Standard AS4590-1999

Field name	Field Format	Optional/ Mandatory or Required	Comments
UnstructuredAddress1	VARCHAR(80)	N/M	Mandatory if a structured address is not provided.
UnstructuredAddress2	VARCHAR(80)	O	
UnstructuredAddress3	VARCHAR(80)	O	

3.5. PERSONNAME definition

(a) While the PersonName element can be populated with more than one name, Participants must ensure that only one name is used. The fields in this format are defined below.

Table 5 Person Name field definition

Element	Field Format	Optional/ Mandatory or Required	Description	Allowed Values
PersonNameTitle	VARCHAR(12)	R	Defines a person's title as per Australian Standard AS4590-2017 – AMD1 2020 or Blank. (Where no title is available to populate PersonNameTitle, an empty string should be used to populate it instead)	
PersonNameGiven	VARCHAR(40)	R	Defines a person's given name as per Australian Standard AS4590-2017 – AMD1 2020 or Blank. (Where no given name is available to populate PersonNameGiven, an empty string should be used to populate it instead)	
PersonNameFamily	VARCHAR(40)	M	Defines a person's family name as per Australian Standard AS4590-2017- AMD1 2020	
PersonNameSuffix	VARCHAR(12)	O	Defines a person's name suffix as per Australian Standard AS4590-2017- AMD1 2020	
PersonNameType	VARCHAR(3)	M	Defines the types of people's names as per Australian Standard AS4590-2017 - AMD 1 2020. Implemented as an attribute of PersonName	LGL, MDN, BTH, TRB, PRF, AKA, XFR, STG

3.6. TELEPHONE definition

(a) While more than one PhoneNumber element can be provided, Participants are required to provide the most appropriate number for the business process. The fields in this format are defined below.

Table 6 Telephone field definition

Element	Field Format	Optional/ Mandatory or Required	Description	Allowed Values
Prefix	VARCHAR(4)	M	Defines Australian telephone number prefix as per Australian Standard AS4590-1999	
Number	VARCHAR(15)	M	Defines Australian telephone number as per Australian Standard AS4590-1999	
ServiceComment	VARCHAR(40)	R	Telephone service comment.	"Home" "Business"
ServiceType	VARCHAR(12)	M	Used to describe the type of telephone service. Implemented as an attribute of AustralianPhoneNumber.	"Fixed Voice" "Mobile Voice" "Fax" "Pager"

3.7. Fields that contain codes or enumerated lists

(a) Where the contents of a field is a list of possible values (i.e. an enumerated list) or a "code", Participants must use reasonable endeavours to ensure that the entries are written in full using title case and with single spaces allowed between words. For example, the *MovementType* field in a CustomerDetailsNotification may contain only one of the following alternatives;

- Site Vacant
- Update
- Reconciliation

(b) Note that when defining an enumerated list, one of the basic field formats described in Section 3.2 should be used unless a new field format is required. Where a basic field format is used, the length of the field should correspond with the maximum anticipated content for the field. For example, *MovementType* is defined as a VARCHAR(14) with contents limited to an enumerated list defined in the B2B Procedure.

(c) Shorthand codes and abbreviations should be avoided unless there is a compelling reason (e.g. where a series of industry agreed coded values already exists).

4. PAYLOAD DEFINITIONS

4.1. CSV Notification Detail

(a) A CSV Notification Detail contains a number of fields (items of information). A Participant must ensure that each field has a defined format and that the format conforms to the definition and requirements of this Procedure.

(b) The CSV Notification Detail contains a description of the B2B Procedure One Way Notification Process *CSVNotificationDetail* message payload.

(c) The format of the Business Document or Business Signal does not describe how the field is implemented in aseXML – the relationship between fields and aseXML Schema elements is defined in the B2B Procedure One Way Notification Process.

(d) The CSV Notification Detail payload name is defined as *CSVNotificationDetail*.

4.1.1. Data Rules

(a) The *CSVNotificationDetail* content will have three types of records C, I and D. Each record starts with one of the three characters C, I or D. C is for Comment, I is for Information and D is for Data.

(b) Fields must not include leading or trailing spaces.

(c) A comma is required between all fields, even if the field is Null.

(d) Commas are not permitted as valid characters in any data field.

(e) The values in fields are not case sensitive.

(f) All record lines must end with a carriage return and line feed (CRLF).

(g) The format of the *CSVNotificationDetail* payload is defined in the following table:

Table 7 CSV field definition

Sequence	RecordIndicator	Definition
1	C	HEADER RECORD The HeaderRecord only contains header information. Example *C,system,MessageType,from,to,CreateDate,CreateTime* system: Always e-Hub MessageType: The message type described by the procedure from: Creating Participant market identifier to: Receiving Participant market identifier CreateDate: Date file created - Format CCYY/MM/DD CreateTime: Time file created - Format HH:MM:SS End of record is signalled with a carriage return line feed (CRLF) Example (of network tariff notification) C,e-Hub,OneWayNotification,PART1,PART2,2009/08/25,15:25:08
2	I	INFORMATION RECORD The Information Record (I) lists the column headings for the data (D) records below. The number of data columns will vary depending on the message type and payload requirements. Column1, column2, column3 and column4 always contain the RecordIndicator, RECORDNUMBER, MESSAGENAME and VERSION respectively. Additional columns are specified in the B2B Procedure One Way Notification Process. End of record is signalled with a carriage return line feed (CRLF). Example (of network tariff notification) I,RECORDNUMBER,MESSAGENAME,VERSION,NMI,NMICHECKSUM,METERSERIALNUMBER,NMISUFFIX,NTPROPOSEDDATE,NOTICEENDDATE,PROPOSEDNTC,REASONFORCHANGE

Sequence	RecordIndicator	Definition
3	D	DATA RECORD The DataRecord must always commence with a "D" followed by the content. The content (message) payload details are specified in the B2B Procedure One Way Notification Process. This will vary depending on the message type and pay load requirements End of record is signalled with a carriage return line feed (CRLF) Example (of network tariff notification) D,1,NTN,2,1234567890,1,87654,E1,20171201,20171220,B101,DNSP Review
4	C	FOOTER RECORD The footer record only contains three values; C,ENDOFREPORT,*RecordCount* RecordCount - The RecordCount is a count of the data (D) records, excludes C and I records. Example C,ENDOFREPORT,5

5. TRANSACTION DELIVERY REQUIREMENTS

5.1. Delivery Mechanisms

(a) All Participants must ensure that all B2B Transactions and Acknowledgements are sent via the e-Hub in accordance with the requirements of this Procedure, subject to contingency provisions set out in section 9 of this Procedure.

(b) The National B2B Infrastructure used to deliver B2B Transactions supports "once and once only delivery". Subject to 5.11~~5.11~~:

Formatt

(i) Participants and AEMO must not re-use *ase:MessageID* where they have received a Message Acknowledgement from the Recipient for that Message.

(ii) Participants and AEMO must not re-use *ase:TransactionID* where they have received a Transaction Acknowledgement from the Recipient for that Transaction.

(c) Participants acknowledge and accept that Transactions and Acknowledgements may be delivered to a Participant out of sequence. Participant systems must not assume a given delivery sequence. Refer to the B2B Procedures for any specific out of sequence handling requirements.

5.2. Participant Addressing

(a) Participants and AEMO must issue B2B Messages using valid ParticipantIDs (as published by AEMO) in the *ase:To* and *ase:From* fields of the Message header.

(b) Participants and AEMO must ensure that ParticipantIDs used relate to the appropriate role for the NMI for the B2B Message (e.g. a Retailer ParticipantID must not be used in the *ase:From* field for an ase:MeterDataNotification).

(c) Participants and AEMO must ensure that the Participant IDs used in Request and Response Transactions match. That is, the *ase:From* Participant ID in the Request (e.g. ase:ServiceOrderRequest) must be the same as the *ase:To* ParticipantID in the Response (e.g. ase:ServiceOrderResponse). Participants must ensure that ParticipantIDs used in any Acknowledgements match the Transaction to which they relate.

5.3. Message Equivalents

(a) The table below provides the mapping of equivalent messages and transactions between protocols.

Table 8 Message terminology by protocol

Message	FTP Term	Webservice Term	Description
Hub Acknowledgement	.ac1	HTTP Response with Hub MessageAcknowledgement payload MACK (positive/negative)	Hub response on receipt of a message.
Message Acknowledgement	.ack MsgAck (positive) MsgNack (negative)	HTTP Response with MessageAckowledgement payload MACK (positive/negative)	Recipient/Notified Party acknowledges receipt of the message from the Initiator/e-Hub.
Transaction Acknowledgement	TranAck	TACK (positive/negative)	Recipient/Notified Party provides a business/logical acceptance or rejection of the contents of the transaction.

5.4. Overview of MSATS B2B Handler Functionality (FTP)

5.4.1. Functional Overview

(a) The MSATS B2B Handler provides the functionality of a "mailbox" service distributing B2B files directly between valid Market Participants.

(b) AEMO must ensure that the functionality of the MSATS B2B Handler includes:

(i) The ability for B2B files to be sent directly to Participant directories ("Inbox") as specified.

(ii) Header and schema validation of files.

(iii) Production of a negative Hub Acknowledgement in the case of B2B Message failure.

(iv) A subdirectory that contains flow control files (.stp files) identifying Participants who are Stopped.

(v) (By reading this subdirectory before lodging a new file and not lodging files for stopped Participants, an Initiator can avoid receiving a negative Acknowledgement).

If a Participant that a B2B Transaction is being sent to has reached its file limit, the B2B file transfer fails and a negative hub Acknowledgement is sent to the B2B Initiator. (This limit is configurable per Participant and is independent of MSATS file limits.)

(vi) Support for specific B2B Transaction types.

(vii) Logging of MSATS B2B Handler activity in an activity log (not the MSATS database).

(viii) Creation of a B2B e-Hub Acknowledgement file with a different extension (.ac1) is created to signify the successful transfer of a B2B Transaction to the intended Recipient. That is, the .ac1 file contains a positive *ase:MessageAcknowledgement*. The intended Recipient signifies successful reception of the B2B file by creation of an Acknowledgement file with an .ack extension, which is then copied to the B2B Initiator.

(c) Where a Recipient's Inbox contains an invalid *ase:MessageAcknowledgement(s)* or invalid standalone *ase:Event(s)*, the Recipient can still initiate Transactions by lodging ".zip" files into their Inbox (refer 5.4.2). These .zip files will be delivered. The Recipient will also continue to receive Transactions, but the MSATS B2B Handler will not deliver the corresponding *ase:MessageAcknowledgements*. This will eventually result in the flow control limit being exceeded if the error(s) is not resolved.

5.4.2. Compression

(a) Participants must send all aseXML Messages containing aseXML Transactions as compressed files with a ".zip" extension. Participants may send Transaction_Acknowledgements either in the same format as aseXML Transactions or, if sent at the same time as corresponding Message Acknowledgements, in the format described in (b) below. Standalone Transaction Acknowledgements must be sent as compressed files.

(b) Participants must send all Message Acknowledgements as an .ack or .ac1 files that are not compressed.

(c) Participants must send all standalone ase:Events as .ack files that are not compressed.

(d) Participants must ensure that any zip file sent is not password protected.

(e) Participants must ensure that the path name is not included in the zip file.

(f) Participants must ensure that the zip file only contains one aseXML file.

(g) Participants must ensure that any embedded file has the same name as the zip file, with an ".xml" extension.

(h) Participants acknowledge and accept that items (e), (f), and (g) listed above are not validated by the MSATS B2B Handler.

5.4.3. Validation

5.4.3.1. Initial Transfer of B2B file

(a) The initiating step of the B2B file handling protocol is largely concerned with the information that is stored in the header of a B2B Transaction file. The header fields are used as the addressing data that determines whom the B2B Transaction goes to. AEMO must ensure that the e-Hub does not process the contents of a B2B Transaction file (excepting the header). The file, as delivered to the e-Hub, is passed to the receiver unchanged.

(b) The following validations apply to incoming B2B files to the e-Hub. When one of these validations is not satisfied either the file is ignored or a negative acknowledgment is created.

5.4.3.1.1. Validate Sending Participant

(a) Participants acknowledge and accept that the e-Hub will return an error if a Participant sends a file using a protocol other than what they have nominated and AEMO has configured in the e-Hub.

(b) Where Participants have elected to use the FTP protocol, Participants must ensure that their Inboxes have been configured accordingly.

(c) Participants acknowledge and accept that files submitted by Participants who are not configured for B2B operation shall be ignored without error by the e-Hub.

5.4.3.1.2. Validate Transaction Group in File Name

(a) Participants must ensure that they use a Transaction Group in their filename which has been configured for B2B operation. Participants acknowledge and accept that the e-Hub shall only handle incoming files from Participants that use a Transaction Group in their filename that has been configured for B2B operation. Files submitted with Transaction Groups other than those configured for B2B operation shall be ignored without error by the e-Hub. Files submitted with Transaction Groups that are used by other systems, such as MSATS, where a common Inbox is used, may be processed by those systems and reported by those systems as an error.

5.4.3.1.3. Validate zip file

(a) AEMO must ensure that the MSATS B2B Handler sends a standalone *ase:Event* (with the *EventCode* = 5) to the Sender in response to the receipt of a corrupted zip file.

5.4.3.1.4. Validate XML payload

(a) AEMO must ensure that if the XML payload is not well formed or schema invalid the e-Hub produces a negative Acknowledgement. AEMO must ensure that the e-Hub also checks that the payload is less than limits prescribed in clause 5.8(a).

5.4.3.1.5. Validate Participant Id in the <From> Field

(a) Each Participant must ensure that the Participant Id in the <From> field of the header is the same as the owner of the Inbox. Participants acknowledge and accept that a failure to comply with this clause will result in the e-Hub producing a negative Acknowledgment.

5.4.3.1.6. Validate Participant Id in the <To> Field

(a) Each Participant must ensure that the Participant Id in the <To> field is a Participant Id which has been configured for B2B operation. Participants acknowledge and accept that a failure to comply with this clause will result in the e-Hub producing a negative Acknowledgment.

5.4.3.1.7. Recipient file limit reached

(a) If the Recipient of the message has exceeded its file limit and a Stop File has been created, AEMO must ensure that the e-Hub produces a negative Acknowledgement having an *ase:Event* (with the *EventCode* = 111), which is returned to the sender.

5.4.3.2. Scenarios for Transfer of Recipient Acknowledgement

(a) Participants and the e-Hub may produce Acknowledgement files in two formats: one containing an aseXML header with Transaction and Message Acknowledgements, or alternatively an event description. This section describes scenarios for both types of Acknowledgement file.

5.4.3.2.1. Acknowledgement with aseXML header

(a) Participants and the e-Hub must produce Acknowledgements with an aseXML header that contain (among other things) information about whom the Acknowledgement is going to and from whom the Acknowledgement is coming. AEMO must ensure that the e-Hub then validates the <From> and <To> fields in the Acknowledgment against the <To> and <From> fields in the file that the Acknowledgement relates to.

(b) Participants acknowledge and accept that where these validations fail, the .ack file is not processed and the acknowledging Participant shall be skipped for further Acknowledgement processing. If the file cannot be opened or parsed for any reason, then these validations shall fail.

5.4.3.2.2. Acknowledgement With Event Info

(a) Participants acknowledge and accept that Acknowledgements do not have aseXML header information in them. The e-Hub shall use information in the header of the file to determine which Participant to send the Recipient Acknowledgement to. It then reverses the <To> and <From> fields to determine whom the 2nd level Acknowledgement needs to go to. If the file cannot be opened or parsed for any reason then this mechanism will fail, the .ack file will not be delivered and the acknowledging Participant is skipped for further Acknowledgement processing.

5.4.3.3. Other validation details

(a) If a Message is schema invalid, Participants must ensure that either an aseXML *ase:MessageAcknowledgement* or standalone *ase:Event* is returned to the Initiator, as described in the aseXML Guidelines.

5.4.4. Flow Control Management

(a) The SMP Hub flow control management functions in parallel with the MSATS B2B Handler flow control management. The flow control configuration will apply to both the SMP Hub and the MSATS B2B Handler. i.e. a Stop File will prevent all further messages and files being delivered from the e-Hub via FTP and webservice (as opted in to) until the Stop File is removed.

(b) AEMO must ensure that the MSATS B2B Handler supports the timely delivery of B2B Transactions as detailed in clause 5.9 (d).

(c) AEMO must ensure that the MSATS B2B Handler provides the following flow control management functionality. This functionality is a protection mechanism against file overloading of a Recipient's Outbox.

(d) AEMO must ensure that the MSATS B2B Handler's flow control management functionality is based on the use of "flow control" files. Two types of flow control files must be used. The first flow control file is named "B2Bholdinp.stp" and is located in the Recipient's Outbox. The second flow control file contains the name of the Recipient who is at the warning limit (ParticipantID_B2Bholdinp.stp) and must be located in a special directory "stopbox" located at the same level as Inbox and Outbox and repeated for each Participant.

(e) AEMO must ensure that when the number of unacknowledged B2B .zip files in a Participant Outbox exceeds a configurable Warning level, the MSATS B2B Handler creates a flow control file of the form ParticipantID_B2Bholdinp.stp in the stopbox directory for each B2B Participant. On a subsequent flow control file processing cycle, when the number of unacknowledged B2B .zip files in a Participants Outbox exceeds a configurable High level set by AEMO for that Participant, AEMO must ensure that the MSATS B2B Handler writes a flow control file (B2Bholdinp.stp) to that Participant's Outbox. AEMO must ensure that when the number of files subsequently drops below a certain Lower level, the MSATS B2B Handler removes the B2Bholdinp.stp flow control file and Message file movements recommence. On a subsequent cycle of the flow control file processing, the MSATS B2B Handler must remove the ParticipantID_B2Bholdinp.stp from all Participant stopbox directories if the number of B2B outstanding files is below the lower level. AEMO configures the Warning, High and Lower flow control file levels for each Participant.

(f) The ParticipantID_B2Bholdinp.stp file acts as a flow control mechanism so that an Initiator can check before lodging if a Participant is at the Warning flow control file limit. The reason that it must be created in a cycle before the B2Bholdinp.stp file is so a race condition is avoided. Similarly, it must be removed after the B2Bholdinp.stp file. The flow control file processing runs as a separate configurable process, with a frequent cycle. It is important to only do the two types of flow control files in separate cycles to ensure that the stopbox can stop the flow before the Participant is totally stopped. All of this requires careful tuning by both the MSATS B2B Handler and Participant Gateways.

(g) The B2Bholdinp.stp file acts to signal to the Participant that further B2B file movements to their Outbox has ceased, as a flag to the MSATS B2B Handler to deny further file movements to this Outbox and create rejection acknowledgment to the B2B Message file Initiator.

5.4.5. MSATS B2B Handler File Naming Convention

(a) Participants and AEMO must use the following file naming convention when using the MSATS B2B Handler:

(i) The MSATS B2B Handler file naming convention is defined by the following regular expression:

[0-9_a-z]{1,4}[h|m|l][0-9_a-z]{1,30}[.](tmp|zip|ack|ac1)

where:

The first four (4) characters represent the Transaction Group.

The fifth character represents the Priority, h = High, m = Medium, l = Low

The remaining 30 characters represent the unique identifier of the Message file.

(ii) The Recipients Outbox is a shared namespace for all Initiator's files. To avoid name collisions Participants must ensure that the Participant Id of the sending Participant is contained at the start of the remaining 30 characters used for Message uniqueness.

(iii) Participants must use their own and appropriate ParticipantID in the 30 character unique identifier.

(iv) For Example: sordmagle123456789.zip or
custmagle_____312301274_____batch.zip

(v) Participants must only use lowercase characters in file names. Participants acknowledge and accept that the MSATS B2B Handler recognises and processes incoming ".zip" files by their four character Transaction group. An invalid Transaction Group file name prefix will cause the ".zip" file to be ignored.

(vi) Participants must ensure that the file names are unique. A Participant may only reuse a file name if the original file was not acknowledged by the Recipient.

5.5. Overview of SMP Hub Functionality (Webservices)

5.5.1. Functional Overview

(a) The SMP Hub is opt-in functionality; the MSATS B2B Hander is the default preference for all participants. Refer to section 6.2 regarding opting in.

(b) The SMP Hub adopts Webservices (RESTful APIs).

(c) Business transactions will be sent as an aseXML document carried as a payload inside the Webservice message and transmitted over HTTPS.

(d) The SMP Hub implements an Asynchronous message exchange pattern (messages delivered via non-blocking thread(s) i.e. multi legged: Initiator → SMP Hub → Recipient), and follows a Push-Push design pattern.

(d) AEMO must ensure that the functionality of the SMP Hub includes:

(i) The ability for B2B Messages to be sent to Participants.

(ii) Header and schema validation of Messages.

(iii) Production of a negative Hub Acknowledgement in the case of B2B Message failure.

(iv) An available webservice for Participants to call that list flow control files (.stp files) identifying Participants who are Stopped.

(v) An available (opt in) webservice initiated by the SMP Hub to notify the Participants of the existence of flow control files identifying Participants that are Stopped.

(vi) An available (opt in) webservice initiated by the SMP Hub to notify the Participants of the removal of flow control files identifying Participants that are no longer Stopped.

(vii) (By identifying Stopped Participants using the available websevice methods in (iv) and (v) before lodging a new file and not lodging files for stopped Participants, an Initiator can avoid receiving a negative Acknowledgement).

If a Participant that a B2B Transaction is being sent to has reached its limit (as prescribed in section 6.5.6), the transfer fails and a negative Hub Acknowledgement is sent to the B2B Initiator.

(viii) A B2B e-Hub Acknowledgement is created to signify that the SMP Hub has successfully validated the incoming message and is undertaking the delivery of a B2B message to the intended Recipient. That is, the SMP Hub Acknowledgement contains a positive *ase:MessageAcknowledgement*.

(ix) The intended Recipient signifies successful reception of the B2B Message by the creation of an Acknowledgement. The SMP Hub will then route the Recipient's Acknowledgement back to the Initiator.

(x) The SMP Hub will archive all incoming and outgoing messages into existing Participant archive folders, with zip file names generated as per 5.4.5(a)(i).

5.5.2. Compression

(a) The SMP Hub does not support compression of Messages.

5.5.3. Access Methods

(a) The e-Hub will be configured to accept messages from participants connected to the Internet and/or MarketNet.

(b) These existing network options in conjunction with the proposed protocols provide appropriate levels of security and reliability for B2B transactions, including faults.

(c) Participants using only FTP will only be able to connect via MarketNet, while those using Webservices can connect via either MarketNet or the Internet.

5.5.4. Webservices Header

(a) Webservices will utilise the aseXML header for the purpose of processing and routing messages.

5.5.5. Invoking a Webservices call

(a) The Initiator will invoke a webservice call using the URL of e-Hub. The URL must be provided by AEMO to Participants that have opted for webservices.

(b) The SMP Hub will invoke a webservice call using the URL of the Recipient. The URL must be provided by the Recipient Participant where they have opted in for webservices.

(c) Details on SMP Hub URL management is contained in the SMP Technical Guide.

5.5.6. Webservices Response codes

(a) A standard HTTP successful response code (e.g. 2xx) will be provided with a *ase:MessageAcknowledgement* when a webservice is successfully invoked (passes all validations).

(b) A standard HTTP successful response code with a negative *ase:MessageAcknowledgement* will be provided if a webservice call is successfully invoked but fails non-technical validation.

(c) An appropriate HTTP response code (e.g. 4xx) will be provided when a webservice fails a technical validation.

(d) An appropriate HTTP response code will be provided by the e-Hub to the Initiator when a webservice fails due to a timeout or other error by a Recipient. Timeout reasons are listed but not limited to the below:

(i) Connection timeout – Unable to establish connection.

(e) Detailed HTTP response codes will be provided by AEMO and will be detailed in the SMP Technical Guide.

(f) Any connectivity or timeout issues between the Initiator and the e-Hub are the responsibility of the Initiator to capture and manage.

5.5.7. Security

(a) Participants should be aware of security risks and institute countermeasures appropriate to the value of the asset(s) that might be placed at risk.

(b) AEMO will utilise the following security features for the inbound messages:

Profile	Security
Transport	HTTPS
Authentication Method	API-Key, plus SSL Client Authentication
Firewall Rules	White List of AEMO Participants' IP addresses

(c) AEMO will support the following security capabilities when consuming the Participant's webservices, as required by the Participants:

Profile	Option 1	Option 2
Transport	HTTPS	HTTPS
Authentication Method	SSL Client Authentication	API-Key, plus SSL Client Authentication

5.5.7.1. Encryption

(a) The SMP Hub will employ transport layer encryption via SSL to protect the data in transit. The API keys used will be 128 bit X509v3 and will be provided by the SMP Hub administrator. In the SMP Hub the transport layer will be encrypted, but the message payload will not be encrypted.

(b) Details of encryption will be provided in the SMP Technical Guide.

5.5.7.2. API Keys

(a) The SMP Hub will require some Public Key Infrastructure. AEMO will provide the API key for the webservices that will be available in the e-Hub. The API key will be specific to a webservice and Participant.

(b) The SMP Hub will utilise the API keys to perform additional authentication and authorising the API request. If the SMP Hub is the consumer of a Participant's webservice, the SMP Hub can provide the Participant's API key if the Participant's systems can support API key management.

(c) Management of API Keys such as key provisioning replacing keys and key expiry will be defined in the SMP Technical Guide.

5.5.8. Validation

(a) The validation detailed in section 5.4.3 also applies for messages sent via the SMP Hub, with the exception of sections 5.4.3.1.2 and 5.4.3.1.3, which are not applicable. Additional validation specific to webservices and the SMP Hub are detailed in this section.

(b) Security validations as described in section 5.5.7 as well as message throttling logic described in section 5.5.10 also take place for any inbound messages received by the SMP Hub. Any failure as a result of these validations will result in the appropriate HTTP response code.

5.5.9. Flow Control Management

(a) The SMP Hub flow control management functions in parallel with the MSATS B2B Handler flow control management. The flow control configuration will apply to both the SMP Hub and the MSATS B2B Handler. i.e. a Stop File will prevent all further messages and files being delivered from the e-Hub via FTP and webservice (as opted in to) until the Stop File is removed.

(b) AEMO must ensure that the SMP Hub supports the timely delivery of B2B Transactions as detailed in clause 5.9 (e).

(c) AEMO must ensure that the SMP Hub provides flow control management functionality. This functionality is a protection mechanism against message overloading of a Recipient's webservice queues.

(d) AEMO must ensure that the SMP Hub's flow control management functionality is based on the use of "flow control" files. Two types of flow control files must be used. The first flow control file is named "B2Bholdinp.stp" and the second flow control file contains the name of the Recipient who is at the warning limit "ParticipantID_B2Bholdinp.stp". This is to ensure interoperability with the MSATS B2B Handler.

(e) The SMP Hub will also provide the functionality for Participants to obtain webservice alerts for Stop Files (when they are added and removed) for Participants using webservices.

(f) AEMO must ensure that when the number of unacknowledged B2B messages in a Participant webservices queue exceeds a configurable warning level (Water Mark - Warn), the SMP Hub issues an alert to the Participants via the webservice invocation.

(g) The SMP Hub will not reject the incoming B2B Messages when a Stop File exists for the Recipient. The SMP Hub will attempt to deliver the messages to the Recipient for a configurable period of time.

(h) AEMO must ensure that when the number of unacknowledged B2B Messages in a Participant webservice queue exceeds a configurable level (Water Mark - High), the SMP Hub will issue an alert to Participants via the webservice invocation.

(i) If the SMP Hub receives any webservice invocation for a Participant that has a Stop File, the SMP Hub will send a negative *ase:MessageAcknowledgement* on the return of the webservice call.

(j) The SMP Hub will provide a webservice that Participants can invoke to retrieve the list of Stop Files in their Stopbox.

(k) Information about current Stop Files for any Participant will be available on the B2B Browser Application.

(l) When Stopped, the SMP Hub will process messages in the Participant's webservice queue according to message priority. Note: this is the only instance where message priority is used for processing messages in webservices; due to the push-push message pattern, a first-in-first-out processing will occur at all other times.

(m) When sufficient messages have been processed from the Participant's webservice queue and the number of messages in the queue falls below a configurable low message volume (Water Mark – Low), the Hub will remove the Stop Files from the Stopbox.

(n) The SMP Hub will invoke a webservice call to notify Participants of the removal of Stop Files.

5.5.10. Message Throttling

(a) The active thread pool (open threads) threshold will be set at the SMP Hub level.

(b) If the number of open threads exceeds the SMP Hub threshold, there will be a minor delay (seconds) in addressing the waiting threads.

(c) AEMO will inform all participants of throttling limit increases/decreases through industry agreed communications methods.

5.5.10.1. Inbound

(a) The SMP Hub will throttle the incoming messages by limiting the number of Messages a Participant can send using a webservice i.e. number of Messages per minute for a webservice.

(b) The SMP Hub will send an appropriate HTTP response when the number of Messages per minute exceeds the threshold.

5.5.10.2. Outbound

(a) If a Participant requires SMP Hub to throttle to a lower number (send fewer messages per minute instead of default), the SMP Hub can send the outbound messages based on the following parameters:

(i) how often the messages should be sent (e.g. messages to be sent every 5 min);

(ii) the number of messages to be sent in each batch (e.g.100 messages in each batch); and

(iii) the Messages to be sent sequentially or concurrently.

(b) The SMP Hub will send messages to the Recipient at the same rate as it is received from the Initiator (Default).

(c) If the SMP Hub receives a 'throttling error' from a Participant, the requests will be queued for re-submission. The messageIDs and the transactionIDs will not be changed during re-submission.

5.6. Authentication and Non-repudiation

(a) Non-repudiation seeks to ensure tamper proof delivery and authentication of the Initiator. AEMO must ensure that this is supported by the MSATS B2B Handler as follows:

(i) Network isolation is provided by the use of the National B2B Infrastructure that is a private, isolated and secure network. The National B2B Infrastructure is only capable of being accessed by authorised Participants.

(ii) Participant Network Authentication whereby, once connected to the National B2B Infrastructure, a Participant may only gain access to the:

(A) MSATS B2B Handler via userID and password authentication; and

(B) SMP Hub via SSL authentication (also see 5.5.7).

(iii) Manual non-repudiation is also supported by the persistence of every B2B Transaction that is processed by the e-Hub.

(iv) All aseXML Transactions and Acknowledgements are delivered by the e-Hub with no modification.

5.7. Priority of aseXML Messages

(a) Unless otherwise specified in a B2B Procedure, Participants must ensure that:

(i) all fully tagged aseXML Transactions are sent as Medium Priority aseXML documents; and

(ii) all aseXML wrapped CSV Transactions are sent as Low Priority aseXML documents.

(b) Participants must ensure that the ase:*MessageAcknowledgements* and ase:*TransactionAcknowledgements* are the same priority as the Initiating Message.

(c) Participants acknowledge and accept that items 5.7(a) and 5.7(b) listed above shall not be validated by the MSATS B2B Handler.

5.8. Size of aseXML Messages

(a) Participants must ensure that the Message size and number of transactions per Message do not exceed the following values for each transaction group:

Transaction Group	Message size limit	Limit on number of transactions in Message
CUST	1 MB	N/A
MRSR	1 MB	N/A
MTRD	10 MB	1000
NPNX	1 MB	N/A
OWNP	1 MB	N/A
OWNX	1 MB	N/A
SITE	1 MB	N/A
SORD	1 MB	N/A

(b) Participants acknowledge and accept that the e-Hub will reject Messages that exceed the limits prescribed in clause 5.8(a). AEMO must use reasonable endeavours to ensure that the e-Hub rejects Messages which exceed the limits prescribed in clause 5.8(a) with an Event Code (*ase:Code*) of "6", i.e. "Message too big".

5.9. Timing Requirements

(a) With the exception of periods covered by any industry agreed outage period, Participants must use reasonable endeavours to adhere to the Timing Requirements stated in this section and as prescribed by the relevant B2B Procedure.

(b) Timing requirements for the delivery of aseXML Transactions and Acknowledgements via the National B2B Infrastructure are summarised below in Figure 1 and the associated Table 9 and Table 10. The figure illustrates the three Acknowledgement cycles. The batch or polling cycle of the hub is also indicated, but the equivalent batch and processing timings of Participants are not illustrated.

Figure 1 Summary of Timing Requirements

(c) The following maximum Timing Requirements apply to the Acknowledgement cycles. Participants and AEMO must meet these Timing Requirements for a minimum of 95% of Transactions during a rolling 5

business day period, or the industry Timing Requirements otherwise agreed. This requirement is based on an agreed industry loading scenario which AEMO has published to industry.

(d) The table below applies to participants interacting using FTP or a combination of FTP and webservices (FTP to FTP, Webservices to FTP, or FTP to Webservices):

Table 9 Timing Requirements

Cycle	Low Priority Transactions	Medium Priority Transactions	High Priority Transactions	Responsibility
Hub Transmission Time	30 minutes	15 minutes	5 minutes	AEMO
MsgAck Cycle Time	240 minutes	60 minutes	30 minutes	Participant
TranAck Cycle Time	By end of next business day	By end of next business day	60 minutes	Participant

(i) The Hub Transmission Time is the time from a Participant placing a file in their Inbox to the Handler moving the file to the other Participant's Outbox.

(ii) This time can be measured by the time taken for an .ac1 to be placed in the Sender's Outbox in the circumstances where an .ac1 file is produced.

(iii) For a valid Message, the MsgAck Cycle Time includes two Hub Transmission Times.

(iv) For a valid Message, the TranAck Cycle Time includes two Hub Transmission Times.

(e) The minimum timeframes are based on the table in clause 5.9 (d) unless both the Initiator and Recipient/Notified Party are on Webservices, then the Webservices table below applies:

Table 10 Timing Requirements (webservices only)

Cycle	Low Priority Transactions	Medium Priority Transactions	High Priority Transactions	Responsibility
Hub Transmission Time	5 seconds	5 seconds	5 seconds	AEMO
MsgAck Cycle Time	10 seconds	10 seconds	10 seconds	Participant
TranAck Cycle Time	By end of next business day	By end of next business day	60 minutes	Participant

(f) A Business Document is deemed to have been received by a Participant on the date and time set out in the *ase:MessageDate* contained in the corresponding .ac1 file. A Participant's obligations under the relevant B2B Procedure are deemed to commence at that time (the *ase:MessageDate* contained in the corresponding .ac1 file).

(g) A Participant must ensure that an *ase:MessageAcknowledgement* for a Request is not sent in the same file as the Response to the Request.

(h) A Participant must ensure that an *ase:TransactionAcknowledgement* for a Request is not sent in the same file as the Response to the Request.

5.10. Transaction Logging

(a) The e-Hub provides a complete audit trail of the delivery and Acknowledgement of a B2B Message/Message Acknowledgement cycle to support the non-repudiation requirements. The e-Hub stores the following information to support the Logging requirements.

(b) AEMO must ensure that the e-Hub stores the following information to support the Logging requirements.

Table 11 Data Logging Requirements

Data Being Logged	Source of Data
User identification fields	FROM and TO fields in aseXML header
Timing fields	Date Time Created Date Time acknowledged
Incoming Message ID	XML header
Hub Acknowledgement Receipt ID	Allocated by MSATS and returned to Participant in Acknowledgement – unique B2B receipt ID – this is only for .ac1 or negative .ack files generated by B2B Handler.
Outgoing Receipt ID	Extracted from Recipient Acknowledgement from Participant – returned to Initiator
Transaction Group	XML header
Date/time of delivery of Message	.ac1 *ase:MessageDate*
Message priority	FTP: File name Webservices: XML header

5.11. Handling of duplicate or resent Transactions and Messages

(a) AEMO and Participants must handle any duplicate Transactions and Messages in accordance with the aseXML Guidelines.

(b) With the exception of an ase:ServiceOrderResponse, a Participant may correct a Business Document and resend it using the same data provided that:

- (i) The original Business Document was rejected via a negative *ase:MessageAcknowledgement* or negative *ase:TransactionAcknowledgement*; and
- (ii) A new *ase:MessageID* and *ase:TransactionID* is used.
- (iii) A Participant must only resend an *ase:MessageID* if the original was rejected by the MSATS B2B Handler; that is, if an .ac1 file is received for a Message, a new *ase:MessageID* must be used for resent Messages.

(c) A Participant must ensure that if they receive a negative Business Receipt and/or Business Rejection that they undertake the action specified in the table below if the rejection relates to an individual error situation. Where multiple errors occur due to system malfunctions, the affected Participants must contact each other and agree a resolution of the situation.

(d) Action Matrix following negative Business Receipts and Business Rejections

Table 12 Transaction Action Matrix for negative acknowledgements

Transaction	Event	Reason/Outcome	Action[1]
Any	Hub sends *ase:Event*	Reason: Recipient is Stopped	Sender waits until the Recipient is no longer stopped and then resends original Message, or may issue a new Message.

[1] Allocation of a new Service Order Number and method of confirming acceptance is a business process decision.

Transaction	Event	Reason/Outcome	Action[1]
		Any reason other than Recipient is Stopped.	Sender corrects and resends as a new Message. Sender may allocate a new *RequestID* or *ServiceOrderNumber*, if applicable.
	Hub sends standalone *ase:Event*	File transport error prevents uncompression of Message	Sender resends original Message, or may issue a new Message.
	Recipient sends standalone *ase:Event*	File transport error prevents uncompression of Message.	If due to transportation error from Hub to Receiver (i.e. when the Receiver copies the file locally from the Hub Outbox) – then Hub will deliver the event to the sender of the original Message. The hub will also clean up the zip file in the Recipient's Outbox. If due to a hub copy failure from Inbox to Outbox (extremely unlikely due to integrity checks performed by the hub) – then the *ase:Event* will be treated as a 'bad acknowledgment" requiring manual intervention.
Business Document	Recipient sends negative *BusinessReceipt* (*ase:MessageAcknowledgement*)	Any reason.	Sender corrects and resends as a new Message. If a Request Transaction, the Sender may allocate a new *RequestID* or *RetServiceOrder*, if applicable.
	Recipient sends negative *BusinessAcceptance/Rejection* (*ase:TransactionAcknowledgement*)	Sender accepts the reason for the Business Rejection.	Sender corrects and resends as a new Message. If a Request Transaction, the Sender must allocate a new *RequestID* or *RetServiceOrder*.
		Recipient admits error (ie. incorrect rejection).	The Sender resends as a new Message. If a Request Transaction, the Sender must allocate a new *RequestID* or *RetServiceOrder*.
ServiceOrderResponse	Initiator sends negative *BusinessAcceptance/Rejection* (*ase:TransactionAcknowledgement*)	Recipient accepts the reason for the Business Rejection	The Recipient and the Initiator negotiate a resolution of the reason for the rejection, with the agreed resolution being reflected in each party's systems.
		Initiator admits error (ie. incorrect rejection).	

5.12. Timestamps

(a) Participants must ensure that:

(i) the "+hh:mm" component of *ase:MessageDateTime* = +10.00; and

(ii) the "+hh:mm" component of *ase:TransactionDateTime* = +10.00.

(b) The time zone selected for date/time stamps within Transactions will be at the discretion of the Participant sending the Transaction. The sending Participant must ensure that the combination of the time and time zone accurately communicates the point in time being defined. For example, 2005-02-11T12:15:23.000+10:00 sent for a NMI in NSW refers to a local time of 13:15:23 on the 11/02/2005 (since Daylight Savings is active).

6. TRANSACTION MODELS

6.1. Background

(a) B2B Procedures have been developed based on Request/Response and Notification Business Transaction patterns to facilitate the electronic transfer of business documents.

(b) These patterns have been adapted from the ebXML Business Process Specification (ebXML Process Spec), and UN/CEFACT Modelling Methodology (UMM).

(c) Note that the way in which these Transaction patterns have been realised in aseXML is referred to as the Transaction Model and details the flow of aseXML Transactions and Acknowledgements between Participants via a centralised e-Hub.

6.2. Delivery Protocols

(a) Participants will be able to use the MSATS B2B Handler (FTP) or the SMP Hub (Webservices) for communicating B2B transactions (or a combination of both). The MSATS B2B Handler will be selected by default.

(b) Participants will be required to select the protocol they wish to use for B2B transactions at the Transaction Group level (e.g. SORD etc.). The selected method will be used for both sending and receiving messages.

(c) The e-Hub will validate each incoming message/file to determine if the Participant is sending the message/file using the opted interfacing method. The e-Hub will trigger an exception if the Participant sends a message/file using an interfacing method that is not opted in the portal.

6.3. Changing Protocols

(a) Participants will be able to update their protocol configuration in the e-Hub via the B2B Browser Application.

(b) Participants must conduct sufficient testing prior to migrating protocols in production systems.

(c) Participants will need to coordinate migration of protocols with AEMO, including but not limited to testing and cutover.

(d) AEMO will notify the market when a Participant plans to, starts, and ends migration of protocols.

6.4. MSATS B2B Handler Transaction Flow Model

6.4.1. File Transfer and Acknowledgement Protocol (FTP)

(a) The MSATS B2B Handler facilitates the flow of aseXML Transactions and Acknowledgements between Participants using an extension of the MSATS File Exchange Protocol.

(b) This protocol is illustrated for a variety of scenarios in Figure 2-Figure 9, and the associated text.

(c) The activity diagrams (Figure 2, Figure 4, Figure 6 and Figure 8) illustrate each of the major activities and decision points of the protocol. These diagrams are then organised with corresponding sequence diagrams to illustrate four possible scenarios associated with Transaction and Acknowledgement delivery (note: this does not represent a complete list of possible scenarios):

(i) Normal processing, i.e. no errors or schema validation failures.

(ii) Message containing aseXML Transactions fails schema validation at the MSATS B2B Handler.

(iii) The Outbox of the Recipient is full when checked by the MSATS B2B Handler.

(iv) *ase:MessageAcknowledgement* returned from Recipient to Initiator fails validation at the MSATS B2B Handler.

Figure 2 FTP Activity Diagram - Acknowledgement Model for normal processing

Activities and decision points that form part of a normal processing cycle are highlighted in blue, i.e. no exceptions.

Figure 3 FTP Sequence Diagram - Acknowledgement Model for normal processing

6.4.1.1. FTP – normal processing

(a) Figure 2 (activity diagram) and Figure 3 (sequence diagram) illustrate the normal processing scenario of the File Transfer and Acknowledgement Protocol as implemented by the MSATS B2B Handler.

(b) The diagrams above illustrate the default behaviour of Participant Gateways and the MSATS B2B Handler.

(c) The activities and decision points from Figure 2 that are involved in this scenario are highlighted in blue.

(d) The steps which must be followed in a normal processing scenario, from the perspective of the Initiator, are as follows (where the step numbers equate to the steps in the sequence diagram):

Step 1: The Initiator must use reasonable endeavours to first check that the Recipient Participant is not Stopped (see Section 5.4.3 for the description of this term). If the Recipient is not Stopped the Initiator must transfer the Initiator's Message to the Initiator's MSATS B2B Handler Inbox. The Initiator must ensure that the Message is sent initially as a file with a ".tmp" extension, and renamed with a ".zip" extension upon completion of the copy process.

Step 2: AEMO must ensure that the MSATS B2B Handler reads the compressed file (.zip) from the Initiator Inbox.

Step 3: AEMO must ensure that the MSATS B2B Handler decompresses the B2B file and validates it as described in section 5.4.3.

Step 4: AEMO must ensure that the MSATS B2B Handler checks that the Recipient, as specified in the *To* field, has space for the Message in their Outbox.

Step 5: If the Message fails one of the validations outlined in steps 3 and 4 above, then the relevant Participant must comply with Sections 6.4.1.2 and 6.4.1.3 of this Procedure. In all other circumstances, AEMO must ensure that the MSATS B2B Handler copies (via a .tmp file) the unmodified ".zip" file from the Inbox of the Initiator to the Outbox of the Recipient.

Step 6: Upon successful completion of the ".zip" file copy, AEMO must ensure that the MSATS B2B Handler writes a Hub Acknowledgement to the Outbox of the Initiator, as notification of the successful transfer. AEMO must ensure that the Hub Acknowledgement is an *ase:MessageAcknowledgement* addressed from the MSATS B2B Handler, and is written as an uncompressed file with an ".ac1" extension (via a .tmp file). Participants are under no obligation to process this file.

Step 7: The Recipient must use reasonable endeavours to retrieve the ".zip" file from the Outbox.

Step 8: The Recipient must decompress the ".zip" file and must validate the contents of the aseXML Message. This validation may include aseXML schema validation, e.g. the Recipient may confirm the contents of the *To* and *From* fields of the aseXML Message for validity.

Step 9: If the received aseXML Message fails validation then the Recipient must generate a negative *ase:MessageAcknowledgement* or *ase:Event* and post this as an uncompressed ".ack" file to their MSATS B2B Handler Inbox. In the case of validation failure, no further processing of the Message contents is required by the Recipient. If the Message passes validation then the Recipient must generate a positive *ase:MessageAcknowledgement* and again post this to their Inbox. In this case, the Recipient must continue with further processing of the aseXML Transactions contained within the Message.

(i) In either case the Recipient must ensure that any ".ack" file generated has an identical file name to the received ".zip" file, except in the case where there is substitution of the ".ack" file extension.

(ii) A Recipient must ensure that an Acknowledgement file is initially written to the Inbox with a ".tmp" extension, and is only renamed with an ".ack" extension upon completion of the "Put" process.

Step 10: AEMO must ensure that the MSATS B2B Handler retrieves the ".ack" file from the Inbox of the Recipient.

Step 11: AEMO must ensure that the MSATS B2B Handler performs aseXML schema validation on the ".ack" file, and validates the contents of the aseXML *To* and *From* fields. In the case of validation failure the file must be dealt with in accordance with Section 6.4.1.2 of this Procedure.

Step 12: If the ".ack" file passes validation, AEMO must ensure that the MSATS B2B Handler copies the ".ack" file, unmodified, from the Inbox of the Recipient to the Outbox of the Initiator.

Step 13: AEMO must ensure that the MSATS B2B Handler then deletes the corresponding ".zip" file from the Outbox of the Recipient. Identification of matching ".ack" and ".zip" files is achieved by file name comparison.

Step 14: The Recipient must use reasonable endeavours to verify that the ".zip" file has been removed from their Outbox. If at the time of verification the "zip" file has not been removed, the Recipient must use reasonable endeavours to verify its removal on the next "cycle".

Step 15: If the ".zip" file has been removed from a Recipient's Outbox, the Recipient must use reasonable endeavours to delete the corresponding ".ack" file from their Inbox. Identification of matching ".zip" and ".ack" files is achieved by file name comparison.

Step 16: The Initiator must use reasonable endeavours to retrieve the ".ack" file from their Outbox.

Step 17: The Initiator must validate the contents of the ".ack" file. This validation should include, but is not restricted to, aseXML schema validation.

Step 18: If the ".ack" does not pass validation the Recipient must use reasonable endeavours to follow the same steps as outlined for the MSATS B2B Handler in Section 6.4.1.2 of this Procedure. Otherwise, if the ".ack" is valid, the Recipient must use reasonable endeavours to delete the corresponding ".zip" file from their Inbox. Identification of the correct ".zip" file is achieved by filename comparison of ".zip" and ".ack" files in the Inbox and Outbox, respectively.

Step 19: AEMO must ensure that the MSATS B2B Handler verifies if the ".zip" file has been removed from the Inbox of the Initiator. If during this verification the MSATS B2B Handler determines that the zip file has not been removed, AEMO must ensure the MSATS B2B Handler checks again for its removal on the next "cycle".

Step 20: If the ".zip" file has been removed, AEMO must ensure that the MSATS B2B Handler deletes the corresponding ".ack" and ".ac1" files from the Outbox of the Initiator. Identification of matching ".zip" and ".ack" or ".ac1" files is achieved by file name comparison.

6.4.1.2. FTP – validation failure of B2B aseXML Message

Figure 4 FTP Activity Diagram - Acknowledgement Model for invalid aseXML Message

Activities and decision points highlighted in blue for scenario where Hub receives an invalid aseXML Message from initiating gateway.

Figure 5 FTP Sequence Diagram - Acknowledgement Model for invalid aseXML Message

(a) Figure 4 and Figure 5 illustrate the scenario where the aseXML B2B Message fails aseXML schema validation by the MSATS B2B Handler. This process would also occur where validation of the aseXML *To* and *From* fields failed.

(b) Steps 1-3 are the same as described in Section 6.4.1.1~~6.4.1.1~~.

(c) Step 4: AEMO must ensure that the MSATS B2B Handler writes a negative *ase:MessageAcknowledgement*, or standalone *ase:Event* to the Outbox of the Initiator, as an uncompressed ".ack" file. An appropriate *ase:Code* identifies where the Outbox of the Recipient is full (*ase:Code*="111") or where the header is incorrect (*ase:Code*="7") or schema invalid (*ase:Code*="2").

(d) Steps 5-9 are the same as described in steps 16-20 of Section 6.4.1.1~~6.4.1.1~~.

6.4.1.3. FTP – Recipient Outbox full

Figure 6 FTP Activity Diagram - Acknowledgement Model when Recipient Outbox full

Activities and decision points highlighted in blue for scenario where the Recipient Outbox is full when checked by the MSATS B2B Handler.

Figure 7 FTP Sequence Diagram - Acknowledgement Model when Recipient Outbox full

(a) Figure 6 and Figure 7 illustrate the scenario where the MSATS B2B Handler determines that the Recipient has exceeded the flow control file limit that allows them to receive an aseXML Message, or ".zip" file.
Note: this check is only performed for aseXML Messages containing Transactions and *ase:TransactionAcknowledgements*. The check is not performed on ".ack" files containing only *ase:MessageAcknowledgements*.

(b) Steps 1-4 are the same as described in Section 6.4.1.1~~6.4.1.1~~.

(c) Step 5: AEMO must ensure that the MSATS B2B Handler writes a negative *ase:MessageAcknowledgement*, or standalone *ase:Event* to the Outbox of the Initiator, as an uncompressed ".ack" file. An appropriate *ase:Code* identifies where the Outbox of the Recipient is full (*ase:Code*="111").

(d) Steps 6-10 are the same as described in steps 16-20 of Section 6.4.1.1~~6.4.1.1~~.

Formatt

Formatt

6.4.1.4. FTP – ase:MessageAcknowledgement validation failure

Figure 8 FTP Activity Diagram - Acknowledgement Model for MACK validation failure

Activities and decision points highlighted in blue for scenario where Hub receives an invalid aseXML Message Acknowledgement from the Recipient gateway

Figure 9 FTP Sequence Diagram - Acknowledgement Model for MACK validation failure

(a) Figure 8 and Figure 9 illustrate the scenario where the *ase:MessageAcknowledgement* fails validation:

(i) Schema validation failure of either an *ase:MessageAcknowledgement* or standalone *ase:Event*.

(ii) Invalid Participant in *To* field – the value in the *To* field does not match the B2B Initiator.

(b) Steps 1-11: These are the same as for the "default" scenario, and are not repeated here (see Section 6.4.1.1~~6.4.1.1~~).

Formatt

Note: the following steps are applicable to both AEMO and a Participant upon validation failure of an *ase:MessageAcknowledgement*.

(c) Step 12: Upon *ase:MessageAcknowledgement* validation failure AEMO must ensure that the MSATS B2B e-Hub logs an error Message. The relevant Participant will be advised by the notification mechanism to be agreed by industry and published by AEMO.

(i) The Initiator of the ".ack" must remove the offending ".ack" file from the Inbox and, if necessary, replace it with a valid ".ack" file to allow the file exchange process to complete. AEMO must ensure that the MSATS B2B Handler does not transfer the offending ".ack" file to the Outbox of the Initiator of the original Message.

6.4.2. FTP – Worked Example

(a) The following diagram (Figure 10) illustrates the full service order request/response process (without exceptions) using the prescribed Transaction model (with Messages sent via the MSATS B2B Handler):

Figure 10 Example of FTP Transaction Model for Service Order process

6.5. SMP Hub Transaction Flow Model

6.5.1. Webservices Transfer and Acknowledgement Protocol

(a) The SMP Hub facilitates the flow of aseXML Transactions and Acknowledgements between Participants using a RESTful Webservice Protocol.

(b) This protocol is illustrated for a variety of scenarios in Figure 11 to Figure 15 and the associated text.

(i) Normal processing, i.e. no errors or schema validation failures.

(ii) Message containing aseXML Transactions fails schema validation at the SMP Hub.

(iii) The Recipient is unavailable when the SMP Hub is attempting to deliver a message.

(iv) *ase:MessageAcknowledgement* returned from Recipient to Initiator fails validation at the SMP Hub.

6.5.1.1. Webservices – normal processing

Figure 11 Webservices Sequence Diagram - Acknowledgement Model normal processing

(a) Figure 11 illustrates the normal processing scenario of the Webservices Protocol as implemented by the SMP Hub.

(b) The diagram in this section illustrates the default behaviour of Participant Gateways and the SMP Hub.

(c) Step 1: The Initiator must use reasonable endeavours to first check that the Recipient Participant is not Stopped. If the Recipient is not Stopped the Initiator will invoke the webservice using the URL of the e-Hub.

(d) Step 1a: The SMP Hub will validate the incoming message (see section 5.5.8). Note: the SMP validation will occur each time a message is received by the SMP Hub, but is only depicted once in the diagram for simplicity and readability.

(e) Step 2: Upon successfully passing validation the SMP Hub will provide a positive Hub Acknowledgement back to the initiator as an *ase:MessageAcknowledgement* from the SMP Hub. Participants are under no obligation to process this file.

(f) Step 3: Upon successfully passing validation the SMP Hub will invoke a webservices call to the recipient using their specific URL and deliver the message.

(g) Step 4: On receipt of the message the recipient must validate the contents of the aseXML Message. This validation may include aseXML Schema validation, e.g. the Recipient may validate the contents of the *To* and *From* fields of the aseXML Message. On completion of this validation the Recipient will send an *ase:MessageAcknowledgement* back to the Initiator.

(h) Step 5: Upon receipt of the *ase:MessageAcknowledgement* the SMP Hub will invoke a webservices call to the Initiator using their specific URL and deliver. The Initiator must validate the contents of the *ase:MessageAcknowledgement.* This validation may include aseXML Schema validation, e.g. the Initiator may validate the contents of the *To* and *From* fields.

(i) Step 6: On completion of this validation the recipient will send a HTTP response to the webservice call to the e-Hub without an aseXML payload.

6.5.1.2. Webservices – validation failure of B2B aseXML Message

Figure 12 Webservices Sequence Diagram – Acknowledgement Model for invalid aseXML Message

(a) Figure 12 illustrates the scenario where the aseXML B2B Message fails aseXML schema validation by the SMP Hub. This process would also occur where validation of the aseXML *To* and *From* fields failed.

(b) The diagram in this section illustrates the expected behaviour of Participant Gateways and the SMP Hub when a message fails validation in the SMP Hub.

(c) Step 1: The Initiator must use reasonable endeavours to first check that the Recipient Participant is not Stopped (see section 5.5.9 for the description of this term). If the Recipient is not Stopped the Initiator will invoke the webservice using the URL of the e-Hub.

(d) Step 2: The SMP Hub will validate the incoming message (see section 5.5.8).

(e) Step 3: Upon failure of this validation the SMP Hub will provide a negative *ase:MessageAcknowledgement*, or standalone *ase:Event* on the return of the original Webservice call.

6.5.1.3. Webservices – Recipient unavailable

Figure 13 Webervices Sequence Diagram – Acknowledgement Model when Recipient unavailable

(a) Figure 13 illustrates the scenario of the Webservices Protocol where the SMP Hub has determined that the intended Recipient of the message is unavailable and the Water Mark – High has been exceeded.

Note: this check is only performed for aseXML Messages containing Transactions and *ase:TransactionAcknowledgements*. The check is not performed on ".ack" files containing only *ase:MessageAcknowledgements*.

(b) Step 1: The Initiator must use reasonable endeavours to first check that the Recipient Participant is not Stopped. If the Recipient is not Stopped the Initiator will invoke the webservice using the URL of the e-Hub.

(c) Step 2: The SMP Hub will validate the incoming message (see section 5.5.8).

(d) Step 3: Upon successfully passing validation the SMP Hub will provide a positive Hhub Acknowledgement back to the Initiator as an *ase:MessageAcknowledgement* from the SMP Hub. Participants are under no obligation to process this file.

(e) Step 4: Upon successfully passing validation the SMP Hub will invoke a webservices call to the recipient using their specific URL and deliver the message. If this is unavailable the SMP Hub will queue the messages and continue to attempt to deliver to the Recipient. Once the Recipient is available the SMP Hub will invoke a webservice call to the Recipient and send the message (refer to 6.5.1.1 (g) Step 4 onwards for the remainder of this process).

(f) Step 5: If the hub queue exceeds its threshold limit (see section 5.5.9) then:

(i) a Stop File will be generated; and

(ii) a webservices call will be made to the Initiator's URL to notify them of a Stop File.

(g) Step 6: On receipt of the webservices call the Initiator will provide a return to close out the webservices call.

(h) Step 7: The Initiator must use reasonable endeavours to first check that the Recipient Participant is not Stopped. If the Recipient is Stopped the Initiator can still invoke the webservice using the URL of the e-Hub.

(i) Step 8: Upon receipt of any new messages and failure of Stop File validation the SMP Hub will provide a negative *ase:MessageAcknowledgement* on the return of the original Webservice call.

6.5.1.4. Webservices – ase:MessageAcknowledgement validation failure

Figure 14 Webservices sequence diagram – message acknowledgement validation failure

(a) Steps 1-4 are the same as described in section 6.5.1.1 ((c) – (f)).

(b) Step 5: Upon *ase:MessageAcknowledgement* validation failure AEMO must ensure that the SMP Hub logs an error Message. The relevant Participant will be advised by the notification mechanism to be agreed by industry and published by AEMO.

(c) The Recipient must resubmit the message Acknowledgement when becoming aware of its validation failure.

6.5.2. Webservices – Worked Example

Figure 15 Example of Webservices Transaction Model for Service Order process

6.6. A Summary of Transaction Model Exception Points

(a) The following table summarises the key exception points through the Transaction Model. Participants must use reasonable endeavours to comply with the actions to be taken as set out in the following table.

Exception	Who needs to take action	Action to be taken	Applicability	
			MSATS B2B Handler	SMP Hub
(i) Initiator determines that intended Recipient of a B2B Message has reached flow control limit.	Initiator	Once relevant Timing Requirements are exceeded, raise issue with appropriate technical contact for Recipient, as indicated by NEM Retail Operations Contacts List.	Y	Y

Exception		Who needs to take action	Action to be taken	Applicability	
				MSATS B2B Handler	SMP Hub
(ii)	B2B Message sent by Initiator fails e-Hub validation.	Initiator	Address the indicated reason for failure and resend.	Y	Y
(iii)	A Hub Acknowledgement not received in response to a B2B Message sent by Initiator. See Section 5.9~~5.9~~ for Timing Requirements.	Initiator	Contact AEMO to raise issue of potential performance issue.	Y	Y
(iv)	B2B Message sent by Initiator fails Business Receipt validation by the Recipient, ie Initiator receives a negative ase: *MessageAcknowledgement*	Initiator	Address the indicated reason for failure and resend.	Y	Y
(v)	*Ase:MessageAcknowledgement* not received in response to a B2B Transaction. Participants should refer to the Section 5.9~~5.9~~ for Timing Requirements.	Initiator	Raise potential performance issue with appropriate technical contact for Recipient, as indicated by NEM Retail Operations Contacts List.	Y	Y
(vi)	*Ase:TransactionAcknowledgement* not received within appropriate timeframe from posting a B2B Message to the e-Hub (and all intermediary events have occurred successfully). (Participants should refer to the B2B Procedures for specific Timing Requirements, however the Business Acceptance/Rejection is typically required within one business day.)	Initiator	Raise non-delivery issue with appropriate technical contact for Recipient, as indicated by NEM Retail Operations Contacts List.	Y	Y

Formatt

Formatt

7. INTEROPERABILITY

(a) There is no requirement for the Initiator to be aware of the Recipient's protocol choice when initiating the message exchange via the e-Hub. The e-Hub will perform this validation and send it in the correct format to the recipient.

(b) All Participants must send and will receive messages based on their preference set in the e-Hub. If that preference is changed during a transaction cycle any inbound messages will be delivered based on this preference.

7.1. Webservices to FTP Hokey Pokey

(a) Messages initiated using webservices will be transformed by the e-Hub and delivered via FTP Hokey Pokey where this is the Recipient's protocol perference as selected in the B2B Browser Application (FTP Hokey Pokey by default).

Figure 16 Interoperability – Webservices to FTP sequence diagram

7.2. FTP Hokey Pokey to Webservices

(a) Messages initiated using FTP Hokey Pokey will be transformed by the e-Hub and delivered via webservices where the Recipient where this is the protocol perference as selected in the B2B Browser Application.

Figure 17 Interoperability – FTP to Webservices sequence diagram

8. NOTIFIED PARTIES

8.1. Overview

(a) The concept of Notified Parties exists to ensure that all relevant Participants associated with a given NMI are made aware of changes and/or activity that is performed by another associated Participant in a timely manner.

(b) The Initiator is responsible for populating the list of Notified Parties in a given transaction to utilise the Notified Party functionality of the e-Hub.

(c) The e-Hub allows for two options for an Initiator to notify other associated Participants:

(i) e-Hub-generated notifications; the e-Hub will generate the required NotifiedParty transactions at the appropriate trigger points as defined in the B2B Procedure Service Order Process, to the Participants specified in the ServiceOrderRequest, and containing the corresponding payload information as defined in the B2B Procedure One Way Notification Process.

(ii) Initiator-generated notifications: the Initiator will be responsible to generate the NotifiedParty transactions required, in accordance with the B2B Procedures referenced in clause (c)(i) above. These transactions will follow the standard message patterns.

(d) The notifications sent by the e-Hub will only be applicable for ServiceOrderRequests with the ActionType of 'New'. Cancellations (ActionType = 'Cancel') will not trigger notifications.

8.2. Sequence Diagrams – e-Hub generated notifications

(a) The following diagrams in this section illustrate when the Initiator has opted to leverage the e-Hub functionality to manage notifications to Notified Parties (i.e. automatically generate the NotifiedParty transaction/s associated to the ServiceOrderRequest).

(b) This functionality is available regardless of the protocol selected by the Initiator.

8.2.1. Normal Processing

Figure 18 e-Hub generated notifications – normal processing

(a) Steps 1 to 6 are as described in section 6.5.1.1.

(b) Step 7: on receipt and successful validation of the *ase:MessageAcknowledgement* from the recipient, the e-Hub will create and send the NotifiedParty transaction (as specified in the B2B Procedure One Way Notification Process).

(c) Step 8: On receipt of the NotifiedParty transaction the Notified Party must validate the contents of the aseXML Message. This validation may include aseXML Schema validation, e.g. the Recipient may validate the contents of the *To* and *From* fields of the aseXML Message. On completion of this validation the Notified Party will send an *ase:MessageAcknowledgement* back to the Initiator.

The e-Hub will receive the *ase:MessageAcknowledgement* and validate. The hub will not transmit this back to the Initiator.

(d) Step 9: The Notified party will provide an *ase:TransactionAcknowledgement* back to the Initiator.

(e) Step 10: on receipt and successful validation of the *ase:TransacationAcknowledgement* from the Notified Party, the e-Hub will create and send a Hub Acknowledgement back to the Notified Party.

(f) Step 11: the e-Hub will send an *ase:MessageAcknowledgement* to the Notified Party on behalf of the Initiator (as the e-Hub generated the NotifiedParty transaction in step 8). This will be addressed from the Initiator of the original request.

(g) Step 12: the Notified Party will validate and send a Hub Acknowledgement back to the e-Hub.

(h) Step 13: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Notified Party (step 9), the e-Hub will determine if the Initiator has opted in to receiving *ase:TransactionAcknowledgement* from Notified Parties. If opted in, the e-Hub will send the *ase:TransactionAcknowledgement* from the Notified Party to the Initiator in the form of a NotifiedParty transaction. If not, skip to step 19.

(i) Step 14: on receipt and successful validation of the NotifiedParty transaction from the Notified Party (generated by the e-Hub with the Notified Party Participant ID), the Initiator will create and send an *ase:MessageAcknowledgement* to the Notified Party (this will be absorbed by the e-Hub).

(j) Step 15: the Initiator will send an *ase:TransactionAcknowledgement* to the Notified Party.

(k) Step 16: the e-Hub will validate and send a Hub Acknowledgement to the Initiator.

(l) Step 17: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Initiator, the e-Hub will create and send an *ase:MessageAcknowledgement* back to the Initiator.

(m) Step 18: the Initiator will validate and send a Hub Acknowledgement to the e-Hub.

(n) Step 19: The Recipient will provide an *ase:TransactionAcknowledgement* back to the Initiator.

(o) Step 20: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Recipient, the e-Hub will create and send a Hub Acknowledgement back to the Recipient.

(p) Step 21: the e-Hub will send the *ase:TransactionAcknowledgement* from the Recipient to the Initiator.

(q) Step 22: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Recipient, the Initiator will send an *ase:MessageAcknowledgement* back to the e-Hub.

(r) Step 23: the e-Hub will send the *ase:MessageAcknowledgement* from the Initiator to the Recipient.

(s) Step 24: the Recipient will validate and send a Hub Acknowledgement to the e-Hub.

(t) Step 25: the Recipient will send a ServiceOrderResponse to the Initiator.

(u) Step 26: the e-Hub will validate and send a Hub Acknowledgement to the Recipient.

(v) Step 27: the e-Hub will deliver the ServiceOrderResponse to the Initiator.

(w) Step 28: the Initiator will generate and send an *ase:MessageAcknowledgement* from the Initiator to the Recipient.

(x) Step 29: on receipt and successful validation of the *ase:MessageAcknowledgement* from the Initiator, the e-Hub will create and send the NotifiedParty transaction (as specified in the B2B Procedure One Way Notification Process) to the Notified Party.

(y) Step 30: On receipt of the NotifiedParty transaction the Notified Party must validate the contents of the aseXML Message. On completion of this validation the Notified Party will send an *ase:MessageAcknowledgement* back to the Initiator (this will be absorbed by the e-Hub).

(z) Step 31: the e–Hub will deliver the *ase:MessageAcknowledgement* from the Initiator to the Recipient.

(aa) Step 32: the Recipient will provide a Hub Acknowledgement to the e-Hub.

(bb) Step 33: The Initiator will provide an *ase:TransactionAcknowledgement* back to the Recipient.

(cc) Step 34: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Initiator, the e-Hub will create and send a Hub Acknowledgement back to the Initiator.

(dd) Step 35: the e-Hub will send the *ase:TransactionAcknowledgement* from the Initiator to the Recipient.

(ee) Step 36: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Initiator, the Recipient will send a *ase:MessageAcknowledgement* back to the e-Hub.

(ff) Step 37: the e-Hub will send the *ase:MessageAcknowledgement* from the Recipient to the Initiator.

(gg) Step 38: the Initiator will validate and send a Hub Acknowledgement to the e-Hub.

(hh) Steps 39 to 48 are similar to steps 9 to 18.

8.2.2. Recipient sends negative message acknowledgement

Figure 19 Notified Parties – Recipient sends negative message acknowledgement

(a) Steps 1 to 6 are as described in section 6.5.1.1, with the exception that in step 4 a message acknowledgement of Reject is sent by the Recipient.

(b) In this scenario, the Notified Party does not receive any NotifiedParty transaction of the Service Order Request.

8.2.3. Recipient sends a Business Rejection

Figure 20 Notified Parties – Recipient sends a Business Rejection

(a) Steps 1 to 6 are as described in section 6.5.1.1.

(b) Steps 7 to 18 are as described in section 8.2.1.

(c) Step 19: The Recipient will provide an *ase:TransactionAcknowledgement* back to the Initiator with a *Status* of 'Reject'.

(d) Step 20: on Receipt and successful validation of the *ase:TransactionAcknowledgement* from the Recipient, the e-Hub will create and send a Hub Acknowledgement back to the Recipient.

(e) Step 21: the e-Hub will send the *ase:TransactionAcknowledgement* from the Recipient to the Initiator.

(f) Step 22: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Recipient, the Initiator will send a *ase:MessageAcknowledgement* back to the e-Hub.

(g) Step 23: the e-Hub will create and send the NotifiedParty transaction (as specified in the B2B Procedure One Way Notification Process) to identify the Rejection. This will cancel the Notification.

(h) Step 24: the Notified Party will validate and send a Hub Acknowledgement to the e-Hub.

(i) Step 25: the e-Hub will send the *ase:MessageAcknowledgement* from the Initiator to the Recipient.

(j) Step 26: the Recipient will validate and send a Hub Acknowledgement to the e-Hub.

(k) Steps 27 to 36 are similar to steps 9 to 18 as described in section 8.2.1.

8.2.4. Incorrect Notified Party

Figure 21 Notified Parties – Incorrect Notified Party

(a) Steps 1 to 6 are as described in section 6.5.1.1.

(b) Step 7: on receipt and successful validation of the *ase:MessageAcknowledgement* from the recipient, the e-Hub will create and send the NotifiedParty transaction (as specified in the B2B Procedure One Way Notification Process).

(c) Step 8: On receipt of the NotifiedParty transaction the Notified Party must validate the contents of the aseXML Message. This validation may include aseXML Schema validation, e.g. the Recipient may validate the contents of the *To* and *From* fields of the aseXML Message. On completion of this validation the Notified Party will send an *ase:MessageAcknowledgement* back to the Initiator.

The e-Hub will receive the *ase:MessageAcknowledgement* and validate. The e-Hub will not transmit this back to the Initiator.

(d) Step 9: The Recipient will provide an *ase:TransactionAcknowledgement* back to the Initiator.

(e) Step 10: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the recipient, the e-Hub will create and send a Hub Acknowledgement back to the Recipient.

(f) Step 11: the e-Hub will send the *ase:TransactionAcknowledgement* from the Recipient to the Initiator.

(g) Step 12: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Recipient, the Initiator will send a *ase:MessageAcknowledgement* back to the e-Hub.

(h) Step 13: the e-Hub will send the *ase:MessageAcknowledgement* from the Initiator to the Recipient.

(i) Step 14: the Recipient will validate and send a Hub Acknowledgement to the e-Hub.

(j) Step 15: The Notified party will provide an *ase:TransactionAcknowledgement* back to the Initiator with a *Status* of 'Reject'.

(k) Step 16: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Notified Party, the e-Hub will remove the Participant ID of the Notified Party from the distribution list for this transaction. The e-Hub will create and send a Hub Acknowledgement back to the Notified Party.

(l) Step 17: the e-Hub will generate and send an *ase:MessageAcknowledgement* from the Initiator to the Recipient. This will be addressed from the Initiator of the original request.

(m) Step 18: the Recipient will validate and send a Hub Acknowledgement to the e-Hub.

(n) Step 19: the e-Hub will send the *ase:TransactionAcknowledgement* from the Notified Party to the Initiator in the form of a NotifiedParty transaction.

(o) Step 20: on receipt and successful validation of the NotifiedParty transaction from the Notified Party (was sent by the e-Hub with the Notified Party Participant ID), the Initiator will create and send an *ase:MessageAcknowledgement* to the Notified Party (this will be absorbed by the e-Hub).

(p) Step 21: the Initiator will send an *ase:TransactionAcknowledgement* to the Notified Party.

(q) Step 22: the e-Hub will validate and send a Hub Acknowledgement to the Initiator.

(r) Step 23: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Initiator, the e-Hub will create and send an *ase:MessageAcknowledgement* back to the Initiator.

(s) Step 24: the Initiator will validate and send a Hub Acknowledgement to the e-Hub.

(t) Step 25 to 36: Initiator to trigger a NotifiedParty transaction that will follow standard Acknowledgement patterns. The e-Hub will add the Participant ID of the Notified Party to the distribution list for this transaction.

(u) Step 37: the Recipient will send a ServiceOrderResponse to the Initiator.

(v) Step 38: the e-Hub will validate and send a hub acknowledgement to the Recipient.

(w) Step 39: the e-Hub will deliver the ServiceOrderResponse to the Initiator.

(x) Step 40: the Initiator will generate and send an *ase:MessageAcknowledgement* from the Initiator to the Recipient.

(y) Step 41: on receipt and successful validation of the *ase:MessageAcknowledgement* from the Initiator, the e-Hub will create and send the NotifiedParty transaction (as specified in the B2B Procedure One Way Notification Process) to the Notified Party.

(z) Step 42: On receipt of the NotifiedParty transaction the Notified Party must validate the contents of the aseXML Message. On completion of this validation the Notified Party will send an *ase:MessageAcknowledgement* back to the Initiator (this will be absorbed by the e-Hub).

(aa) Step 43: the e–Hub will deliver the *ase:MessageAcknowledgement* from the Initiator to the Recipient.

(bb) Step 44: the Recipient will provide a Hub Acknowledgement to the e-Hub.

(cc) Step 45: The Initiator will provide an *ase:TransactionAcknowledgement* back to the Recipient.

(dd) Step 46: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Initiator, the e-Hub will create and send a Hub Acknowledgement back to the Initiator.

(ee) Step 47: the e-Hub will send the *ase:TransactionAcknowledgement* from the Initiator to the Recipient.

(ff) Step 48: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Initiator, the Recipient will send a *ase:MessageAcknowledgement* back to the e- Hub.

(gg) Step 49: the e-Hub will send the *ase:MessageAcknowledgement* from the Recipient to the Initiator.

(hh) Step 50: the Initiator will validate and send a Hub Acknowledgement to the e-Hub.

(ii) Steps 51 to 60 are similar to steps 9 to 18 as described in section 8.2.1.

8.2.5. Notified Party Stopped

Figure 22 Notified Parties – Notified Party Stopped

(a) Steps 1 to 6 are as described in section 6.5.1.1.

(b) Step 7: the e-Hub will generate a NotifiedParty transaction to indicate that the Notified Party has a stop file in place.

(c) Step 8: on receipt and successful validation of the NotifiedParty transaction from the Notified Party (generated by the e-Hub with the Notified Party Participant ID), the Initiator will create and send an *ase:MessageAcknowledgement* to the Notified Party (this will be absorbed by the e-Hub).

(d) Step 9: the Initiator will send an *ase:TransactionAcknowledgement* to the Notified Party.

(e) Step 10: the e-Hub will validate and send a Hub Acknowledgement to the Initiator.

(f) Step 11: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Initiator, the e-Hub will create and send an *ase:MessageAcknowledgement* back to the Initiator.

(g) Step 12: the Initiator will validate and send a Hub Acknowledgement to the e-Hub.

(h) Step 13: The Initiator will need to use other means of communications to notify the Notified Party that has been Stopped.

(i) Step 14: The Recipient will provide an *ase:TransactionAcknowledgement* back to the Initiator.

(j) Step 15: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Recipient, the e-Hub will create and send a Hub Acknowledgement back to the Recipient.

(k) Step 16: the e-Hub will send the *ase:TransactionAcknowledgement* from the Recipient to the Initiator.

(l) Step 17: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Recipient, the Initiator will send an *ase:MessageAcknowledgement* back to the e-Hub.

(m) Step 18: the e-Hub will send the *ase:MessageAcknowledgement* from the Initiator to the Recipient.

(n) Step 19: the Recipient will validate and send a Hub Acknowledgement to the e-Hub.

(o) Step 20: the Recipient will send a ServiceOrderResponse to the Initiator.

(p) Step 21: the e-Hub will validate and send a Hub Acknowledgement to the Recipient.

(q) Step 22: the e-Hub will deliver the ServiceOrderResponse to the Initiator.

(r) Step 23: the Initiator will generate and send an *ase:MessageAcknowledgement* from the Initiator to the Recipient.

(s) Step 24: the e-Hub will deliver the *ase:MessageAcknowledgement* from the Initiator to the Recipient.

(t) Step 25: the Recipient will provide a Hub Acknowledgement to the e-Hub.

(u) Step 26: The Initiator will provide an *ase:TransactionAcknowledgement* back to the Recipient.

(v) Step 27: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Initiator, the e-Hub will create and send a Hub Acknowledgement back to the Initiator.

(w) Step 28: the e-Hub will send the *ase:TransactionAcknowledgement* from the Initiator to the Recipient.

(x) Step 29: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Initiator, the Recipient will send a *ase:MessageAcknowledgement* back to the e- Hub.

(y) Step 30: the e-Hub will send the *ase:MessageAcknowledgement* from the Recipient to the Initiator.

(z) Step 31: the Initiator will validate and send a Hub Acknowledgement to the e-Hub.

(aa) Step 32: if the Notified Party is still Stopped, the e-Hub will generate a NotifiedParty transaction to indicate that the Notified Party has a stop file in place. If the Notified Party is no longer Stopped, then go to step 39 as defined in section 8.2.1.

(bb) Step 33: on receipt and successful validation of the NotifiedParty transaction from the Notified Party (generated by the e-Hub with the Notified Party Participant ID), the Initiator will create and send an *ase:MessageAcknowledgement* to the Notified Party (this will be absorbed by the e-Hub).

(cc) Step 34: the Initiator will send an *ase:TransactionAcknowledgement* to the Notified Party.

(dd) Step 35: the e-Hub will validate and send a Hub Acknowledgement to the Initiator.

(ee) Step 36: on receipt and successful validation of the *ase:TransactionAcknowledgement* from the Initiator, the e-Hub will create and send an *ase:MessageAcknowledgement* back to the Initiator.

(ff) Step 37: the Initiator will validate and send a Hub Acknowledgement to the e-Hub.

(gg) Step 38: The Initiator will need to use other means of communications to notify the Notified Party that has been Stopped.

8.3. Trigger Points for Notifications

(a) Notifications will be triggered by the e-Hub to the Notified Party at the following points (Service Orders used as an example below):

(i) When a Recipient provides a positive Message Acknowledgement for a ServiceOrderRequest; NotifiedParty transaction *NotificationStatus* of 'SO Requested'.

(ii) When a Recipient provides a negative transaction acknowledgement for a ServiceOrderRequest; NotifiedParty transaction *NotificationStatus* of 'SO Rejected'.

(iii) When a ServiceOrderResponse is sent by the Recipient; NotifiedParty transaction *NotificationStatus* of 'SO Completion'.

(b) If the Initiator has opted for the e-Hub to manage notifications to Notified Parties, and the Notified Party rejects the NotifiedParty transaction, a new NotifiedParty transaction with a *NotificationStatus* of 'Rejection by Notified Party' will be triggered by the e-Hub to the Initiator.

(c) Notifications must be triggered manually by the Initiator to the Notified Party at the following points if *NotifiedPartyID* is not populated in the ServiceOrderRequest (Service Orders used as an example below):

(i) When a Recipient provides a positive message acknowledgement for a ServiceOrderRequest; NotifiedParty transaction *NotificationStatus* of 'SO Requested'.

(ii) When a Recipient provides a negative transaction acknowledgement for a ServiceOrderRequest; NotifiedParty transaction *NotificationStatus* of 'SO Rejected'.

(iii) When a ServiceOrderResponse is sent by the Recipient; NotifiedParty transaction *NotificationStatus* of 'SO Completion'.

(d) Participants must not populate both the *NotifiedPartyID* in the ServiceOrderRequest and send manually triggered notifications.

8.4. Notification Format

(a) The notification will be in the form of a NotifiedParty transaction, as defined in the B2B Procedure One Way Notification Process.

(b) The Priority will be as per the related transaction e.g. for a High Priority Service Order, all related notifications will also be classified as High Priority.

8.5. Notification Bundling

(a) The e-Hub will bundle NotifiedParty transactions in accordance with clause 2.13 (a).

8.6. Error Handling

8.6.1. Updating the Notified Parties List

(a) Where a Notified Party has been nominated in error (i.e. not responsible for the NMI), and has subsequently rejected the Notification, the Initiator is expected to trigger a NotifiedParty transaction to the correct Notified Party(s).

(i) If the Service Order is not yet completed, the NotifiedParty transaction (*NotificationStatus* of 'SO Requested') will contain the ServiceOrderRequest in the payload,

(ii) If the Service Order has been completed, the NotifiedParty transaction (*NotificationStatus* of 'SO Completion') will contain the ServiceOrderResponse in the payload.

(b) On receipt of the NotifiedParty transaction, the e-Hub will update the Notified Parties list maintained for the given initiating transaction.

(c) Any subsequent notifications in relation to the original initiating transaction will be sent to the updated Notified Parties.

(d) The Recipient of the initiating transaction or other unchanged Notified Parties will not be aware of the updated Notified Parties.

9. CONTINGENCY RECOVERY REQUIREMENTS

9.1. Overview of National B2B Infrastructure

(a) The following diagram illustrates the components of the National B2B Infrastructure that are covered by contingency requirements:

Figure 23 National B2B Infrastructure

(a) As shown, the term "National B2B Infrastructure" relates to centralised B2B e-Hub as well as components (hardware and software) maintained by Participants.

9.2. Need for Contingency Arrangements

(a) The use of a National B2B Infrastructure, coupled with the use of aseXML B2B Transactions, has been assumed as the basis for the realisation of National B2B Procedures.

(b) A series of contingency arrangements have been defined to cover situations when National B2B Infrastructure performance (IT systems and communications) does not meet the needs and priorities of Participants (and their Customers).

(c) These contingency arrangements take account of:

(i) Time frame and responsibility to advise system problems and to activate contingency arrangement;

(ii) Prioritisation of Transactions;

(iii) Alternate delivery method(s) and format;

(iv) Escalation process and timing for lengthy or persistent problems; and the

(v) Handling of contingency Transactions once normal operations resume.

(d) Also relevant to the contingency arrangements is the reduction of situations requiring contingency arrangements:

(i) Steps to ensure an appropriate level of performance, integrity, robustness and redundancy of any National B2B Infrastructure and systems; and

(ii) Potential steps Participants may take to improve their internal performance, integrity, robustness, and redundancy relevant to the B2B Procedures.

9.3. Basic Principles for Contingency Arrangements

(a) The basic principle underlying the contingency arrangements detailed in this Procedure is that the Participant activates its contingency arrangements to minimise any adverse impact on other Participants and to itself.

(b) Participants must ensure that the contingency arrangements of that Participant preserve normal business operations of other Participants whenever possible and practicable.

(i) Participants must use reasonable endeavours to ensure that that Participant's contingency arrangements preserve the Message format, Transaction models and general delivery requirements detailed earlier in this Procedure.

(ii) Participants receiving Business Documents via a contingency delivery method must use reasonable endeavours to respond using the normal delivery method, and not the contingency method originally used.

9.4. Overview of Major Contingency Requirements

9.4.1. Participants

(a) Participants must use reasonable endeavours to establish internal contingency arrangements to minimise disruption to other market Participants in the event of a material internal infrastructure failure. Participants must use reasonable endeavours to process Messages and Acknowledgements within the timeframes prescribed in this Procedure and elsewhere in the B2B Procedures.

(b) Where a Participant is unable to process Messages and/or Acknowledgements within timeframes prescribed in this Procedure or any other B2B Procedure, that Participant must as soon as reasonably practicable to inform affected parties and:

(i) detail actions and timeframes to recover; and

(ii) negotiate appropriate intermediate working arrangements.

(iii) Use the B2B Notice for Electricity B2B;

(A) to provide an email notification process to advise other parties of problems with their gateway and /or systems (that may impact other participants);

(B) to provide email notifications to advise the market of any inability to meet its obligations under B2B Procedures; and

(C) to provide email notifications as set out in (A) and (B) above that must not contain attachments.

(c) An alternative mechanism for a Participant to manage Messages in their MSATS B2B Handler Inbox and Outbox shall be provided to Participants. This mechanism is called the B2B Browser Application.

(d) In the event of a series of failures, which prevent a Participant from accessing the MSATS B2B Handler (and the B2B Browser Application is effectively unavailable), urgent B2B Messages may be sent via email as aseXML attachments (as a last resort).

9.5. Major Failure Events and Contingency Steps

(a) The following table identifies key failure events and the contingency steps that Participants and AEMO must follow (in the order shown):

Failure Event	Contingency Steps
Central MSATS B2B Handler ("hub") failure.	1. AEMO supports multiple "backup hubs". In the event of a failure of the operating MSATS B2B Handler, which prevents the business timings being achieved, AEMO will switch to a back-up hub. 2. In the unlikely event that all MSATS B2B Handlers become unavailable, Participants should defer non-urgent Messages and send all urgent B2B Messages as compressed aseXML email attachments, without password protection, adhering to the requirements specified later in this Section. 3. If the MSATS B2B Handler fails, AEMO must notify all Participants. When the MSATS B2B Handler is available after a failure, AEMO should notify all Participants.
Participant communications link failure Or Participant gateway failure	4. Participants should maintain at least one alternative communication link between their internal National Infrastructure components and the MSATS B2B handler gateways. 5. In the event of a communications failure between a Participant and the MSATS B2B Handler (including any appropriate contingency communications infrastructure), the Participant should then seek to defer non-urgent B2B Messages and must raise any urgent Messages via the industry-supported "B2B Browser Application". 6. Where the B2B Browser Application is unavailable, Participants should raise urgent B2B Messages as compressed aseXML email attachment, without password protection adhering to the requirements specified later in this Section.

Failure Event	Contingency Steps
Participant unable to issue *ase:MessageAcknowledgements*	7. Where a Participant has a temporary inability to respond to a B2B Transaction with a Message Acknowledgement, the Participant must notify any affected parties and must send the *ase:MessageAcknowledgements* as soon as they are able to. 8. Should the temporary problem be ongoing then Participants should utilise the B2B Browser Application to acknowledge Messages. 9. Note that it is likely that the Participant expecting the Message Acknowledgement would raise the issue first.
Participant unable to issue *ase:TransactionAcknowledgements*	10. Where a Participant has a temporary inability to respond to a B2B Transaction with a Transaction Acknowledgement, the Participant should manually process Transactions via the B2B Browser Application to issue negative *ase:TransactionAcknowledgements*. The Initiator of the original Transaction will assume acceptance of the Transaction unless a negative *ase:TransactionAcknowledgement* is received. 11. Should the temporary problem be ongoing then Participants should utilise the B2B Browser Application to acknowledge Transactions. 12. Note that it is likely that the Participant expecting the Transaction Acknowledgement would raise the issue first.
Participant unable to do the requested activity.	13. Refer to the appropriate B2B Procedures for details of the appropriate Business Rejection or Response requirements.
Participant unable to issue BusinessDocuments.	14. The first level of contingency should involve the activation of backup system/service (if available). Participants should maintain at least one alternative means of raising Business Documents. 15. In the event of a failure of the primary and backup mechanism to generate a Business Document, the B2B Browser Application may be used. 16. In the event of a further failure with the B2B Browser Application, and as a last resort, the Participant may create and send a Business Document as a compressed aseXML email attachment, without password protection adhering to the requirements specified later in this Section.

9.6. Contingency Messages

(a) Participants must ensure that any Messages produced by a contingency system are normal aseXML messages and be issued pursuant to the B2B Procedures. This includes the generation of ase:Transactions, *ase:MessageAcknowledgements* and *ase:TransactionAcknowledgements*. Participants acknowledge and accept that this means Participants must have more than one method of producing aseXML messages.

9.7. Use of the B2B Browser Application as a Contingency Solution

(a) During a contingency event, a Participant must use reasonable endeavours to raise any urgent Messages via the B2B Browser Application. Participants may also utilise the B2B Browser Application for other Transactions as appropriate.

9.8. Use of Email as a Contingency Solution

(a) Participants may use email during a contingency event provided that emails are only used where the B2B Browser Application is unavailable.

(b) Any Participant moving to the usage of email as a contingency solution must notify all affected Participants.

(c) Participants must ensure that aseXML Transactions are sent as compressed attachments to an email Message.

(d) Participants must ensure that any email Message sent pursuant to and in accordance with clause 9.8~~9.8~~ is sent to the appropriate email address specified in the NEM Retail Operations Contacts List (ROCL).

(e) Participants must ensure that only one attachment is sent per email.

(f) Participants must ensure that the subject line of the email contains the file name of the attached Message, in accordance with paragraph 5.4.5 of this Procedure.

(g) Any Business Document sent by email does not require a Business Receipt. A Participant may provide an email equivalent of a receipt or an acceptance/rejection.

9.9. Use of Telephone and Fax

(a) The use of phone or fax as part of the process is detailed in the relevant B2B Procedure.

9.10. Notification and Activation Requirements

9.10.1. General requirements

(a) Activation of contingency arrangements specific to Participants is at that Participant's discretion, provided the required failover timeframes are achieved:

(i) A Participant affected by a contingency event must contact other Participants as necessary to address any operational issues associated with the outage;

(ii) A Participant affected by a contingency event must advise other Participants of the resumption of normal processes as soon as practicable after these have resumed; and

(iii) Notification to the affected Participants will be by the mechanism to be agreed and published by AEMO.

9.10.2. Customer and Site Details Notification

(a) In the case of Transactions included in the B2B Procedure Customer and Site Details Notification Process, a Participant affected by a contingency event must:

(i) Advise other Participants of system problems within 24 hours of becoming aware of the problem. Notification will be by email to the nominated addresses of affected Participants.

9.10.3. Service Orders

(a) In the case of Transactions included in the B2B Procedure Service Order Process, a Participant affected by a contingency event must:

(i) Advise other Participants of system problems within 2 hours of becoming aware of the problem; and

(ii) Provide at least twice daily updates to other Participants via the industry agreed notification process.

9.10.4. Meter Data

(a) In the case of Transactions included in the B2B Procedure Meter Data Process, a Participant affected by a contingency event must:

(i) Advise other Participants of system problems within 24 hours of becoming aware of the problem. Notification will be by email to the nominated addresses of affected Participants; and

(ii) Provide daily updates to other Participants via the notification process.

9.10.5. One Way Notification

(a) In the case of transactions included in the B2B Procedure One Way Notification Process, a participant affected by a contingency event must:

(i) Advise other Participants of system problems within 24 hours of becoming aware of the problem. Notification will be by email to the nominated addresses of affected Participants; and

(ii) Provide daily updates to other Participants via the notification process.

9.11. Prioritisation of Transactions

(a) Prioritisation of Transactions is supported by the B2B Procedures and is supported primarily as a technical requirement to differentiate "small" aseXML-based Messages from potentially "large" aseXML-wrapped CSV Messages.

(b) Retailers may choose to prioritise their Requests based on the level of automation of their contingency solution.

(i) If Retailer system failure delays delivery of Requests, the Retailer must recognise that the DNSP may not be able to meet the originally requested / regulated timeframe.

9.12. Handling of contingency Transactions once normal operations resume

(a) No duplicates Transactions are allowed. That is, if a Participant sends a Transaction via a contingency system, that Participant must not resend that Transaction using normal delivery systems (or even an alternative contingency system).