Guide to NEM Retail B2M APIs

Provides details for the B2M API interface to communicate with AEMO.

Version: **1.01**

Published: **Wednesday, 26 July 2023**

Important Notice

AEMO has prepared this Guide to NEM Retail B2M APIs to provide guidance on the use of the B2M API under the National Electricity Rules (Rules), as at the date of publication.

TRADEMARK NOTICES

Microsoft is a trademark of Microsoft Corporation in the United States and/or other countries.
Oracle and Java are registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

DOCUMENTS MADE OBSOLETE

The release of this document changes any version of the Guide to NEM Retail B2M APIs.

DISTRIBUTION

Available to the public.

DOCUMENT IDENTIFICATION

Business custodian: Business Custodian
IT custodian: IT Development
Prepared by: Technology, Technical Writers
Last update: Wednesday, 26 July 2023 9:24 AM

FURTHER INFORMATION

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

Contents

Introduction

Purpose

This Guide to NEM Retails B2M APIs, prepared by AEMO, provides guidance for B2M APIs under the National NER (Rules).

Audience

Developers of AEMO's B2M retail APIs.

What's in this guide

Chapter 1 Need to know on page 3 explains what you need to know before you start using e-Hub APIs, such as related rules, procedures, assumed knowledge, prerequisites, user rights, access, and participant implementation.

Chapter 2 About NEM Retail B2M APIs on page 7 explains e-Hub APIs what they are for, who can use them, and how to access them, submission, payload, query parameters, and response codes.

Chapter 3 B2MMessagingSync on page 14 explains the supporting Endpoints: generateC4Report, getMSATSLimits, NMIDiscovery, getNMIDetail, getParticipantSystemStatus, and getMeterData.

Chapter 4 B2MMessagingAsync on page 41 explains the supporting Endpoints submitMessages, submitMessageAcknowledgements, and getQueueMetaData.

Chapter 5 B2MMessagingPull on page 55 explains the supporting Endpoints submitMessages, submitMessageAcknowledgements, getMessages, and getQueueMetaData.

Chapter 6 HubMessageManagement on page 70 explains how to retrieve the current list of B2B and B2M stop files.

Needing Help on page 78 provides information to assist participants with IT related issues, requesting assistance from AEMO, and using the Set Participant option.

References on page 87 contains a list of resources mentioned throughout this guide.

Rules terms on page 81 explains the Rules terms used throughout this guide.

Glossary on page 82 explains the terms and abbreviations used throughout this guide.

How to use this guide

- This document is written in plain language for easy reading. Where there is a discrepancy between the Rules, and information or a term in this document, the Rules take precedence.
- Text in this format indicates a resource on **AEMO's website**.
- **Text in this format** indicates a direct link to a section in this guide.
- Glossary terms are capitalised and have the meanings listed against them in the Glossary on page 82.
- *Italicised terms* are defined in the National Electricity Rules. Any rules terms not in this format still have the same meaning.
- Actions to complete in the web portal interface are **bold and dark grey**.

Chapter 1 Need to know

Related rules and procedures

The following rules and procedures relate to B2M Retail APIs.

Resource	Location
Metering Data Management (MDM) procedures	AEMO website > Energy Systems > Electricity > National Electricity Market (NEM) > Market Operations > Retail and Metering > Market Settlement and Transfer Solutions (MSATS)
Metrology Procedures	AEMO website > Energy Systems > Electricity > National Electricity Market (NEM) > Market Operations > Retail and Metering > Metrology Procedures and Unmetered Loads
MSATS Procedures	AEMO website > Energy Systems > Electricity > National Electricity Market (NEM) > Market Operations > Retail and Metering > Market Settlement and Transfer Solutions (MSATS)
National Electricity Rules	https://www.aemc.gov.au/regulation/energy-rules/national-electricity-rules/current
Retail Electricity Market Procedures – Glossary and Framework	AEMO website >Metering procedures, guidelines and processes

Assumed knowledge

This guide assumes you have knowledge of the:

1. Metrology Procedures
2. MDM Procedures
3. MSATS Procedures
4. National Electricity Rules
5. REST architecture
6. Open API Specification (OAS)
7. aseXML standards
8. JSON or YAML schemas

Prerequisites

To use B2M API you must complete the following:

B2M APIs do not require an SSL Certificate.

1. Register with AEMO to use APIs. For help, see the API Reference. For help managing TLS certificates, see TLS Certificate Management.
2. If required, set up your Participant API Gateway. For help, see Participant implementation on the next page.
3. Build the APIs needing implementation at the Participant API Gateway according to the specifications in this guide.
4. Participant Administrators (PA) use the MSATS Web Portal to grant the required Participant User access to the User ID accessing the AEMO APIs. For help, see User rights access on the next page.

User rights access

Your company's Participant Administrator (PA) provides access to B2M API for Participant Users in the MSATS>Administration>Maintain Entities menu, using the entity documented in each Endpoint section. For example, see User Rights Access in generateC4Report on page 18.

If you don't know who your company's PA is, contact AEMO's support hub.

For more details about participant administration and user rights access, see Guide to User Rights Management.

Participant implementation

To accept messages and return acknowledgements to and from the AEMO e-Hub Gateway, participants must implement API endpoints at their gateways.

You can define your own URL and API name. The e-Hub only registers the API name of the participant.

API	Type	Use	Participant Gateway required
B2MMessagingSync	Sync	The initiator and the recipient must have bilateral agreement related to this API and have the service enabled This API does not support queuing or interoperability	Yes
B2MMessagingAsync	Push-Push	Suits high volume exchange of messages	Yes

API	Type	Use	Participant Gateway required
B2MMessagingPull	Push-Pull	Suits low volume exchange of messages Requires participants to set up polling to pull messages from the Participant Hub Queue	No
HubMessageManagement	Push-pull	Retrieves B2B and B2M stop file alerts from the Participant Hub Queue	Yes

Participant API gateway connection and read timeout settings

Timeout	Description	Value
Connection	AEMO's e-Hub is unable to connect to the participant's endpoint	max 10 secs
Read	AEMO's e-Hub is connected to the participant's endpoint but the Participant API Gateway does not respond within the configured time	max 30 secs

Chapter 2 About NEM Retail B2M APIs

What B2M Retail APIs are for

Participants use these APIs to manage the following NEM B2M retail communications:

1. Generate a C4 report
2. NMI discovery searches one, two, and three
3. MSATS limits
4. Participant system status
5. Submit messages
6. Submit message acknowledgements
7. Retrieve queue metadata
8. Get alerts for B2M and B2B stop files

Standards

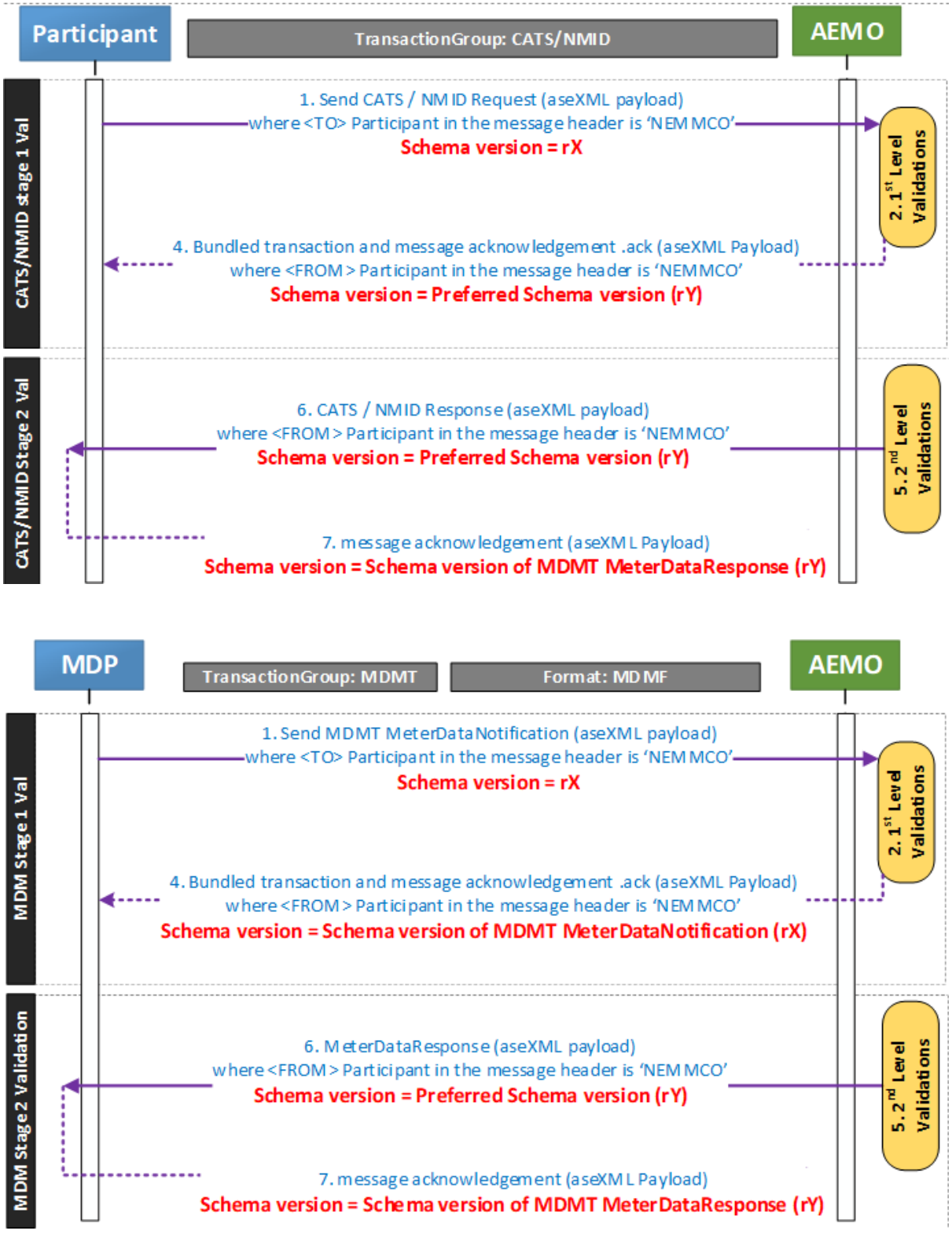
aseXML

For MSATS transactions, the architecture supports the aseXML format. aseXML defines an XML format for data exchange specific to the electricity and gas industries in Australia and is already in use, and well known to participants and AEMO.

For more information about aseXML, see aseXML Standards.

Except for the B2MMessagingSync API, the response aseXML schema version is based on the Participant ID aseXML schema transaction group (see Figure 1 on the next page) configured in the MSATS Web Portal > Participants > Participant schema. For help, see Guide to MSATS Web Portal.

Figure 1 Transaction group schema version for MACK and TACK responses



There is an optional aseXML_version parameter in the Accept header of the B2MMessagingSync API. For details, see B2MMessagingSync header parameters on page 16.

When participants update their aseXML schema version in the MSATS Web Portal, the change is reflected after 4:00 am the next day (the schema change for Batch services is immediate).

Security and authentication

To provide encrypted communication and secure identification, interactions between participant systems and AEMO are secured using HTTPS.

File size limits

- Files are limited to a size of 1 MB (uncompressed).

API URL format

```
<BaseURL><Market><API name><Version><Endpoint>
```

BaseURL	Marketnet い. Pre-production: https://apis.preprod.marketnet.net.au: 9319: ろ. Production: https://apis.prod.marketnet.net.au:9319 Internet い. Pre-production: https://apis.preprod.aemo.com.au:9319 ろ. Production: https://apis.prod.aemo.com.au:9319
Market	NEMRetail
API name	B2MmessagingSync B2MmessagingAsync B2MmessagingPull HubMessageManagement

Version	v1 or v2
Endpoint	See Endpoints for each API

Submission and payload

Payload	1 per submission
Payload size	1 MB payload limit (uncompressed)
Compression	AEMO's gateway sends responses to Participant API Gateways the same format in the request. Participants can use any of the following: 1. gzip 2. deflate
Inbound throttling limit	Set at the API level If the number of API requests exceeds the threshold limits (e.g. 1000 requests/minute), AEMO's e-Hub rejects the incoming API calls using the code 429.
Connection timeout	Cannot connect to the e-Hub's endpoint (e.g. e-Hub infrastructure is not available). Recommended settings = 10 seconds
Read timeout	Connected to the endpoint but the e-Hub does not respond within the configured time. Recommended settings = 30 seconds

Query parameters

Query string parameters must be in camel case, for example:

```
initiatingParticipantID='NEMMCO'
```

You can send multiple query parameters in the URL, for example:

```
/ehub/GeneratorRecall/v1/queues?initiatingParticipantID='NEMMCO'&
msgType='messages'
```

Response codes

Code	Explanation
200	OK/successful
400	Invalid API URL Missing header No payload Malformed JSON payload
401	Expired Participant User password Invalid Credentials No BASIC Auth information in the request header The Participant ID in the payload does not match the Participant ID for the Participant User No response payload
404	URL resource not found
405	Invalid method used. For example used GET instead of POST
422	Business validation failure
429	Exceeds throttling limits. Too many requests.
500	No payload The SSL Certificate is not associated with a valid user Service is running but the Endpoint is unavailable Internal server error

Testing a GET request

For testing purposes, you can make GET requests from a web browser by entering the URL and query parameters. The following is an example of an MSATS NMI Discovery search by meter serial GET request.

1. Enter the URL including the NMI Discovery parameters in the address bar. Substituting your participant ID for PARTICIPANTID, and the correct parameters for jurisdictionCode, transactionId, and meterSerialNumber. For example:
   https://apis.preprod.marketnet.net.au:9319/NEMRetail/B2MmessagingSync/v2/NMIDiscovery?jurisdictionCode=SA&stateOrTerritory=SA&transactionId=TX123&postcode=5087
2. In the authentication dialog box, enter your MSATS user ID and password.
3. If you have entered the GET parameters correctly the File Download dialog box displays, where you can Open or Save the file.

Chapter 3 B2MMessagingSync

The B2MMessagingSync API supports generateC4Report, getMSATSLimits, NMIDiscovery, getNMIDetail, getParticipantSystemStatus, and getMeterData endpoints.

The API participant pushes their outgoing message to the e-Hub and the e-Hub pushes the message to the API recipient.

The message delivery occurs synchronously where the participant sends the message to the e-Hub and AEMO immediately sends the response (such as MACKs, TACKs, or responses) in the same thread to the Participant's Gateway.

Participants require a gateway to use this API, see Participant implementation on page 5.

B2MMessagingSync non-functional summary

API Throttling Rate	600 pm/x-initiatingParticipantID
API Timeout	30 seconds
Bundling	not permitted
Quota	Initial (revised after performance testing) 5000 pm/x-initiatingParticipantID The allowed number of requests per minute for all B2M APIs combined
URM Throttling Rate	1000 pm/x-client certificate

B2MMessagingSync URL parameters

URL	<Base URL>/B2MMessagingSync/v2/<endpoint>
Endpoints	**generateC4Report** **NMIDiscovery** **getMSATSLimits** **getNMIDetail** **getParticipantSystemStatus**

B2MMessagingSync header parameters

Below are the common B2MMessagingSync header parameters for all endpoints.

Name	Required	Type	Example
Accept (response)	Yes	string	text/xml
Authorization	Yes	string	Two-way SSL and Basic Auth Authorization: Basic VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ== Base64 encoding of the URM username and password.
X-initiatingParticipantID	Yes	string	P01
X-market	Yes	string	NEM
X-transactionID	No	string	12345

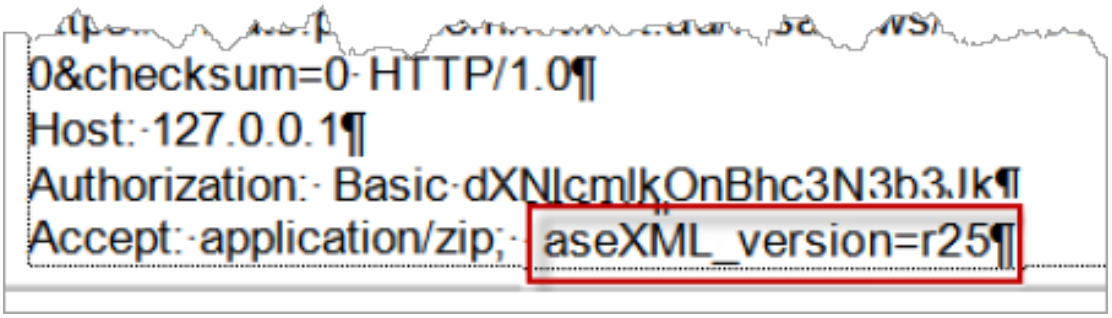
aseXML version

The output of the many B2M API requests is an aseXML response. Participants often require a specific aseXML schema version so there is an optional aseXML_version parameter in the Accept header of the B2MMessagingSync API. This parameter is ignored for other B2M APIs.

If the requested aseXML version is not supported, a 406 HTTP response code returns.

If the aseXML_version is not provided, the participant's aseXML schema for the current or superseded schema is used. You can see your participant schema version in the MSATS Web Portal > Participants > Participant Schema. For help see the Guide to MSATS Web Portal.

Figure 2 schema version parameter example



HTTPS requests

All HTTPS requests contain a method, a URL, headers, and optional file attachments. The method is either GET or POST. The HTTP response code 405 returns if the requested method is not supported.

- GET requests require all parameters in the URL.
- POST requests contain a Payload in the body or an attached file. The file format is either an .XML or a .ZIP file.
  Posted files have a size limit of 1 MB uncompressed. If the limit is exceeded, a response code of 413 returns.
- URL parameters are case sensitive.
- Request character sets are UTF-8 encoded.
- The .XML file must only contain one request.

generateC4Report

See also, B2MMessagingSync header parameters on page 16.

Description	Provides a single C4 NMI Master Report detailing one or more NMIs based on the parameters entered.
Method	GET
aseXML schema version for optional Accept header	r39 Or based on participant's configured schema version
Input	Header and query parameters
Input protocol	HTTP GET
Output	C4 NMI Master Report
Output protocol	aseXML/http
User access rights	C4 - NMI Master Report

generateC4Report query parameters

For examples, see Standing Data for MSATS.

Parameter	Required	Type
asatDate	Yes	Date
fromDate	Yes	Date

Parameter	Required	Type
nmi (case-sensitive)	Yes	string
toDate	Yes	Date
transactionId	Yes	string
inittransid	No	string
roleId	No	string

GenerateC4Report header parameters

See B2MMessagingSync header parameters on page 16.

generateC4Report request example

```
GET
'http://&lt;baseURL&gt;/NEMRetail/B2MmessagingSync/v2/generateC4Report
?initiatingParticipantID=PARTID&;NMI=4102242006&;fromDate=2019-    01-
01&;toDate=2019-11-14&;asatDate=2019-11-14' \
--header 'Accept: text/xml' \
--header 'Authorization: Basic VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ=='
```

generateC4Report response

```
<?xml version="1.0" ?>
<ase:aseXML
xmlns:ase="urn:aseXML:r35"xmlns:xsi ="http://www.w3.org/2001/XMLSchema-
instance"
xsi:
schemaLocation
="urn:aseXML:r35""http://www.nemmco.com.au/aseXML/schemas/r35/aseXML_
r35.xsd">
    <Header>
        <From    description="Australian    Energy    Market    Operator
Limited">NEMMCO</From>
        <To description="PARTID Pty Ltd">PARTID</To>
        <MessageID>NEMMCO-MSG-42187053</MessageID>
        <MessageDate>2020-01-13T21:58:39+10:00</MessageDate>
        <TransactionGroup>CATS</TransactionGroup>
        <Priority>Medium</Priority>
        <SecurityContext>NEMMCOBATCH</SecurityContext>
        <Market>NEM</Market>
    </Header>
    <Transactions>
        <Transaction                    transactionID          ="CATS-
42187053" transactionDate="2020-01-13T21:58:39+10:00">
            <ReportResponse version="r10">
                <ReportParameters
xsi:type="ase:CATSMasterReportParameters">
                    <ReportName>Master</ReportName>
                    <FromDate>2019-01-01</FromDate>
                    <ToDate>2019-11-14</ToDate>
                    <AsAtDate>2019-11-14</AsAtDate>
                    <LastSequenceNumber>0</LastSequenceNumber>
                    <NMI>4102242006</NMI>
                    <Participant>PARTID</Participant>
                    <Role>FRMP</Role>
                    <ReportType>Detailed</ReportType>
                </ReportParameters>
                <ReportResults xsi:type="ase:ReplicationReportFormat">
                    <ReplicationBlock
tableName="ElectricityNMIMaster">
                        <Row
xmlns:xsi="http://www.w3.org/2001/XMLSchema-                    instance"
xsi:type="ase:ElectricityNMIMasterRow">
                            <SequenceNumber>1888169</SequenceNumber>
                            <CreationDate>2002-                    01-
09T01:45:27+10:00</CreationDate>
                            <MaintenanceDate>9999-                 12-
31T00:00:00+10:00</MaintenanceDate>
                            <RowStatus>A</RowStatus>
                            <FromDate>2001-                        12-
22T00:00:00+10:00</FromDate>
```

```
                            <ToDate>9999-12-31T00:00:00+10:00</ToDate>
                            <NMI>4102242006</NMI>
                            <JurisdictionCode>NSW</JurisdictionCode>

<NMIClassificationCode>SMALL</NMIClassificationCode>

<TransmissionNodeIdentifier>NKU3</TransmissionNodeIdentifier>

<DistributionLossFactorCode>JLDL</DistributionLossFactorCode>
                            <Address>
                                <StructuredAddress>
                                    <House>
                                        <HouseNumber>227</HouseNumber>
                                    </House>
                                    <Street>
                                        <StreetName>NEW
ENGLAND</StreetName>
                                        <StreetType>HWY</StreetType>
                                    </Street>
                                </StructuredAddress>

<SuburbOrPlaceOrLocality>RUTHERFORD</SuburbOrPlaceOrLocality>

<StateOrTerritory>NSW</StateOrTerritory>
                                <PostCode>2320</PostCode>

<DeliveryPointIdentifier>57901720</DeliveryPointIdentifier>
                            </Address>
                            <Aggregate>Yes</Aggregate>
                            <Status>A</Status>

<CustomerClassificationCode>RESIDENTIAL</CustomerClassificationCode>

<CustomerThresholdCode>LOW</CustomerThresholdCode>
                        </Row>
                </ReportResults>
                <Event severity="Information">
                    <Code>0</Code>
                    <Explanation>Success</Explanation>
                </Event>
            </ReportResponse>
        </Transaction>
    </Transactions>
</ase:aseXML>
```

NMIDiscovery

See also, B2MMessagingSync header parameters on page 16.

Description	Provides a list of NMIs matching the search criteria for the 3 types of NMI Discovery search: 1. Delivery point identifier (DPID) 2. Meter serial 3. Address For details about NMI Discovery search criteria, see **MSATS-CATS Hints and Tips and NMI Discovery.**
Method	GET
aseXML schema version for optional Accept header	r39 Or based on participant's configured schema version
Input	Header and query parameters
Input protocol	HTTP GET
Output	aseXML NMI Discovery Request
Output protocol	aseXML/http
User access rights	NMI Discovery

NMIDiscovery query parameters

For examples, see Standing Data for MSATS.

Parameter	Required	Type
jurisdictionCode	Yes	string
transactionId	Yes	string
meterSerialNumber	If searching by meter serial number	string
deliveryPointIdentifier	If searching by DPID	string
stateOrTerritory	If searching by address	string
locationDescriptor	No	string
lotNumber	No	string
flatOrUnitType	No	string
flatOrUnitNumber	No	string
postcode	No	string
floorOrLevelNumber	No	string
houseNumber	No	string
houseNumberSuffix	No	string
streetName	No	string

Parameter	Required	Type
streetSuffix	No	string
streetType	No	string
suburbOrPlaceOrLocality	No	string
buildingOrPropertyName	No	string
floorOrLevelType	No	string

NMIDiscovery header parameters

See B2MMessagingSync header parameters on page 16.

NMIDiscovery request example

```
curl --location --request GET
'http://baseURL/NEMRetail/B2MmessagingSync/v2/NMIDiscovery/?jur
isdictionCode=SA&stateOrTerritory=SA&transactionId=TX123&postco
de=5087' \
--header 'Accept: text/xml' \
--header 'Content-Type: text/xml' \
--header 'Authorization: Basic
VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ=='
```

NMIDiscovery response

```
<?xml version="1.0" ?>
<ase:aseXML                                    xmlns:ase="urn:aseXML:r35"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-                    instance"
xsi:schemaLocation="urn:aseXML:r35
http://www.nemmco.com.au/aseXML/schemas/r35/aseXML_r35.xsd">
    <Header>
        <From    description="Australian    Energy    Market    Operator
Limited">NEMMCO</From>
        <To description="PARTID Pty Ltd">PARTID</To>
        <MessageID>NEMMCO-MSG-730445523</MessageID>
        <MessageDate>2020-01-13T21:38:45+10:00</MessageDate>
        <TransactionGroup>NMID</TransactionGroup>
        <Priority>High</Priority>
        <SecurityContext>NEMMCOBATCH</SecurityContext>
        <Market>NEM</Market>
    </Header>
    <Transactions>
        <Transaction                      transactionID          ="NMID-
730445523" transactionDate="2020-                                    01-
13T21:38:45+10:00" initiatingTransactionID="TX123">
            <NMIDiscoveryResponse version="r17">
                <NMIStandingData
xsi:type="ase:ElectricityStandingData" version="r35">
                    <NMI checksum="6">SASMPL0205</NMI>
                    <MasterData>
                        <Address>
                            <StructuredAddress>
                                <House>
                                    <HouseNumber>9</HouseNumber>
                                </House>
                                <Street>
                                    <StreetName>COLE</StreetName>
                                    <StreetType>ST</StreetType>
                                </Street>
                            </StructuredAddress>

<SuburbOrPlaceOrLocality>KLEMZIG</SuburbOrPlaceOrLocality>
                            <StateOrTerritory>SA</StateOrTerritory>
                            <PostCode>5087</PostCode>
                        </Address>
                    </MasterData>
                    <RoleAssignments>
                        <RoleAssignment>
                            <Party>UMPLP</Party>
                            <Role>LNSP</Role>
                        </RoleAssignment>
                    </RoleAssignments>
                </NMIStandingData>
                <Event severity="Information">
```

```
                    <Code>0</Code>
                </Event>
            </NMIDiscoveryResponse>
        </Transaction>
    </Transactions>
</ase:aseXML>
```

getNMIDetail

See also, B2MMessagingSync header parameters on page 16.

Description	Provides details of a specified NMI to authorised users. It is the same criteria as a NMI Discovery type 2. For details about NMI Detail search criteria, see **MSATS-CATS Hints and Tips and NMI Discovery**.
Method	GET
aseXML schema version for optional Accept header	r39 Or based on participant's configured schema version
Input	Header and query parameters
Input protocol	HTTP GET
Output	aseXML NMI Standing Data Request
Output protocol	aseXML/http
User access rights	NMI Discovery

getNMIDetail query parameters

For examples, see Standing Data for MSATS.

Parameter	Required	Type
NMI (case-sensitive)	Yes	string

Parameter	Required	Type
checksum	Yes	string
transactionId	Yes	string
type	No	string
reason	No	string

getNMIDetail header parameters

See B2MMessagingSync header parameters on page 16.

getNMIDetail request example

```
curl --location --request GET
'http://baseURL/NEMRetail/B2MmessagingSync/v2/NMIDetail/PARTID?
transactionId=Tx13&NMI=2001108337&checksum=1' \
--header 'Accept: text/xml' \
--header 'Authorization: Basic
VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ=='
```

getNMIDetail response

```
<?xml version="1.0" ?>
                <ase:aseXML                    xmlns:ase="urn:aseXML:r35"
xmlns:xsi                              ="http://www.w3.org/2001/XMLSchema-
instance"xsi:schemaLocation="urn:aseXML:r35
http://www.nemmco.com.au/aseXML/schemas/r35/aseXML_r35.xsd">
                <Header>
                <From  description="Australian  Energy  Market  Operator
Limited">NEMMCO</From>
                <To description="PARTID Pty Ltd">PARTID</To>
                <MessageID>NEMMCO-MSG-730445527</MessageID>
                <MessageDate>2020-01-13T21:51:28+10:00</MessageDate>
                <TransactionGroup>NMID</TransactionGroup>
                <Priority>High</Priority>
                <SecurityContext>NEMMCOBATCH</SecurityContext>
                <Market>NEM</Market>
                </Header>
                <Transactions>
                <Transaction                transactionID       ="NMID-
730445527" transactionDate="2020-                                    01-
13T21:51:28+10:00" initiatingTransactionID="Tx13">
                <NMIStandingDataResponse version="r4">
                <NMIStandingData
xsi:type="ase:ElectricityStandingData" version="r35">
                <NMI>2001108337</NMI>
                <MasterData>
                <JurisdictionCode>SA</JurisdictionCode>
                <NMIClassificationCode>SMALL</NMIClassificationCode>

<TransmissionNodeIdentifier>SJP1</TransmissionNodeIdentifier>

<DistributionLossFactorCode>NLV2</DistributionLossFactorCode>
                <Address>
                <StructuredAddress>
                <FlatOrUnit>
                <FlatOrUnitType>F</FlatOrUnitType>
                <FlatOrUnitNumber>23</FlatOrUnitNumber>
                </FlatOrUnit>
                <House>
                <HouseNumber>85</HouseNumber>
                </House>
                <Street>
                <StreetName>WINDSOR</StreetName>
                <StreetType>GR</StreetType>
                </Street>
                </StructuredAddress>

<SuburbOrPlaceOrLocality>KLEMZIG</SuburbOrPlaceOrLocality>
                <StateOrTerritory>SA</StateOrTerritory>
                <PostCode>5087</PostCode>
```

```
                    </Address>
                    <Status>D</Status>

<CustomerClassificationCode>RESIDENTIAL</CustomerClassificationCode>
                    <CustomerThresholdCode>LOW</CustomerThresholdCode>
                    </MasterData>
                    <RoleAssignments>
                    <RoleAssignment>
                    <Party>ETSAMDP</Party>
                    <Role>MPC</Role>
                    </RoleAssignment>
                    <RoleAssignment>
                    <Party>UMPLP</Party>
                    <Role>LNSP</Role>
                    </RoleAssignment>
                    <RoleAssignment>
                    <Party>ETSAPMP</Party>
                    <Role>MPB</Role>
                    </RoleAssignment>
                    <RoleAssignment>
                    <Party>UMPLP</Party>
                    <Role>RP</Role>
                    </RoleAssignment>
                    <RoleAssignment>
                    <Party>ETSAMDP</Party>
                    <Role>MDP</Role>
                    </RoleAssignment>
                    </RoleAssignments>
                    <DataStreams>
                    <DataStream>
                    <Suffix>11</Suffix>
                    <ProfileName>NSLP</ProfileName>
                    <AveragedDailyLoad>17</AveragedDailyLoad>
                    <DataStreamType>Consumption</DataStreamType>
                    <Status>I</Status>
                    </DataStream>
                    </DataStreams>
                    <MeterRegister>
                    <Meter>
                    <SerialNumber>814788</SerialNumber>
                    <NextScheduledReadDate>2018-                        02-
01</NextScheduledReadDate>
                    <InstallationTypeCode>BASIC</InstallationTypeCode>
                    <Status>C</Status>
                    </Meter>
                    <Meter>
                    <SerialNumber>814788</SerialNumber>
                    <RegisterConfiguration>
                    <Register>
                    <RegisterID>1</RegisterID>
                    <NetworkTariffCode>QRSR</NetworkTariffCode>
                    <UnitOfMeasure>KWH</UnitOfMeasure>
```

```
        <TimeOfDay>ALLDAY</TimeOfDay>
        <Multiplier>1</Multiplier>
        <DialFormat>5</DialFormat>
        <Suffix>11</Suffix>
        <ControlledLoad>NO</ControlledLoad>
        <Status>C</Status>
        </Register>
        </RegisterConfiguration>
        </Meter>
        </MeterRegister>
        </NMIStandingData>
        <Event severity="Information">
        <Code>0</Code>
        </Event>
        </NMIStandingDataResponse>
        </Transaction>
        </Transactions>
    </ase:aseXML>
```

getMSATSLimits

See also, B2MMessagingSync header parameters on page 16.

Description	Provides the current status of MSATS Limits. If the participant is a member of a group then the limits are the group limits, otherwise they are the participant limits
Method	GET
aseXML schema version for optional Accept header	r39 Or based on participant's configured schema version
Input	Header and query parameters
Input protocol	HTTP GET
Output	aseXML MSATS Limits Report
Output protocol	aseXML/http
User access rights	Web Service MSATS Limits

getMSATSLimits query parameters

Parameter	Required	Type	Example
transactionId	Yes	string	CATS-730445525

getMSATSLimits header parameters

See B2MMessagingSync header parameters on page 16.

getMSATSLimits request example

```
curl --location --request GET
'http://baseURL/NEMRetail/B2MmessagingSync/v2/getMSATSLimits/PA
RTID?transactionId=Tx13&NMI' \
--header 'Accept: text/xml' \
--header 'Authorization: Basic
VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ=='
```

getMSATSLimits response

```
?xml version="1.0" ?>
<ase:aseXML                                    xmlns:ase="urn:aseXML:r35"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-                    instance"
xsi:schemaLocation="urn:aseXML:r35
http://www.nemmco.com.au/aseXML/schemas/r35/aseXML_r35.xsd">
    <Header>
        <From    description="Australian    Energy    Market    Operator
Limited">NEMMCO</From>
        <To description="PARTID Pty Ltd">PARTID</To>
        <MessageID>NEMMCO-MSG-730445520</MessageID>
        <MessageDate>2020-01-13T21:20:49+10:00</MessageDate>
        <TransactionGroup>CATS</TransactionGroup>
        <Priority>Medium</Priority>
        <SecurityContext>NEMMCOBATCH</SecurityContext>
        <Market>NEM</Market>
    </Header>
    <Transactions>
        <Transaction                      transactionID          ="CATS-
730445520" transactionDate="2020-                                    01-
13T21:20:49+10:00" initiatingTransactionID="tx234">
            <ReportResponse version="r10">
                <ReportParameters
xsi:type="ase:CATSMsatsLimitsReportParameters">
                    <ReportName>MSATSLimits</ReportName>
                </ReportParameters>
                <ReportResults
xsi:type="ase:CATSMsatsLimitsReportFormat">
                    <MsatsLimits>
                        <Participant>EASTENGY</Participant>
                        <Cr>
                            <DefaultLimit>1</DefaultLimit>
                            <UpperLimit>1</UpperLimit>
                            <ParticipantActual>0</ParticipantActual>
                            <StopFlag>Y</StopFlag>
                        </Cr>
                        <CrNotification>
                            <DefaultLimit>105</DefaultLimit>
                            <UpperLimit>105</UpperLimit>
                            <ParticipantActual>0</ParticipantActual>
                            <StopFlag>N</StopFlag>
                        </CrNotification>
                        <NsrdNotification>
                            <LowerLimit>1</LowerLimit>
                            <UpperLimit>3</UpperLimit>
                            <ParticipantActual>0</ParticipantActual>
                            <StopFlag>N</StopFlag>
                        </NsrdNotification>
                        <NsrdResponse>
                            <LowerLimit>160000</LowerLimit>
```

```
                                <UpperLimit>200000</UpperLimit>
                                <ParticipantActual>0</ParticipantActual>
                                <StopFlag>N</StopFlag>
                            </NsrdResponse>
                            <OutboxFile>
                                <LowerLimit>30</LowerLimit>
                                <UpperLimit>100</UpperLimit>
                                <ParticipantActual>105</ParticipantActual>
                                <StopFlag>Y</StopFlag>
                            </OutboxFile>
                            <Report>
                                <LowerLimit>1</LowerLimit>
                                <UpperLimit>2</UpperLimit>
                                <ParticipantActual>0</ParticipantActual>
                                <StopFlag>N</StopFlag>
                            </Report>
                        </MsatsLimits>
                    </ReportResults>
                    <Event severity="Information">
                        <Code>0</Code>
                        <Explanation>Success</Explanation>
                    </Event>
                </ReportResponse>
            </Transaction>
        </Transactions>
    </ase:aseXML>
```

getParticipantSystemStatus

See also, B2MMessagingSync header parameters on page 16.

Description	Provides feedback to participants about the status of their processing by MSATS systems (such as batch handlers), and any changes to the status of their processing by MSATS systems. The available statuses are: Running, Skipped, Stopped, and Down.
User access rights	Web Service Participant System Status
Method	GET
Input	Header and query parameters
Input protocol	HTTP GET
Output	aseXML message (current or superseded)
Output protocol	aseXML/http
aseXML schema version for optional Accept header	r39 Or based on participant's configured schema version

getParticipantSystemStatus query parameters

Parameter	Required	Type	Example
transactionId	Yes	string	CATS-730445525

getParticipantSystemStatus header parameters

See B2MMessagingSync header parameters on page 16.

getParticipantSystemStatus request example

```
curl --location --request GET
'http://baseURL/NEMRetail/B2MmessagingSync/v2/NMIDetail/?transa
ctionId=TX123 \
--header 'Accept: text/xml' \--header 'Content-Type: text/xml'
\
--header 'Authorization: Basic
VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ=='
```

getParticipantSystemStatus response

```
<?xml version="1.0" ?>
                <ase:aseXML                    xmlns:ase="urn:aseXML:r35"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-                    instance"
xsi:schemaLocation="urn:aseXML:r35
http://www.nemmco.com.au/aseXML/schemas/r35/aseXML_r35.xsd">
                <Header>
                <From  description="Australian  Energy  Market  Operator
Limited">NEMMCO</From>
                <To description="PARTID Pty Ltd">PARTID</To>
                <MessageID>NEMMCO-MSG-730445525</MessageID>
                <MessageDate>2020-01-13T21:44:40+10:00</MessageDate>
                <TransactionGroup>CATS</TransactionGroup>
                <Priority>Medium</Priority>
                <SecurityContext>NEMMCOBATCH</SecurityContext>
                <Market>NEM</Market>
                </Header>
                <Transactions>
                <Transaction                transactionID       ="CATS-
730445525" transactionDate="2020-                                   01-
13T21:44:40+10:00" initiatingTransactionID="TX123">
                <ReportResponse version="r10">
                <ReportParameters
xsi:type="ase:GenericReportParameters">
                <ReportName>ParticipantSystemStatus</ReportName>
                </ReportParameters>
                <ReportResults
xsi:type="ase:CATSParticipantSystemStatusReportFormat">
                <ParticipantSystemsStatus>
                <ParticipantSystemStatus>
                <System>Batch</System>
                <ModeType>B2B</ModeType>
                <TransactionGroups>
                <TransactionGroup>FLTS</TransactionGroup>
                <TransactionGroup>NETB</TransactionGroup>
                <TransactionGroup>NOTF</TransactionGroup>
                <TransactionGroup>OUTG</TransactionGroup>
                <TransactionGroup>HSMD</TransactionGroup>
                </TransactionGroups>
                <Box>Inbox</Box>
                <Status>Running</Status>
                <Description>B2B   .ack   files   being   processed   in
participant inbox.</Description>
                <StartTime>10/JAN/20</StartTime>
                <HeartbeatTime>13/JAN/20</HeartbeatTime>
                </ParticipantSystemStatus>
                </ReportResults>
                <Event severity="Information">
                <Code>0</Code>
                <Explanation>Success</Explanation>
```

```
        </Event>
        </ReportResponse>
        </Transaction>
        </Transactions>
    </ase:aseXML>
```

Chapter 4 B2MMessagingAsync

The B2MMessagingAsync push-push API supports inbound submitMessages, submitMessageAcknowledgements, and getQueueMetaData endpoints and suits a high volume exchange of messages.

Participants push their outgoing message to the e-Hub and the e-Hub pushes the message to the Participants Gateway.

The message delivery occurs in asynchronously, where the initiator sends the message to the e-Hub and expects the outcome (such as Message Acknowledgments (MACKs), Transaction Acknowledgements (TACKs), or responses) in a different API call.

The response, sent later, is sent to the Participants Gateway, see Participant implementation on page 5.

B2MMessagingAsync non-functional summary

API Throttling Rate	600 pm/x-initiatingParticipantID
API Timeout	30 seconds
B2MMessagingAsync outbound throttling limit (bundling)	If the number of messages in the queue are greater than the participant requested, the remaining messages are pushed to the participant in the next scheduled delivery cycle one after another.
Quota	Initial (revised after performance testing) 5000 pm/x-initiatingParticipantID The allowed number of requests per minute for all B2M APIs combined
URM Throttling Rate	1000 pm/x-client certificate

B2MMessagingAsync URL parameters

URL	<Base URL>/B2MMessagingAsync/v1/<endpoint>
Endpoints	**submitMessages** **submitMessageAcknowledgements** **getQueueMetaData**

B2MMessagingASync header parameters

Below are the common B2MMessagingASync header parameters.

Name	Required	Type	Example
X-initiatingParticipantID	Yes	string	P01
X-market	Yes	string	NEM
Content-type (request)	Yes	string	text/xml
Accept (response)	Yes	string	text/xml
Authorization	Yes	string	Two-way SSL and Basic Auth Authorization: Basic VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ== Base64 encoding of the URM username and password
X-transactionID	No	string	12345

submitMessages

See also, B2MMessagingASync header parameters on the previous page.

Description	Submit messages and transaction acknowledgements to AEMO's market systems
Method	POST
aseXML schema version	Based on participant's configured transaction group schema version
Input	A B2M aseXML message with a list of retail electricity transactions
Input protocol	xml/http
Output	An acknowledgement containing 1 MACK plus 1 TACK for each input transaction in aseXML format
Output protocol	API
User access rights	Web Service Submit Messages

Submit B2M MTRD Meter Data Notifications

Participants can use the B2B Push API or B2B Pull API with the B2BmessagingAsync/messages or B2BmessagingPull/messages endpoint to submit B2M MTRD Meter Data Notifications.

For details, see B2B SMP Technical Guide.

The B2B Sync API is not available for B2M MDMT Meter Data Notification submissions and errors if tried.

submitMessages query parameters

None

submitMessages header parameters

See B2MMessagingASync header parameters on page 43.

messageContextID	Yes	string	[TransactionGroup 0-9_a-z]{1,4} + [Priority h|m|l] + "_" + [FromParticipantID]{1,10} + "_" + [0-9_a-z]{1,18}

submitMessages request example

```
curl --location --request POST '<Base
URL>/B2MMessagingAsync/v1/submitMessages' \
--header 'X-initiatingParticipantID: PARTID' \
--header 'Content-Type: text/xml' \
--header 'Accept: application/xml' \
--header 'Authorization: Basic
VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ=='
--data-raw '
--header 'X-market: NEM'
```

submitMessages response

```
<?xml version="1.0"?>
                <ase:aseXML xmlns:ase="urn:aseXML:r38"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="urn:aseXML:r38
http://www.nemmco.com.au/aseXML/schemas/r38/aseXML_r38.xsd">
                <Header>
                <From    description="National    Electricity    Market
Management Company">NEMMCO</From>
                <To description="PARTID - LNSP">PARTID</To>
                <MessageID>NEMMCO-MSG-2875769</MessageID>
                <MessageDate>2001-                                    10-
31T13:20:10.100+10:00</MessageDate>
                <TransactionGroup>CATS</TransactionGroup>
                <Priority>Low</Priority>
                <SecurityContext>zz023</SecurityContext>
                <Market>NEM</Market>
                </Header>
                <Acknowledgements>
                <MessageAcknowledgement   initiatingMessageID ="PARTID-
MSG- 11234569" receiptID="NEMMCO- MSGR- 1234342" receiptDate="2001- 10-
31T13:20:10.050+10:00" status="Accept"/>
                <TransactionAcknowledgement
initiatingTransactionID="PARTID- TNS- 12348990" receiptID="NEMMCO- TNSR-
9904567" receiptDate="2001-10-31T13:20:10.070+10:00" status="Accept"/>
                </Acknowledgements>
            </ase:aseXML>
```

submitMessageAcknowledgements

See also, B2MMessagingASync header parameters on page 43.

Description	Submit messages acknowledgements to AEMO's market systems
Method	POST
aseXML schema version	Based on participant's configured transaction group schema version
Input	B2M aseXML acknowledgement containing 1 MACK
Input protocol	API
Output	HTTP response
Output protocol	API
User access rights	Web Service Submit Message Acknowledgments

submitMessageAcknowledgements query parameters

None

submitMessagesAcknowledgements header parameters

See B2MMessagingASync header parameters on page 43.

messageContextID	Yes	string	[TransactionGroup 0-9_a-z]{1,4} + [Priority h|m|l] + "_" + [FromParticipantID]{1,10} + "_" + [0-9_a-z]{1,18}

submitMessageAcknowledgements request example

```
<?xml version="1.0"?>
                <ase:aseXML                    xmlns:ase="urn:aseXML:r38"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-                    instance"
xsi:schemaLocation="urn:aseXML:r38
http://www.nemmco.com.au/aseXML/schemas/r38/aseXML_r38.xsd">
                <Header>
                <From    description="National    Electricity    Market
Management Company">NEMMCO</From>
                <To description="PARTID - LNSP">PARTID</To>
                <MessageID>NEMMCO-MSG-2875769</MessageID>
                <MessageDate>2001-                                      10-
31T13:20:10.100+10:00</MessageDate>
                <TransactionGroup>CATS</TransactionGroup>
                <Priority>Low</Priority>
                <SecurityContext>zz023</SecurityContext>
                <Market>NEM</Market>
                </Header>
                <Acknowledgements>
                <MessageAcknowledgement
initiatingMessageID="PARTICIPANTID- 11234569" receiptID="NEMMCO- MSGR-
1234342" receiptDate="2001-10-31T13:20:10.050+10:00" status="Accept"/>
                </Acknowledgements>
            </ase:aseXML>
```

submitMessageAcknowledgements response

```
HTTP/1.1 200 OK
```

submitMessages participant implementation

To push messages to the AEMO e-Hub, participants must implement the following resources and methods on their gateway.

Endpoint	Method	Description	Payload
submitMessages	POST	Accept aseXML payload. Respond with a 200 OK and a mandatory aseXML MACK payload. If the response code is not 200, the message is re-delivered	aseXML Transaction Acknowledgement Message where status = Accept, Partial, or Reject Only one aseXML message payload is allowed in the HTTP request

submitMessages query parameters

None

submitMessages header attributes

Attribute	Required	Example
API URL	Yes	As provided by the participant POST <participant-provided URL>getMessages
messageContextID	Yes	[TransactionGroup 0-9_a-z]{1,4} + [Priority h|m|l] + "_" + [FromParticipantID]{1,10} + "_" + [0-9_a-z]{1,18} The messageContextID is case sensitive and required in lower case, for example:

Attribute	Required	Example
		sordl_retailer1_abcd1234 It provides: い. A contextID for the message exchange the e-Hub uses when delivering its corresponding MACK(s) ろ. Context for the returned failure messages where the incoming payload is unreadable such as, the payload is schema invalid
Authorization: Basic	Yes	VGt2nHWXH3n6jl6CIggW

submitMessages participant response parameters

Parameter	Validation	Payload Validation	Example
Appropriate response code	No	n/a	404 when resource name is invalid 405 when method is invalid
aseXML TACK, or Response payload	Yes	Yes	200 OK
MACK payload	Yes	No	aseXML message acknowledgment a Reject status

getQueueMetaData

See also, B2MMessagingASync header parameters on page 43.

Description	Retrieve details of messages queued in the Participant Hub Queue
Method	GET
aseXML schema version	Based on participant's configured transaction group schema version
Input	header and query parameters
Input protocol	API
Output	aseXML HubQueueReport
Output protocol	API
User access rights	Web Service Get Queue Meta Data

getQueueMetaData query parameters

Parameter	Required	Type	Example
transactionGroup	No	string	CATS, NMID, MDMT or null

getQueueMetaData header parameters

See B2MMessagingASync header parameters on page 43.

getQueueMetaData request example

```
curl --location --request GET '<Base
URL>/B2MMessagingAsync/v1/getQueueMetaData' \
--header 'X-initiatingParticipantID: PARTID'
--header 'Authorization: Basic
VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ=='
--header 'X-market: NEM'
```

getQueueMetaData response

```
<?xml version="1.0" ?>
                    <ase:aseXML      xsi:schemaLocation="urn:aseXML:r37
http://www.nemmco.com.au/aseXML/schemas/r37/aseXML_            r37.xsd"
xmlns:ase="urn:aseXML:r37"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                    <Header>
                    <From   description="National   Electricity   Market
Management Company">NEMMCO</From>
                    <To description="RECIPIENT Pty Ltd">RECIPIENT</To>
                    <MessageID>B2B-                    180319103024920-
b28a8b8d92ce4d6c</MessageID>
                    <MessageDate>2019-                                03-
18T10:30:24.920+10:00</MessageDate>
                    <TransactionGroup>HMGT</TransactionGroup>
                    <Priority>Low</Priority>
                    <Market>NEM</Market>
                    </Header>
                    <Transactions>
                    <Transaction   transactionID="B2B- 180319103024920-
b28a8b8d92ce4d6c" transactionDate="2019-03-18T10:30:24.920+10:00">
                    <HubQueueReport version="r37">
                    <ResultCount>2</ResultCount>
                    <MessageDetails>
                    <MessageMetaData>
                    <TransactionGroup>CATS</TransactionGroup>
                    <Priority>Medium</Priority>
                    <FromParticipantID>NEMMCO</FromParticipantID>
                    <MessageID>939084340846-SOMW-4</MessageID>
                    <MessageType>Transaction Message</MessageType>
                    <MessageContextID>catsm_                RECIPIENT_
730393109</MessageContextID>
                    <ReceivedDateTime>2019-                           02-
20T15:37:50</ReceivedDateTime>
                    </MessageMetaData>
                    <MessageMetaData>
                    <TransactionGroup>NMID</TransactionGroup>
                    <Priority>Medium</Priority>
                    <FromParticipantID>NEMMCO</FromParticipantID>
                    <MessageID>939084340846-SOMW-4</MessageID>
                    <MessageType>Transaction
Acknowledgement</MessageType>
                    <MessageContextID>catsl_                RECIPIENT_
730391291</MessageContextID>
                    <ReceivedDateTime>2019-                           02-
20T15:37:50</ReceivedDateTime>
                    </MessageMetaData>
                    </MessageDetails>
                    </HubQueueReport>
                    </Transaction>
```

```
</Transactions>
</ase:aseXML>
```

Chapter 5
B2MMessagingPull

The B2MMessagingPull push-pull API supports inbound submitMessages, submitMessageAcknowledgements, getMessages, and getQueueMetaData endpoints and suits low volume exchange of messages.

Participants using B2MMessagingPull are responsible for implementing the polling logic to poll their Participant Hub Queue to retrieve new messages, similar to batch programs used to poll their outbox using FTP. For details about the Participant Hub Queue, see Guide to MSATS Web Portal.

Implementation of a Participant API Gateway is **not** required for B2MMessagingPull but the speed of message delivery is slower compared to the B2BMessagingAsync API. The response is sent in the calling thread.

B2MMessagingPull non-functional summary

API Throttling Rate	600 pm/x-initiatingParticipantID
API Timeout	30 seconds
Bundling	Not permitted
Quota	Initial (revised after performance testing) 5000 pm/x-initiatingParticipantID The allowed number of requests per minute for all B2M APIs combined
URM Throttling Rate	1000 pm/x-client certificate

B2MMessagingPull URL parameters

URL	<Base URL>/B2MMessagingPull/v1/<endpoint>
Endpoints	submitMessages submitMessageAcknowledgements getMessages getQueueMetaData

B2MMessagingPull header parameters

Below are the common B2MMessagingPull header parameters.

Name	Required	Type	Example
Accept (response)	Yes	string	text/xml (default) application/xml
Authorization	Yes	string	Two-way SSL and Basic Auth Authorization: Basic VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ== Base64 encoding of the URM username and password
messageContextID	Yes	string	[TransactionGroup 0-9_a-z]{1,4} + [Priority h|m|l] + "_" + [FromParticipantID]{1,10} + "_" + [0-9_a-z]{1,18}
X-initiatingParticipantID	Yes	string	P01
X-market	Yes	string	NEM
X-transactionID	No	string	12345

submitMessages

See also, B2MMessagingPull header parameters on the previous page.

Description	Submit messages and transaction acknowledgements to AEMO's market systems
Method	POST
aseXML schema version	Based on participant's configured transaction group schema version
Input	A B2M aseXML message with a list of retail electricity transactions
Input protocol	API
Output	n/a
Output protocol	API
User access rights	Web Service Submit Messages

submitMessages query parameters

None

submitMessages header parameters

See B2MMessagingPull header parameters on the previous page.

submitMessages request example

```
curl --location --request POST '<Base
URL>/B2MMessagingAsync/v1/submitMessages' \
--header 'X-initiatingParticipantID: PARTID' \
--header 'Content-Type: text/xml' \
--header 'Accept: application/xml' \
--header 'Authorization: Basic
VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ==' 
--data-raw '
--header 'X-market: NEM'
```

submitMessages response

```
<?xml version="1.0"?>
                <ase:aseXML xmlns:ase="urn:aseXML:r38"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="urn:aseXML:r38
http://www.nemmco.com.au/aseXML/schemas/r38/aseXML_r38.xsd">
                <Header>
                <From    description="National    Electricity    Market
Management Company">NEMMCO</From>
                <To description="PARTID - LNSP">PARTID</To>
                <MessageID>NEMMCO-MSG-2875769</MessageID>
                <MessageDate>2001-                                    10-
31T13:20:10.100+10:00</MessageDate>
                <TransactionGroup>CATS</TransactionGroup>
                <Priority>Low</Priority>
                <SecurityContext>zz023</SecurityContext>
                <Market>NEM</Market>
                </Header>
                <Acknowledgements>
                <MessageAcknowledgement   initiatingMessageID ="PARTID-
MSG- 11234569" receiptID="NEMMCO- MSGR- 1234342" receiptDate="2001- 10-
31T13:20:10.050+10:00" status="Accept"/>
                <TransactionAcknowledgement
initiatingTransactionID="PARTID- TNS- 12348990" receiptID="NEMMCO- TNSR-
9904567" receiptDate="2001-10-31T13:20:10.070+10:00" status="Accept"/>
                </Acknowledgements>
            </ase:aseXML>
```

submitMessageAcknowledgements

See also, B2MMessagingPull header parameters on page 57.

Description	Submit messages acknowledgements to AEMO's market systems
Method	POST
aseXML schema version	n/a
Input	B2M aseXML acknowledgement containing 1 MACK
Input protocol	API
Output	HTTP response
Output protocol	API
User access rights	Web Service Submit Message Acknowledgments

submitMessageAcknowledgements query parameters

None

submitMessagesAcknowledgements header parameters

See B2MMessagingPull header parameters on page 57.

submitMessageAcknowledgements request example

```
<?xml version="1.0"?>
                <ase:aseXML                    xmlns:ase="urn:aseXML:r38"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-                    instance"
xsi:schemaLocation="urn:aseXML:r38
http://www.nemmco.com.au/aseXML/schemas/r38/aseXML_r38.xsd">
                <Header>
                <From    description="National    Electricity    Market
Management Company">NEMMCO</From>
                <To description="PARTID - LNSP">PARTID</To>
                <MessageID>NEMMCO-MSG-2875769</MessageID>
                <MessageDate>2001-                                      10-
31T13:20:10.100+10:00</MessageDate>
                <TransactionGroup>CATS</TransactionGroup>
                <Priority>Low</Priority>
                <SecurityContext>zz023</SecurityContext>
                <Market>NEM</Market>
                </Header>
                <Acknowledgements>
                <MessageAcknowledgement
initiatingMessageID="PARTICIPANTID- 11234569" receiptID="NEMMCO- MSGR-
1234342" receiptDate="2001-10-31T13:20:10.050+10:00" status="Accept"/>
                </Acknowledgements>
            </ase:aseXML>
```

submitMessageAcknowledgements response

```
HTTP/1.1 200 OK
```

getMessages

See also, B2MMessagingPull header parameters on page 57.

Description	Retrieve queued messages
Method	GET
aseXML schema version	Based on participant's configured transaction group schema version
Input	Header and query parameters
Input protocol	API
Output	If the messageContextID is provided, a specific queue message, otherwise the oldest message in the queue.
Output protocol	API
User access rights	Web Service Get Messages

getMessages query parameters

Parameter	Required	Type	Example
messageContextID	No If not provided the TransactionGroup must be provided	string	[TransactionGroup 0-9_a-z]{1,4} + [Priority h|m|l] + "_" + [FromParticipantID] {1,10} + "_" + [0-9_a-z]{1,18}
transactionGroup	Required if the messageContextID is **not** provided: い. If the transactionGroup is populated the resource responds with the matching message from the oldest to newest in the queue (FIFO) ろ. If an invalid TransactionGroup is passed, the API sends a HTTP response code of 500 stating the transaction group is invalid	string	MDMT, CATS, MDMT, NMID

getMessages header parameters

See B2MMessagingPull header parameters on page 57.

getMessages request example

```
curl --location --request GET
'<BaseURL>/NEMRetail/B2MMessagingPull/v1/getMessages' \
--header 'X-initiatingParticipantID: PARTID' \
--header 'Accept: application/xml' \
```

getMessages response

200 OK	asexML Message or Transaction Acknowledgement (TACK)

getQueueMetaData

See also, .B2MMessagingPull header parameters on page 57.

Description	Retrieve details of messages queued in the Participant Hub Queue
Method	GET
aseXML schema version	Based on participant's configured transaction group schema version
Input	header and query parameters
Input protocol	API
Output	aseXML HubQueueReport
Output protocol	API
User access rights	Web Service Get Queue Meta Data

getQueueMetaData query parameters

Parameter	Required	Type	Example
transactionGroup	No	string	CATS, NMID, MDMT or null

getQueueMetaData header parameters

See B2MMessagingPull header parameters on page 57.

getQueueMetaData request example

```
curl --location --request GET '<Base
URL>/B2MMessagingAsync/v1/getQueueMetaData' \
--header 'X-initiatingParticipantID: PARTID'
--header 'Authorization: Basic
VEVTVEVBU1RFTkdZOlRlc3RAMjAxOQ=='
--header 'X-market: NEM'
```

getQueueMetaData response

```
<?xml version="1.0" ?>
                    <ase:aseXML      xsi:schemaLocation="urn:aseXML:r37
http://www.nemmco.com.au/aseXML/schemas/r37/aseXML_            r37.xsd"
xmlns:ase="urn:aseXML:r37"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                    <Header>
                    <From   description="National   Electricity   Market
Management Company">NEMMCO</From>
                    <To description="RECIPIENT Pty Ltd">RECIPIENT</To>
                    <MessageID>B2B-                    180319103024920-
b28a8b8d92ce4d6c</MessageID>
                    <MessageDate>2019-                                03-
18T10:30:24.920+10:00</MessageDate>
                    <TransactionGroup>HMGT</TransactionGroup>
                    <Priority>Low</Priority>
                    <Market>NEM</Market>
                    </Header>
                    <Transactions>
                    <Transaction   transactionID="B2B- 180319103024920-
b28a8b8d92ce4d6c" transactionDate="2019-03-18T10:30:24.920+10:00">
                    <HubQueueReport version="r37">
                    <ResultCount>2</ResultCount>
                    <MessageDetails>
                    <MessageMetaData>
                    <TransactionGroup>CATS</TransactionGroup>
                    <Priority>Medium</Priority>
                    <FromParticipantID>NEMMCO</FromParticipantID>
                    <MessageID>939084340846-SOMW-4</MessageID>
                    <MessageType>Transaction Message</MessageType>
                    <MessageContextID>catsm_                RECIPIENT_
730393109</MessageContextID>
                    <ReceivedDateTime>2019-                           02-
20T15:37:50</ReceivedDateTime>
                    </MessageMetaData>
                    <MessageMetaData>
                    <TransactionGroup>NMID</TransactionGroup>
                    <Priority>Medium</Priority>
                    <FromParticipantID>NEMMCO</FromParticipantID>
                    <MessageID>939084340846-SOMW-4</MessageID>
                    <MessageType>Transaction
Acknowledgement</MessageType>
                    <MessageContextID>catsl_                RECIPIENT_
730391291</MessageContextID>
                    <ReceivedDateTime>2019-                           02-
20T15:37:50</ReceivedDateTime>
                    </MessageMetaData>
                    </MessageDetails>
                    </HubQueueReport>
                    </Transaction>
```

```
</Transactions>
</ase:aseXML>
```

Chapter 6 HubMessageManagement

The HubMessageManagement push API retrieves the current list of B2B and B2M stop files. Business transactions are sent as aseXML documents carried as payloads inside the API message and transmitted over HTTPS.

For more details, see B2B SMP Technical Guide

Participants can also use this API to:

1. Ensure technical requirements required to connect to the e-Hub are validated (for example, appropriate network ports are open).
2. Ensure their systems can connect to the e-Hub using their SSL certificates and API keys.
3. Determine if AEMO's e-Hub is operational (ping-pong test).

HubMessagingManagement non-functional summary

API Throttling Rate	600 pm/x-initiatingParticipantID
API Timeout	30 seconds
Bundling	Not permitted
Quota	Initial (revised after performance testing) 5000 pm/x-initiatingParticipantID The allowed number of requests per minute for all B2M APIs combined
URM Throttling Rate	1000 pm/x-client certificate

HubMessageManagement URL parameters

URL	<Base URL>/HubMessageManagement/v2/getAlerts
Endpoints	**getAlerts**

getAlerts

Description	Receive B2B and B2M stop file alerts
Method	GET
aseXML schema version	r37
Input	Header and query parameters
Input protocol	xml/https
Output	aseXML HubFlowControlReport
Output protocol	xml/https
User access rights	Web Service Get Flow Control Status

getAlerts query parameters

Parameter	Required	Type	Example
alertType	No If not provided, the default is B2BStopFile AEMO's API Gateway determines the endpoint depending on the alertType header parameter: B2BStopFile or B2MStopFile	string	B2BStopFile or B2MStopFile
queryParticipantID	No	string	P01

Parameter	Required	Type	Example
	Used for B2B alerts only. This value is ignored for B2M alerts.		

getAlerts header parameters

Name	Required	Type	Example
X-initiatingParticipantID	Yes	string	P01
X-transactionID	Yes	string	12345

getAlerts request example

Not applicable

getAlerts response

```
<?xml version="1.0"?>
<ase:aseXML xsi:schemaLocation="urn:aseXML:r37
http://www.nemmco.com.au/aseXML/schemas/r37/aseXML_ r37.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance"
xmlns:ase="urn:aseXML:r37">
<Header>
<To description="RECIPIENT Pty Ltd">RECIPIENT</To>
<From description="National Electricity Market
Management Company">NEMMCO</From>
<MessageID>B2M- 170419093023143-
4379b71c84da487c</MessageID>
<MessageDate>2019- 04-
17T09:30:23.144+10:00</MessageDate>
<TransactionGroup>HMGT</TransactionGroup>
<Priority>Low</Priority>
<Market>NEM</Market>
</Header>
<Transactions>
<Transaction transactionID="B2B- 170419093023143-
4379b71c84da487c" transactionDate="2019-04-17T09:30:23.144+10:00">
<HubFlowControlReport version="r37">
<RequestParameters>
<Parameter>
<Name>initiatingParticipantID</Name>
<Value>RECIPIENT</Value>
</Parameter>
</RequestParameters>
<ResultCount>32</ResultCount>
<FlowControlStandingData>
<FlowControlAlert>
<AlertType>B2MStopFile</AlertType>
<ParticipantID>RECIPIENT</ParticipantID>
<StopfileName>HOLDINP.STP</StopfileName>
<Cause>CR NOTIFICATION</Cause>
<StopDateTime>2018- 08-
17T09:42:37.304+10:00</StopDateTime>
</FlowControlAlert>
<FlowControlAlert>
<AlertType>B2MStopFile</AlertType>
<ParticipantID>RECIPIENT</ParticipantID>
<StopfileName>HOLDINP.STP</StopfileName>
<Cause>OUTBOX FILE</Cause>
<StopDateTime>2018- 02-
20T15:25:13.120+10:00</StopDateTime>
</FlowControlAlert>
<FlowControlAlert>
<AlertType>B2MStopFile</AlertType>
```

```
                <ParticipantID>RECIPIENT</ParticipantID>
                <StopfileName>HOLDINP.STP</StopfileName>
                <Cause>NSRD RESPONSE</Cause>
                <StopDateTime>2018-                                    02-
20T15:24:49.111+10:00</StopDateTime>
                </FlowControlAlert>
                </FlowControlStandingData>
                </HubFlowControlReport>
                </Transaction>
                </Transactions>
            </ase:aseXML>
```

getAlerts participant implementation

To receive alerts from the AEMO e-Hub, participants must implement the following resources and methods on their gateway.

Endpoint	Method	Description	Payload
alerts	POST	Receive stop file alerts Accept aseXML payload. Respond with a 200 OK. If the response code is not 200, the message is re-delivered	aseXML schema TransactionType: HubFlowControlAlertNotification or PayloadExceptionAlert.

getAlerts participant query parameters

None

getAlerts participant header attributes

Attribute	Required	Example
API URL	Yes	As provided by the participant POST <participant-provided URL>/alerts

Attribute	Required	Example
Accept (response)	Yes	application/xml
Content-type (request)	Yes	application/xml
API-Key	No Required if participants require the e-Hub to send their API Key	Participant-provided name-value pair
Content-Length	Yes	Nnn

getAlerts participant response parameters

HTTP Response Code	Validation	Payload Validation	Example
Appropriate response code	No	n/a	404 when resource name is invalid 405 when method is invalid
200 OK	Yes	No	No Payload
500 + description stating the exception	Yes	No	No Payload

Needing Help

AEMO's support hub

Contacting AEMO's support hub

IT assistance is requested through one of the following methods:

- Support.Hub@aemo.com.au

  For non-urgent issues, normal coverage is 8:00 AM to 6:00 PM on weekdays, Australian Eastern Standard Time (AEST).

- Support.Hub@aemo.com.au

AEMO recommends participants call AEMO's support hub for all urgent issues, whether or not you have logged a call in the Customer Portal.

Information to provide

Please provide the following information when requesting IT assistance from AEMO:

- Your name
- company name

3. Participant ID

- System or application name
- Environment: production or pre-production
- Problem description

7. Screenshots

For AEMO software-related issues please also provide:

- Version of software
- Properties or log files

3. Replication Manager support dump and instance name (if Data Interchange problem)

Information to provide

Please provide the following information when requesting assistance from AEMO:

- Your name
- company name
- Participant ID
- System or application name
- Environment: production or pre-production
- Problem description
- Screenshots

For AEMO software-related issues please also provide:

- Version of software
- Properties or log files
- Replication Manager support dump and instance name (if Data Interchange problem)

Feedback

Your feedback is important and helps us improve our services and products. To suggest improvements, please contact AEMO's support hub.

Rules terms

You can find the following terms defined in the relevant **Energy rules** and procedures.

Glossary

(missing or bad snippet)

A

AEMO API Gateway
The gateway on AEMO's side providing participant communication options, accessible over the internet or MarketNet. It uses resources and methods to push messages to Participants' API Gateways .

AEST
Australian Eastern Standard Time

API
Application Programming Interface. A set of clearly defined methods of communication between various software components.

API Portal
Where you can view available APIs, manage your API Keys, and obtain OAS files.

API Protocol
An e-Hub delivery method.

aseXML
A standard developed by Australian energy industries to facilitate the exchange of information between energy industry participants using XML.

C

CSR
Certificate Signing Request is a block of encoded text given to a Certificate Authority when applying for an SSL Certificate. It also contains the Public Key to include in the certificate. Usually, a Private Key is created at the same time, making a Key Pair.

csv
Comma-separated values; a file format for exchanging data.

Curl
A command line utility used to interact with REST API endpoints.

E

e-Hub
Consists of the API Portal and the API Gateway for both electricity and gas.

EMMS
Wholesale Electricity Market Management System; software, hardware, network and related processes to implement the energy market.

Endpoint
Where the API request is sent and where the response comes from.

F

FTP
File transfer protocol; a standard network protocol used for the transfer of computer files between a client and server on a computer network.

H

Header Parameters
Parameters included in the request header.

J

JSON
Java Standard Object Notation. An agreed format for text files and data exchange. This is now used by AEMO to receive Bids and Offers and provide responses

JSON Schema
Defines the structure and content of the bidding details.

K

Key Pair
SSL uses a technique called public-key cryptography, based on the concept of a Key Pair. The Key Pair consists of encrypted Public and Private Key data. It is only possible to decrypt the Public Key with the corresponding Private Key.

M

MACK
Message Acknowledgement

MarketNet
AEMO's private network available to participants having a participant ID

Markets Portal
Web portal for access to AEMO's wholesale web-based applications.

Method
The allowed operation for a resource, e.g. GET, POST, PUT, DELETE, and so on. These operations determine whether you're reading information, creating new information, updating existing information, or deleting information.

MSATS
Retail Market Settlement and Transfer Solution

MSATS Web Portal
MSATS web-based interactive interface

MW
Megawatt

N

NACK
Negative Acknowledgement (Rejection)

NER
National Electricity Rules

O

OAS
OpenAPI specification

OpenAPI specification document
The file, either in YAML or JSON, describing your REST API. Follows the OpenAPI specification format.

P

PA
Participant Administrator who manages participant company's user access and security. The initial PA is set up by the AEMO system administrator as part of the registration process.

Parameters
Parameters are options you pass with the endpoint (such as specifying the response format or the amount returned). There are four types of parameters: header parameters, path parameters, query string parameters, and request body parameters. The different types of parameters are often documented in separate groups on the same page. Not all endpoints contain each type of parameter. See Parameters for more details.

Participant API Gateway
The interface implemented by participants where AEMO pushes messages.

Participant File Server
The publishing point from AEMO systems to participant systems. Each participant is allocated an account and access to private and public areas. Participants are responsible for interfacing with the Participant File Server. If uncollected, files are moved to the archive folder after a couple of days. If your Data Interchange environment is configured properly it automatically retrieves the missing files from the archive. Files are kept in the archive for approximately six months. AEMO's production and pre-production environments are independently operated, so each environment has its own IP address for its Participant File Server. For help, see Connection to AEMO's IT Systems.

Participant ID
Registered participant identifier

Participant User ID
The user ID you used to login to the system.

Participant Users
Set up by the company's Participant Administrator.

Path
Parameters in the path of the endpoint, before the query string (?). Path parameters are usually set off within curly braces.

Payload
The data sent by a POST request. The Payload section sits after the header.

PID
Participant ID

Pre-production
AEMO's test system available to participants

Private Key
The secret Private Key is a text file used initially to generate a Certificate Signing Request (CSR), and later to secure and verify connections.

Production
AEMO's live system

Public Key
The Public Key is included as part of your SSL certificate, and works together with your Private Key to make sure your data is encrypt Key (i.e. the certificate) can verify the digital signature is authentic without having to know the secret Private Key.

Q

Query String Parameters
Parameters in the query string of the endpoint, after the ?.

R

Request
The way information is returned from an API. In a request, the client provides a resource URL with the proper authorization to an API server. The API returns a response with the information requested.

Request Body Parameters
Parameters in the request body. Usually submitted as JSON.

Response
The information returned by an API after a request is made. Responses are usually in JSON or XML format.

Response Example
The response example shows a sample response from the request example; the response schema defines all possible elements in the response. The response example is not comprehensive of all parameter configurations or operations, but it should correspond with the parameters passed in the request example. The response lets developers know if the resource contains the information they want, the format, and how that information is structured and labeled. The description of the response is known as the response schema. The response schema documents the response in a more comprehensive, general way, listing each property that could possibly be returned, what each property contains, the data format of the values, the structure, and other details.

REST
The Representational State Transfer API architecture

S

SSL
Secure Sockets Layer, cryptographic protocol providing API communication security

Swagger file
Refers to the OpenAPI specification

T

TACK
Transaction Acknowledgement

Throttling
AEMO uses API throttling to prevent overwhelming the API Gateway.

TLS
Transport Layer Security, cryptographic protocol providing API communication security

U

Unit
Generating Unit

URM
User Rights Management; see the Guide to URM on AEMO's website

Z

zip
The file compression format used for exchanging data with AEMO.

References

In this chapter:

The resources listed in this section contain related information that may assist you.
(missing or bad snippet)

Rules, law, and government bodies

(missing or bad snippet)

Oracle

(missing or bad snippet)

AEMO's website

(missing or bad snippet)(missing or bad snippet)

Index

I

M

N

P

Q

R

S

T

U

V