

DUROM

Engineering Project

WebService Interface for Participant



Unicorn © 2023 – Unicorn Systems a.s.

Jankovcova 1037/49, CZ – 170 00 Prague 7

| Project: |  | DUROM |
| --- | --- | --- |
| Project – Subject: |  | Engineering Project |
| Document Title: |  | WebService Interface |
|  |  |  |
| Author: |  | Ondřej Koller, Libor Sluka |
|  |  |  |
| Contact: |  | E-mail: info@unicorn.com |
|  |  | Tel.: (+420) 221 400 111 |

Unicorn © 2017 – Unicorn Systems a.s.

Jankovcova 1037/49, CZ – 170 00 Prague 7

| Project: |  | DUROM |
| --- | --- | --- |
| Project – Subject: |  | Engineering Project |
| Document Title: |  | WebService Interface |
|  |  |  |
| Author: |  | Forename Surname |
|  |  |  |
| Contact: |  | E-mail: info@unicorn.com |
|  |  | Tel.: (+420) 221 400 111 |

# Objective and Scope

This document is specification for general approach that must be taken when accessing DUROM Web Service interface by a user. Web services are built on industry standard technologies. They are available on the Internet and ensure the same level of the privacy and security as DUROM Damas web site.

This document covers the following topics:

* **Web Service Description**
* **Web Service Interface**
* **Web Service Security**
* **Data Flows**
* **XSD Schemas**

# Content

1.  Objective and Scope 3

2.  Content 4

3.  Revision History 6

4.  Web Services Interface 7

4.1  Business Terms and Definitions 7

4.2  Transfer Technology 7

4.3  Data Format 8

4.4  Interface of Damas Web Services 8

4.5  Client Application Development Guideline 9

4.5.1  Development prerequisites 10

4.5.2  Production prerequisites 10

4.5.3  Best practices of the client implementation 10

4.6  SOAP 11

4.6.1  SOAP Message 11

4.7  WSDL 15

4.7.1  Synchronous Request 15

4.7.2  Asynchronous Request 17

4.7.3  Asynchronous Request State 18

4.7.4  Current Date and Time 20

5.  Web Service Security 22

5.1  Damas Security Model 22

5.2  SOAP Request Preparation 22

5.2.1  SOAP Request Description 22

5.3  SOAP Response Parsing 23

5.3.1  SOAP Response Description 23

5.4  Error Handling 24

6.  Data Flows 25

6.1  Data Flows for Data Upload 25

6.1.1  Submit Auction Bids 25

6.1.2  Submit (Internal) Domestic Schedules 27

6.1.3  Submit Generation Schedules 29

6.1.4  Submit Availability Declaration 31

6.1.5  Submit Cross-border Schedules (Nominations) 33

6.1.6  Submit Balancing Bids 36

6.1.7  Submit Ancillary Services Bids 39

6.1.8  Submit AnS Demand File 42

6.1.9  Submit AnS Contract File 44

6.2  Data Flows for Data Download 46

6.2.1  Download (Internal) Domestic Schedules 46

6.2.2  Download Cross-border Schedules (Nominations) 47

6.2.3  Download Auction Results (LT, DA, ID auctions) 49

6.2.4  Download Dispatch Orders 50

6.2.5  Download DAM Shipper to Shipper report (XML) 51

6.2.6  Download XBID Shipper to Shipper report (XML) 53

6.2.7  Download Generation Schedules 54

7.  XSD SCHEMAS 56

7.1  List of the XSD Schemas 56

7.1.1  ENTSO-E XSD Schemas 56

7.2  Description of ENTSO-E XSD Schemas 57

7.2.1  Acknowledgement Document 57

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 01.00 | 22.5.2023 | Ondřej Koller, Libor Sluka | Final version intended for Participants |
| 01.01 | 5.1.2024 | Ondřej Koller | Attached WSDL for client code generating purposes |
| 01.02 | 17.1.2024 | Ivan Maly | Attached test WSDL for client code generating purposes |
| 01.03 | 22.01.2024 | Ivan Maly | Update of description to Auction Result chapter |

# Web Services Interface

DUROM can be accessed via the DUROM website or web services interfaces. This document provides an overview and explanation of the major properties of Damas web services implementation to access DUROM.

## Business Terms and Definitions

|  |  |
| --- | --- |
| Term | Description |
| (ENTSO-E) EDI | (ENTSO-E) Electronic Data Interchange (EDI) standards  <https://www.entsoe.eu/publications/electronic-data-interchange-edi-library> |
| ENTSO-E | The European Network of Transmission System Operators, as established by Regulation (EC) n°714/2009, which currently represents 42 electricity transmission system operators (TSOs) from 35 countries |
| SOAP | Simple Object Access Protocol. Messaging protocol specification for exchanging structured information in the implementation of web services in computer networks. Its purpose is to induce extensibility, neutrality and independence. It uses XML Information Set for its message format, and relies on application layer protocols, most often Hypertext Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission |
| SSL | Secures Sockets Layer : Cryptographic protocol that provide communications security over a computer network |
| WS | Web services are applications whose logic and functions are accessible using the standard Internet protocols and data formats, such as Hypertext Transfer Protocol (HTTP) and Extensible Markup Language (XML). |
| WSDL | Web Services Description is an XML-based interface definition language that is used for describing the functionality offered by a web service |
| WSS | Web Services Security (WS-Security, WSS) is an extension to SOAP to apply security to Web services |
| XML | Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable |
| XSD | XML Schema Definition, a recommendation of the World Wide Web Consortium (W3C), specifies how to formally describe the elements in an Extensible Markup Language (XML) document |

## Transfer Technology

Web services in Damas can be used for automated data exchange or for machine-controlled data exchange. Use of this technology significantly simplifies communication between Damas and Market Participants.

The main transfer unit is a text file containing the SOAP XML message and the format of the SOAP message in Damas was designed according to the SOAP 1.2 specification recommended by W3C (<https://www.w3.org/TR/soap12-part1/>).

The supported communication protocol is HTTPS via TLS 1.2. A common authentication process, containing login name and password, is defined. Login and password details must be sent with each SOAP message so that message could be processed. All actions performed using web services are executed in Damas with permissions of the user whose credentials are provided in the SOAP message.

Credentials must be provided in form of the Username token in accordance with Web Services Security specification. For details of the Web Services Security, see [http://www.oasis-open.org/committees/tc\_home.php?wg\_abbrev=wss.](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)

## Data Format

Every web service message used in Damas consists of two parts:

* Header of the web service message
* Body of the web service message.

For all data flows designed for sending data to Damas, the XML file containing business data to be transferred is included in body of the web service message. Structure of the XML file is defined by XSD charts, which make it possible to validate semantics of the XML message. The XML files used in Damas are implemented according to the *CIM - Schedule Market Document v5r1* and *ENTSO-E code lists v54*

For successful data exchange, it is necessary to synchronize mechanism of the entity identification (Market Participant) in order to match scheduling charts. Damas uses EIC codes standardized by ENTSO-E to identify entities and their partners abroad.

## Interface of Damas Web Services

Damas web services are accessible at the following addresses:

|  |  |  |  |
| --- | --- | --- | --- |
| **Environment** | **Address** | **Protocol** | **Port** |
| Production | https://newmarkets.transelectrica.ro/usy-durom-wsendpointg01/00121002300000000000000000000100/ws | https | 443 |
| Test | <https://test.newmarkets.transelectrica.ro/usy-durom-wsendpointg01/00127002300000000000000000000100/ws> | https | 443 |

The following web service interfaces are implemented in Damas to provide communication with neighbouring systems:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **SOAP request** | **SOAP response** | **Description** |
| [**Synchronous Request**](#_Synchronous_request) | RunSynchrous  or  RunSynchronous | RunSynchrous Response  Or  RunSynchronous Response | Provides synchronous exchange of the commercial data with Damas. |
| [**Asynchronous Request**](#_Asynchronous_Request_1) | RunAsynchrous  or  RunAsynchronous | RunAsynchrous Response  Or  RunAsynchronous Response | Provides asynchronous exchange of the commercial data with Damas. |
| [**Asynchronous Request Stat**](#_Asynchronous_Request_State_1)**us** | **CheckRQResult** | **CheckRQResult Response** | Returns the status of the asynchronous request that is being processed by Damas. |
| [**Current Date and Time**](#_Current_Date_and_Time) | **GetActualDateTime** | **GetActualDateTime Response** | Returns the current system date and time that is important for automatic operations carried out by the system. |

Damas web services can be used in either synchronous or asynchronous mode:

* **Synchronous call of the web service -** Data are passed to web service via the RunSynchrous/RunSynchronous method. By performing this step, a synchronous request is established within Damas and processed, and result is returned back to client application. Output parameter of this web service is XML with structure varying for individual data streams.
* **Asynchronous call of the web service -** Data are passed to web service via the RunAsynchrous/RunAsynchronous method. The request is established within Damas as asynchronous. Output of the web service does not include processing of the established request but rather it contains only the ID of the request. This ID is used to request result of such request later on.

## Client Application Development Guideline

This chapter explains the client application implementation based on information available in the description of Damas web service interfaces.

The following two options are available for client applications to communicate with Damas Web Service interface:

* to use the SOAP standard by means of the HTTP/HTTPS protocol

To create a request in XML in compliance with the SOAP standard and to build client application capable of sending this XML as HTTPS request to web server. In addition, it is necessary to implement functionality capable of web service reply processing. Description of the web service interface also includes description of the SOAP requests and replies.

* to create proxy class based on the WSDL interface description

Description of Damas web service interface also includes description of the WSDL interface. The WSDL is XML standard that is designed to describe arbitrary web service. Current development platforms can generate source code based on the WSDL document. Result is usually class, which allows handling of the web service as object. There is no need to implement actual communication protocol as development environment does this for you. Examples of such platforms supporting this kind of functionality include Visual Studio .NET and Java.

Communication with Damas web service interface takes place via secured SSL channel; client authenticates itself using valid login and password (also referred to as **Access code 1** and **Access code 2**) identical to those used for accessing Damas web platform. If the WS client applies a method for automatic source code generation based on the WSDL document, then this code must be extended to include such functions.

### Development prerequisites

The Test Environment is intended for tests with support of both ways of the authentication:

* Username, password over HTTPS

The following prerequisites must be fulfilled prior to the starting of tests on the Train Environment:

* User accounts
  + User accounts for Train Environment will be issued by the Helpdesk as part of the standard procedure of the creating new Damas user. Results of the procedure will be an account for target authentication using HTTPS, username, password.
* User Rights for WS functionalities
  + The WS functionalities must be allowed by the Helpdesk for the newly created user accounts.

### Production prerequisites

The following prerequisites must be fulfilled prior to using the WS Interface on the Production Environment:

* User Accounts
  + The User accounts for the Production Environment will be issued by the Service as part of standard procedure of the creating new Damas user. Production Environment requires authentication using username, password over HTTPS
* User Rights for WS functionalities
  + The WS functionalities must be allowed by the Helpdesk for the newly created user accounts.

### Best practices of the client implementation

As the first step, it is best to implement a synchronous call for action - Current Date and Time (this functionality serves as simple check of the general functionality of the WS). It is also the easiest implementation of the interface to Damas Web services. It is not connected to any business, but helps solving initial technical issues, authentication problems, etc.

Recommended list of the development steps:

1. Prepare a sample SOAP request message (ideally using some suitable software tool such as SoapUI) Replace sample username and password according to chapter [5. Web Service Security](#_Web_Service_Security)
2. Modify value in tag <wsu:Expires> to be in the future, otherwise message would be rejected by server
3. Before any implementation, try to send a message using a tool e.g. SoapUI. This will ensure you have the valid SOAP message.

Once the initial step ensuring validity of the request message is completed successfully, the implementation can start with the web service client providing the same type of the message as created manually before.

The next step is the implementation of additional necessary actions (i.e. sending nominations, bids etc.).

The asynchronous call using the RunAsychrous/RunAsynchronous actions and the CheckRQResult actions can be implemented if needed.

## SOAP

The structure of the SOAP message is implemented according to the SOAP 1.2 specification recommended by W3C (<https://www.w3.org/TR/soap12-part1/>).

### SOAP Message

The SOAP message implemented in Damas consists of the SOAP header and body.

UTF-8 encoding is required for all SOAP messages passed into Damas. All outgoing messages are UTF-8 encoded as well.

The SOAP header contains information that is essential for user authorization, such as the user's login name and password.

<soap:Header>

<!-- WSS Security Header -->

</soap:Header>

The WSS Header contains security tokens necessary to authenticate sender and check message integrity. These tokens are user credentials. For details of the WSS see Chapter Web Service Security[.](#_Web_Service_Security)

The SOAP Body message includes element, which contains input/output parameter class. Element name is derived from name of the web service that is used.

<soap:Body>

<**WebServiceName** xmlns="http://markets.transelectrica.ro/wse">

**Input/Output Parameters**

</**WebServiceName**>

</soap:Body>

For details of the WSS Header, see Chapter Web Service Security.

#### Input Parameters

The parameter class defined for input parameters is given below:

<Input>

<FID>**FID**</FID>

<Parameters>

<**XXXParam** Name="**param\_name1**">**param\_val1**</**XXXParam**>

<**XXXParam** Name="**param\_name2**">**param\_val2**</**XXXParam**>

…

<**XXXParam** Name="**param\_nameN**">**param\_valN**</**XXXParam**>

</Parameters>

</Input>

The highlighted parameter shall be replaced by values according to following rules:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Description** | **Note** |
| **FID** | String | Identification of the dataflow. See chapter 6. DATA FLOWS. | Unique for each data flow. |
| **XXXParam** |  | Element name of the parameter represents its data type. For overview of the supported data types see table below. | Depends on the data flow. |
| **param\_nameX** | String | Name of the data flow input parameter | Depends on the data flow. |
| **param\_valX** | String, Number, Date | Value of the data flow input parameter | Depends on the data flow. |

List of the input parameter data types:

|  |  |  |
| --- | --- | --- |
| **Data type element name** | **Corresponding XSD type** | **Example** |
| **BooleanParam** | xs:Boolean | True |
| **DateParam** | xs:date | 2018-04-24 |
| **DateTimeParam** | xs:dateTime | 2018-04-24T09:30:10Z |
| **DecimalParam** | xs:decimal | 999.50 |
| **IntParam** | xs:int | 999 |
| **StringParam** | xs:string | TEXT |
| **XmlParam** | Any XML node tree (corresponds to <xs:any> XSD element). | Any XML node |

Elements with data flow input parameters (XXXParam) must be alphabetically ordered by their type names (that is <BooleanParam> elements come first, <DateParam> elements come second etc.).

#### Output Parameters

The parameter class defined for output parameters is given below:

<Output>

<RQID>**RQID**</RQID>

<Result>**resultXML**</Result>

<RQState>

<Code>**RQState\_Code**</Code>

<Description>**RQState\_Description**</Description>

</RQState>

</Output>

The highlighted parameter shall be replaced by values according to following rules:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Description** | **Note** |
| **RQID** | Number | Unique identification of the asynchronous request in Damas |  |
| **resultXML** | String | Contains result of the request | Depends on the data flow; see data flows description |
| **RQState\_Code,** | String | Code of the state of the request. (For list of the possible codes see Chapter SOAP CheckRQResultResponse). | For synchronous requests the RQState\_Code value is always “COMPLETED” |
| **RQState\_Description** | String | Description of the state of the request. |  |

#### ­Error Handling

Errors returned by Damas web services interface are divided into two basic groups:

* Business errors – These errors originate in business control algorithms and it express that imported business data violates business rules. This applies only to input data flows (see Chapter Data Flows for Data Upload). These errors are returned in form of the Acknowledgement as a standard output of the data flow (see Chapter [Output Parameters](#_Output_Parameters)) and therefore are not subject of this chapter.
* System errors – These errors represent non-business faults. This includes user authentication errors, bad format of the SOAP xml, input parameters etc. These errors should be handled by client applications. System errors are listed below.

Errors are distributed to the client by using <soap:Fault> element, as defined in SOAP/1.2 specification (see <https://www.w3.org/TR/soap12-part1/#soapfault> ).

Detailed information about the error is carried in the <Error> element (see example of the SOAP fault below):

<soap:Fault>

<soap:Code>

<soap:Value>**faultcode**</soap:Value>

</soap:Code>

<soap:Reason>

<soap:Text>**faultstring**</soap:Text>

</soap:Reason>

<soap:Detail>

<Error xmlns=" http://markets.transelectrica.ro/wse/xsd/errors.xsd">

<ErrID>**errID**</ErrID>

<ErrDescr>**errDescr**</ErrDescr>

<ErrXML>**errXML**</ErrXML>

</Error>

</soap:Detail>

</soap:Fault>

The highlighted parameter shall be replaced by values according to following rules:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **Description** | **Note** |
| **faultcode** | String | Code of the error as specified in SOAP/1.2. |  |
| **faultstring** | String | Description of the error as specified in SOAP/1.2. |  |
| **ErrID** | Number | Identification number of the error. |  |
| **ErrDescr** | String | Short description of the error. |  |
| **ErrXML** | XML | Additional debug information are not intended to be processed by client applications. |  |

The <e:Error> element doesn’t have to be present in the Fault message. It is present only for errors with faultcode of the “soap:Client” value or “soap:Server” value (see link <https://www.w3.org/TR/soap12-part1/#soapfault> for details on faultcode).

Errors resulting from sender's identity and message integrity checks are returned to client application according to the WSS standard (see <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>, Chapter Error Handling).

The list of the Standard Errors is as follows:

|  |  |  |
| --- | --- | --- |
| **Error ID** | **Error Description** | **Fault Code** |
| **-500** | User is not authorized for the requested data stream. | soap:Client |
| **-501** | Date is invalid. | soap:Client |
| **-510** | Data flow with requested FID does not exist. | soap:Client |
| **-513** | Invalid data flow input parameters  Note: The message will be accompanied by the list of “<parameter\_name> - <validation problem description>” for every invalid input parameter. | soap:Client |
| **-514** | Internal server error | soap:Server |
| **-517** | Asynchronous request does not exist | soap:Client |
| **-518** | Requested operation is not permitted for this data flow | soap:Client |
| **-520** | User is not authorized to access data of the another entity. | soap:Client |

## WSDL

This part of the document contains description of all web services provided by Damas as interface for automatic communication with other system. The WSDL file that can be used for client code generation is attached within this document in chapter [8. WSDL File](#_WSDL_File). The file can be used as an input for code generators (e.g. SOAP UI tool) to generate basic client code. For convenience and backwards compatibility, both Soap 1.1 and Soap 1.2 clients specifications are available.

### Synchronous Request

This web service ensures synchronous exchange of the commercial data with Damas.

#### SOAP RunSynchronous

The SOAP request format for establishing the synchronous request in Damas

POST /usy-durom-wsendpointg01/**awid**/ws HTTP/1.1

Accept-Encoding: gzip,deflate

Host: **host:port**

Content-Type: application/soap+xml;charset=UTF-8;action="http://markets.transelectrica.ro/wse/RunSynchronous"

Content-Length: **length**

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns="http://markets.transelectrica.ro/wse">

<soap:Header>

<!-- WSS Security Header -->

</soap:Header>

<soap:Body>

<RunSynchronous>

<Input>

<FID>**FID**</FID>

<Parameters>

<**XXXParam** Name="**param\_name1**">**param\_val1**</**XXXParam**>

<**XXXParam** Name="**param\_name2**">**param\_val2**</**XXXParam**>

…

<**XXXParam** Name="**param\_nameN**">**param\_valN**</**XXXParam**>

</Parameters>

</Input>

</RunSynchronous>

</soap:Body>

</soap:Envelope>

#### SOAP RunSynchronous Response

The SOAP response format with result of the synchronous request returned from Damas:

HTTP/1.1 200 OK

content-type: application/soap+xml;charset=utf-8

Content-Length: **length**

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">

<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<RunSynchrousResponse xmlns="http://markets.transelectrica.ro/wse">

<Output>

<RQID>-1</RQID>

<Result>**resultXML**</Result>

<RQState>

<Code>COMPLETED</Code>

<Description>The request is completed.</Description>

</RQState>

</Output>

</RunSynchrousResponse>

</s:Body>

</s:Envelope>

For details of the WSS Header, see Chapter [*Web Service Security*](#_Web_Service_Security). Note that <RQID> element in this case contains -1 (ID of the request is not returned for synchronous requests).

### Asynchronous Request

This Web serviceensures the asynchronous exchange of commercial data with Damas. The RunSynchronous and RunAsynchronous methods use almost identical formats of the SOAP request and response. Asynchrounous call is used for uploading larger XML files (more than 2 timeseries).

#### SOAP RunAsynchronous

The SOAP request format for establishing the asynchronous request in Damas

POST /usy-durom-wsendpointg01/**awid**/ws HTTP/1.1

Host: **host:port**

Content-Type: application/soap+xml;charset=UTF-8;action="http://markets.transelectrica.ro/wse/RunAsynchronous"

Content-Length: **length**

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:wse="http://markets.transelectrica.ro/wse">

<soap:Header>

<!-- WSS Security Header -->

</soap:Header>

<soap:Body>

<wse:RunAsynchronous>

<wse:Input>

<wse:FID>**FID**</wse:FID>

<wse:Parameters>

<wse:**XXXParam** Name="**param\_name1**">**param\_val1**</wse:**XXXParam**>

<wse:**XXXParam** Name="**param\_name2**">**param\_val2**</wse:**XXXParam**>

…

<wse:**XXXParam** Name="**param\_nameN**">**param\_valN**</wse:**XXXParam**>

</wse:Parameters>

</wse:Input>

</wse:RunAsynchronous>

</soap:Body>

</soap:Envelope>

#### SOAP RunAsynchronous Response

The SOAP response format with result of the asynchronous request returned from Damas

HTTP/1.1 200 OK

content-type: application/soap+xml;charset=utf-8

Content-Length: **length**

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">

<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<RunAsynchronousResponse xmlns="http://markets.transelectrica.ro/wse">

<Output>

<RQID>**RQID**</RQID>

<RQState>

<Code>REGISTERED</Code>

<Description>The request is registered for execution. </Description>

</RQState>

</Output>

</RunAsynchronousResponse >

</soap:Body>

</soap:Envelope>

For details of the WSS Header, see Chapter [*Web Service Security*](#_Web_Service_Security). Note that <Result> element is in this case empty (or missing); result is not available at this moment, only ID of the asynchronous request is returned (RQID). You can check request result later by calling CheckRQResult method (see Chapter [*Asynchronous Request State*](#_Asynchronous_Request_State_1) for details).

### Asynchronous Request State

This web service returns the status of the asynchronous request that is being processed in Damas. The asynchronous request is identified by request ID which can be obtained by calling RunAsynchrous/RunAsynchronous method (see Chapter [*Asynchronous Request*](#_Asynchronous_Request_1)).

#### SOAP CheckRQResult

The SOAP request format for downloading a status of the asynchronous request from Damas

POST /usy-durom-wsendpointg01/**awid**/ws HTTP/1.1

Host: **host:port**

Content-Type: application/soap+xml;charset=UTF-8;action="http://markets.transelectrica.ro/wse/CheckRQResult"

Content-Length: **length**

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:wse="http://markets.transelectrica.ro/wse">

<soap:Header>

<!-- WSS Security Header -->

</soap:Header>

<soap:Body>

<wse:CheckRQResult>

<wse:RQID>**RQID**</wse:RQID>

</wse:CheckRQResult>

</soap:Body>

</soap:Envelope>

For details of the WSS Header, see Chapter [*Web Service Security*](#_Web_Service_Security). Please note, the highlighted item RQID must be replaced with the ID of the existing asynchronous request.

#### SOAP CheckRQResultResponse

The SOAP response format with status of the asynchronous request returned from Damas

HTTP/1.1 200 OK

content-type: application/soap+xml;charset=utf-8

Content-Length: **length**

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">

<s:Body>

<wse:CheckRQResultResponse xmlns:wse="http://markets.transelectrica.ro/wse">

<wse:Output>

<wse:RQID>**RQID**</wse:RQID>

<wse:Result>**resultXML**</wse:Result>

<wse:RQState>

<wse:Code>**RQState\_Code**</wse:Code>

<wse:Description>**RQState\_Description**</wse:Description>

</wse:RQState>

</wse:Output>

</wse:CheckRQResultResponse>

</s:Body>

</s:Envelope>

For details of the WSS Header, see Chapter [*Web Service Security*](#_Web_Service_Security). Element <RQState> contains information about state of the asynchronous request. The following table contains an overview of the possible request states:

|  |  |  |
| --- | --- | --- |
| Code | Description | Note |
| **REGISTERED** | Asynchronous request is registered for execution. | <Result> element is empty; you should check request state later. |
| **COMPLETED** | Asynchronous request is completed. | <Result> element is filled with result of asynchronous request. |
| **RUNNING** | Asynchronous request is not completed. | <Result> element is empty; you should check request state later. |
| **ERROR** | Error occurred while running asynchronous request. | Internal server error occurred; in this case you should contact the system administrator. |

### Current Date and Time

This web service returns current system date and time that is important for automatic operations carried out by system.

#### SOAP GetActualDateTime

The SOAP request format for downloading current date and time from Damas

POST /usy-durom-wsendpointg01/**awid**/ws HTTP/1.1

Host: **host:port**

Content-Type: application/soap+xml;charset=UTF-8;action="http://markets.transelectrica.ro/wse/GetActualDateTime"

Content-Length: **length**

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:wse="http://markets.transelectrica.ro/wse">

<soap:Header>

<!-- WSS Security Header -->

</soap:Header>

<soap:Body>

<wse:GetActualDateTime/>

</soap:Body>

</soap:Envelope>

#### SOAP GetActualDateTime Response

The SOAP response format with current date and time returned from Damas

HTTP/1.1 200 OK

content-type: application/soap+xml;charset=utf-8

Content-Length: **length**

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">

<s:Body>

<GetActualDateTimeResponse xmlns="http://markets.transelectrica.ro/wse">

<Output>

<RQID>-1</RQID>

<Result>

<GetDateTime>

<DateTime>**DatetimeValue**</DateTime>

</GetDateTime>

</Result>

<RQState>

<Code>COMPLETED</Code>

<Description>The request is completed.</Description>

</RQState>

</Output>

</GetActualDateTimeResponse>

</s:Body>

</s:Envelope>

# Web Service Security

## Damas Security Model

A Damas user account is necessary to access Damas GUI or to use the Web Service Interface. The user account must have the following security elements assigned:

1. Username and password (also referred to as **Access Code 1** and **Access Code 2**)

The Damas web service interface security is implemented in accordance with the Web Services Security standard (see [http://www.oasis-open.org/committees/tc\_home  
.php?wg\_abbrev=wss](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)).

Based on this standard, the following security issues are addressed:

1. Transferring credentials (username and password) in the SOAP requirements;
2. Encrypted communication is ensured by the HTTPS (HTTP over SSL) protocol. Because of this fact, the SOAP requirements/responses are not encrypted further, using procedures described in the WSS specification.

## SOAP Request Preparation

In addition to web service input parameters, the SOAP request also includes authentication data of Damas user account and digital signature of the sent data.

### SOAP Request Description

The SOAP request format with user authentication information without digital signature:

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:wse="http://markets.transelectrica.ro/wse"> <soap:Header>

<wsse:Security soap:mustUnderstand="true" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"> <wsse:UsernameToken wsu:Id="**username\_token\_id**">

<wsse:Username>**access\_code\_1**</wsse:Username>

<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">**access\_code\_2**</wsse:Password>

</wsse:UsernameToken>

<wsu:Timestamp wsu:Id="**timestamp\_id**">

<wsu:Created>**creation\_time**</wsu:Created>

<wsu:Expires>**expiration\_time**</wsu:Expires>

</wsu:Timestamp>

</wsse:Security>

</soap:Header>

<soap:Body>

<wse:RunSynchronous>

<!-- Input parameters come here -->

</wse:RunSynchronous>

</soap:Body>

</soap:Envelope>

**Description of <wsse:Security> element**

According to the WSS all security tokens are included in the wsse:Security element. This element is part of the SOAP header and consists of the following items

1. Digital signature of the message (not used at the moment)
2. User authentication information (access code 1 and access code 2); and
3. Timestamp of the soap request creation and its expiration

**Description of <wsse:UsernameToken> element**

This element contains the username and password assigned to the relevant Damas user account.

|  |  |
| --- | --- |
| XML Element | Description |
| **Username** | **Access Code 1** for Damas user account |
| **Password** | **Access Code 2** for Damas user account. |
| **Password/@Type** | Type of used UsernameToken; must be always "**http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText**”. |

## SOAP Response Parsing

The server SOAP response is not signed using digital certificate. Unlike in the SOAP request, no authentication data of Damas user account are transferred in this case.

### SOAP Response Description

The SOAP response format:

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">

<s:Body>

<wse:RunSynchronousResponse xmlns:ns2="http://markets.transelectrica.ro/wse">

<!-- Output parameters come here -->

</wse:RunSynchronousResponse>

</s:Body>

</s:Envelope>

Provided example is similar to request example from Chapter SOAP Request Description. For detailed description of each element, see Chapter [SOAP Request Description.](#_SOAP_Request_Description)

The SOAP response differs from the SOAP request in following points:

Credentials are not sent back to client – element <wsse:UsernameToken> or the whole <soap:Header> element is missing.

## Error Handling

Errors resulting from sender's identity and message integrity checks are returned to client application according to the WSS standard (see <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>, Chapter Error Handling).

# Data Flows

This chapter provides description of all data flows for downloading and uploading data from/to Damas. The following chapters deal with detailed definition of the uploading/downloading data flows:

* 6.1 Data Flows for Data Upload
* [6.2](#_Data_Flows_for_1) [Data Flows for Data Download](#_Data_Flows_for_1)

Each data flow is described in detail, with input and output parameters explained.

## Data Flows for Data Upload

Delivery of the values to Damas is automatically confirmed and sender is immediately informed about processing result by an Acknowledgement Document - please see the chapter [*7.2.1 - Acknowledgment Document*](#_Acknowledgement_Document) for further information. Processing results are delivered as a single response message which includes acknowledgement to the request and results from processing the request.

### Submit Auction Bids

This service enables submitting/modifying explicit auction bids in Damas for all Market Participants registered as CBT Manager on behalf of Rights Holder and CBT Participant.

#### Description

This data flow enables uploading explicit auction bids Timeseries in the XML format. The standard *ECAN - Bid Document v4r0* is used. The auction bid Timeseries are declared for the Auction (i.e., for Delivery Day/Interval, Border direction and Capacity Contract Type).

The authorized user(s) are allowed to submit auction bids between the gate opening and the gate closure for the given auction.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | AUC\_BID\_IN |  |
| **XML** | XmlParam | XML *Bid Document* with auction bid time series. | The XML file may contain multiple Auction Bid Timeseries for one Market Participant and Auction. Each Bid Timeseries can consist of one, up to multiple time intervals depending on Capacity Contract Type of the Auction. E.g. only one time interval in case of Yearly Auction covering the whole year. 24 time intervals are expected for Daily Auction in one hour resolution. |

#### Output Parameters

Users receive an acknowledgement message as response confirming data delivery and describing processing results. Please see the chapter [*7.2.1 – Acknowledgment Document*](#_Acknowledgement_Document) for further information.

#### Business Validations

The following business validations are performed when uploading Auction Bids. An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes. The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Please see the table below.

|  |  |  |
| --- | --- | --- |
| **Validation** | **Description** | **Error text / ACK Reason code** |
| **Auction must exist** | Auction for which auction bid(s) are submitted must exist in Damas. | Given auction does not exist.  A74 |
| **Auction Bidding Gate must be open** | The auction must be in state opened for bidding. | Bidding gate for auction *auction.code* is not open.  A74 |
| **Sender must be authorized to submit bids for the subject party** | Sender must be authorized to submit auction bids on behalf of the subject party or must be the subject party. | Sender *name(EIC)* is not authorized to submit Auction Bids on behalf of Subject Party *name (EIC)*.  999 |
| **Market Participant is allowed to bid to the Auction** | Market Participant must be registered in Damas as Auction Participant for the border bid(s) are submitted for. | The Market Participant *name* is not registered as the Auction Participant for the border *auction.borderCode*.  999 |
| **Document Code must be consistent** | DocumentCode must be the same in case the auction bid has been received already. | The Auction Bid document identification for the auction *auction.code* is not consistent with the one previously stored. The previously stored is *previouslyStored*.  A51 |
| **Each bid covers the whole auction period** | The system checks for each bid separately that its time intervals cover continuously the whole auction period.  This check is not performed for Long-term auctions. | Message contains errors at the time series level.  A03 |
| **Bid time interval is consistent with Auction Definition** | The system checks for each bid separately that the resolution and structure of time intervals are consistent with the auction definition. | Bid resolution for the Yearly Auction must be P1Y.  Bid resolution for the Monthly auctions must be P1D.  Bid resolution for the Day-Ahead and Intraday Auctions must be PT60M.  A41 |
| **Bid codes are unique** | Each bid identification must be unique. | Bid Identifications must be unique for the whole set of the Auction Bids.  A55 |
| **Number of bids does not exceed parameter in bidding configuration** | The system checks that the number of bids per trader is equal to or lower than the maximum defined in Bidding Configuration Parameter. | The maximum number of bids per trader is *auction.biddingConfiguration.maximumNumberOfBidPerTrader*.  A59 |
| **Bid requested capacity according to the bidding configuration** | The system checks that the Bid Requested Amount is according to the range defined by the Biding Configuration.  This validation is skipped in case the requested capacity is 0. | The Bid Requested Capacity for the Bid with ID *list.bidIdentification* and time interval *timeinterval* must be in the range between *auction.biddingConfigurationList.minBidValue* MW and *auction.biddingConfigurationList.maxBidValue* MW.  A59 |
| **Bid requested capacity is up the level of Offered Capacity** | The system checks that the value of the Bid Requested Capacity is equal to or lower than the Offered Capacity. | The Bid with ID *list.bidIdentification* and time interval *timeinterval* exceeds the Offered Capacity.  A27 |
| **Bid Requested Price is unique** | The system checks that the Bid Requested Price is unique in case multiple bids are submitted per trader.  This validation is skipped in case the requested price is 0. | Multiple bids with the price *itemList.requestedPrice* exist for the time interval *timeinterval*.  999 |
| **Bid Requested Price according to the auction bidding configuration** | The system checks that the Bid Requested Price is according to the range defined by the Biding Configuration.  This validation is skipped in case the requested price is 0. | The Bid Requested Price for the Bid with ID *list.bidIdentification* and time interval *timeinterval* must be in the range between *auction.biddingConfiguration.minBidPrice* and *auction.biddingConfiguration.maxBidPrice*.  A59 |

### Submit (Internal) Domestic Schedules

This service enables submitting/modifying domestic schedules in Damas for all Market Participants registered as SCH Manager on behalf of Domestic BRP, Local Market Coupling Operator as a result of Day-Ahead Market, and Market Participants registered as BRP or BSP.

#### Description

This data flow enables uploading domestic schedule Timeseries in the XML format. The standard *ESS – Schedule Message v3r1* is used. The domestic schedule Timeseries are declared for the buyer and seller in the given domestic bilateral contract and sender of this domestic schedule, which enables the matching of values provided by the buyer and the seller. the Auction (i.e., for Delivery Day/Interval, Border direction and Capacity Contract Type.

The authorized user(s) are allowed to submit domestic schedules between the gate opening and the gate closure for the given delivery day. Further modifications are possible to make by authorized user(s) when the scheduling gate is open for modifications.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | DOM\_SCH\_IN |  |
| **XML** | XmlParam | XML *Schedule Message* with domestic schedule time series. | The XML file may contain multiple Domestic Schedule Timeseries provided for the buyer and seller couple. |

#### Output Parameters

Users receive an acknowledgement message as response confirming data delivery and describing processing results. Please see the chapter [*7.2.1 – Acknowledgment Document*](#_Acknowledgement_Document) for further information.

#### Business Validations

The following business validations are performed when uploading Domestic Schedules. An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes. The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Please see the table below.

|  |  |  |
| --- | --- | --- |
| **Validation** | **Description** | **Error text / ACK Reason code** |
| **Domestic Scheduling Gate is Open** | The domestic scheduling gate must be open for domestic schedules entering or modifications. | The domestic scheduling gate for the Delivery Day *deliveryDay* is closed.  A57 |
| **BRPs and BSPs cannot create Domestic Schedules for OPCOM** | In case the sender is not OPCOM, the system checks that the counterpart of the domestic BRP is not OPCOM. | Sender *senderMarketParticipant.name* (*senderMarketParticipant.EIC*) is not authorized to create Domestic Schedules with OPCOM as counterpart. |
| **Domestic Schedule is not read only** | In case of update, the system checks that the existing Domestic Schedule can be edited. | The Domestic Schedule is marked as read only and may not be edited. |
| **Time Intervals are consistent** | The system checks for each domestic schedule separately that the time interval is fully within the time interval of the Schedule Message. | Time intervals in message header and time series header must be consistent.  A04 |
| **Domestic Schedule in the past cannot be updated** | The system checks that the values provided for the already closed MTUs are not changed. | Domestic Schedule in the past cannot be updated. The MTU *itemList.timeInterval* is already closed.  A03 |
| **The buyer and seller cannot be the same party** | The system checks for each domestic schedule separately that the buyer and seller are not the same participant. | The buyer and seller must not be equal.  A22 |
| **Data are provided for correct party** | The system checks for each domestic schedule separately that the sender is equal to either buyer or seller. | The domestic schedule must be received for the sender *senderMarketParticipant.name* (*senderMarketParticipant.EIC*). Data for incorrect parties provided. Seller: *seller.name* (*seller.EIC*), Buyer: *buyer.name* (*buyer.EIC*).)  A22 |
| **Combination of buyer and seller is unique** | The system checks that the combination buyer and seller is unique. | The combination of buyer *buyer.name* (*buyer.EIC*) and seller *seller.name* (*seller.EIC*) must be unique.  A22 |
| **The seller is not virtual BRP** | The system checks for each combination of buyer and seller whether the seller is not Virtual BRP. | The seller *seller.name* (*seller.EIC*) is a virtual BRP. Domestic Schedules may be received only for non-virtual BRPs.  A22 |
| **The buyer is not Virtual BRP** | The system checks for each combination of buyer and seller whether the seller is not Virtual BRP. | The seller *buyer.name* (*buyer.EIC*) is a virtual BRP. Domestic Schedules may be received only for non-virtual BRPs.  A22 |

### Submit Generation Schedules

This service enables submitting/modifying generation schedules in Damas for all Market Participants registered as SCH Manager on behalf of providers and Market Participants registered as BRP or BSP.

#### Description

This data flow enables uploading generation schedule Timeseries in the XML format. The standard *ESS – Schedule Message v3r1* is used. The generation schedule Timeseries are declared for the power unit and its provider.

The authorized user(s) are allowed to submit generation schedules between the gate opening and the gate closure for the given delivery day. Further modifications are possible to make by authorized user(s) when the scheduling gate is open for modifications.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | GEN\_SCH\_IN |  |
| **XML** | XmlParam | XML *Schedule Message* with generation schedule time series. | The XML file may contain multiple Generation Schedule Timeseries provided for the power unit and its provider. |

#### Output Parameters

Users receive an acknowledgement message as response confirming data delivery and describing processing results. Please see the chapter [*7.2.1 – Acknowledgment Document*](#_Acknowledgement_Document) for further information.

#### Business Validations

The following business validations are performed when uploading Generation Schedules. An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes. The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Please see the table below.

|  |  |  |
| --- | --- | --- |
| **Validation** | **Description** | **Error text / ACK Reason code** |
| **Generation Scheduling Gate is Open** | The generation scheduling gate must be open for generation schedules entering or modifications. | Delivery day *deliveryDay* is closed for Day-ahead or Intraday Generation Schedule entering.  A57 |
| **Sender is authorized to submit schedules for the given power unit** | The system checks that the sender is authorized to submit generation schedules for the given power unit. | Sender *senderMarketParticipantId.name* (*senderMarketParticipantId.EIC*) is not authorized to submit generation schedules for the power unit *unitCode.unitName*.  A05 |
| **Provider is authorized to submit schedules for the given power unit** | The system checks for each generation schedule separately that the declared provider is the provider of the given power unit. | Provider *bspId.marketParticipant.name* (*bspId.marketParticipant.EIC*) is not authorized to submit generation schedules for the power unit *unitCode.unitName*.  A05 |
| **Generation Schedule in the past cannot be updated** | The system checks that the values provided for the already closed MTUs are not changed. | Generation Schedule in the past cannot be updated. The MTU *itemList.timeInterval* is already closed.  A57 |
| **Generation Schedule does not exceed the Availability Declaration** | The system checks for each power unit separately that the values of generation schedule are between availability declaration for generation and availability declaration for load. | Generation Schedule of the unit *unitCode.unitName* is not compliant with its Availability Declaration, Availability Declaration Load or both.  A65 |
| **Generation Schedule is not below the minimum** | The system checks for each power unit separately that the values of generation schedule are higher than the minimum defined for the given power unit. | Generation Schedule (*quantity*) of the unit *unitCode.unitName* is below its PminBM (*pminBM*) or PminBMLoad (*pminBMLoad*).  A65 |

### Submit Availability Declaration

This service enables submitting/modifying availability declaration schedules in Damas for all Market Participants registered as SCH Manager on behalf of providers and Market Participants registered as BRP or BSP.

#### Description

This data flow enables uploading availability declaration schedule Timeseries in the XML format. The standard *Damas type – Availability Declaration Document v6r0* is used. The availability declaration schedule Timeseries are declared for the power unit and its provider.

The authorized user(s) are allowed to submit availability declaration schedules between the gate opening and the gate closure for the given delivery day. Further modifications are possible to make by authorized user(s) when the scheduling gate is open for modifications.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | AVA\_DEC\_IN |  |
| **XML** | XmlParam | XML *Availability Declaration Document* with availability declaration schedule time series. | The XML file may contain multiple Availability Declaration Schedule Timeseries provided for the power unit and its provider. |

#### Output Parameters

Users receive an acknowledgement message as response confirming data delivery and describing processing results. Please see the chapter [*7.2.1 – Acknowledgment Document*](#_Acknowledgement_Document) for further information.

#### Business Validations

The following business validations are performed when uploading Availability Declaration Schedules. An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes. The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Please see the table below.

|  |  |  |
| --- | --- | --- |
| **Validation** | **Description** | **Error text / ACK Reason code** |
| **Availability Declaration Scheduling Gate is Open** | The availability declaration scheduling gate must be open for availability declaration schedules entering or modifications. | Delivery day *deliveryDay* is closed for Day-ahead or Intraday Availability Declaration Schedule entering.  A57 |
| **Availability Declaration Schedule in the past cannot be updated** | The system checks that the values provided for the already closed MTUs are not changed. | Availability Declaration in the past cannot be updated. The MTU *itemList.timeInterval* is already closed.  A57 |
| **Sender is authorized to submit schedules for the given power unit** | The system checks that the sender is authorized to submit generation schedules for the given power unit. | Sender is not allowed to submit Availability Declaration Request for at least one Power Unit.  A05 |
| **Power Unit is assigned to a declared BSP** | The system checks for each availability declaration schedule separately that the declared provider is the provider of the given power unit. | At least one Power Unit is not assigned to a declared BSP.  A05 |
| **Availability Declaration Schedule does not exceed the generation maximum** | The system checks for each power unit separately that the values of availability declaration schedule are equal to or lower than the generation maximum defined for the given power unit.  This validation is skipped in case the maximum is not defined or the availability declaration schedule generation value is 0. | Declaration for unit *list.unitCode.unitName* exceeds its PMaxBM (*list.unitCode.pmaxBM*).  A65 |
| **Availability Declaration Schedule does not exceed the load maximum** | The system checks for each power unit separately that the values of availability declaration schedule are equal to or lower than the load maximum defined for the given power unit.  This validation is skipped in case the maximum is not defined or the availability declaration schedule load value is 0. | Declaration for unit *list.unitCode.unitName* exceeds its PMaxBMLoad (*list.unitCode.pmaxBMLoad*).  A65 |
| **Generation Schedule does not exceed the Availability Declaration declared for generation** | The system checks for each power unit separately that the respective generation schedule is up to the level of availability declaration schedule values declared for generation.  This validation is skipped in case the availability declaration schedule for generation was not provided or the respective generation schedule does not exist yet. | The Generation Schedule ([*generationSchedule\_pun\_pt15m*](https://uuapp.plus4u.net/uu-bookkit-maing01/1f4d182ee7eb4638a98336807757f2c1/book/page?code=generationSchedule_pun_pt15m)) exceeds the updated value of Availability Declaration (*list.itemList.availabilityDeclaration*). Please adjust the Generation Schedule first before updating Availability Declaration.  A65 |
| **Generation Schedule does not exceed the Availability Declaration declared for load** | The system checks for each power unit separately that the respective generation schedule is up to the level of availability declaration schedule values declared for load.  This validation is skipped in case the availability declaration schedule for load was not provided or the respective generation schedule does not exist yet. | The Generation Schedule Load ([*generationSchedule\_pun\_pt15m*](https://uuapp.plus4u.net/uu-bookkit-maing01/1f4d182ee7eb4638a98336807757f2c1/book/page?code=generationSchedule_pun_pt15m)) exceeds the updated value of Availability Declaration (*list.itemList.availabilityDeclarationLoad*). Please adjust the Generation Schedule first before updating Availability Declaration.  A65 |

### Submit Cross-border Schedules (Nominations)

This service enables submitting/modifying cross-border schedules in Damas for all Market Participants registered as SCH Manager or BRP. SCH Manager may submit operational and compensation schedules as well as commercial schedules on behalf of Domestic BRP. Domestic BRP may submit only commercial schedules.

#### Description

This data flow enables uploading cross-border schedule Timeseries in the XML format. The standard *ESS – Schedule Message v3r1* is used. The cross-border schedule Timeseries are declared for the domestic BRP and counterpart couple for the given border direction, capacity contract type and capacity agreement identification (CAI).

The authorized user(s) are allowed to submit cross-border schedules between the gate opening and the gate closure for the given border direction, timescale, and delivery day.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | CBS\_IN |  |
| **XML** | XmlParam | XML *Schedule Message* with domestic schedule time series. | The XML file may contain multiple Cross-Border Schedule Timeseries provided for the border direction, capacity contract type, domestic BRP, counterpart and capacity agreement identification. |

#### Output Parameters

Users receive an acknowledgement message as response confirming data delivery and describing processing results. Please see the chapter [*7.2.1 – Acknowledgment Document*](#_Acknowledgement_Document) for further information.

#### Business Validations

The following business validations are performed when uploading Cross-Border Schedules. An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes. The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Please see the table below.

|  |  |  |
| --- | --- | --- |
| **Validation** | **Description** | **Error text / ACK Reason code** |
| **Cross-Border Scheduling Gate is Open** | The cross-border scheduling gate must be open for cross-border schedules entering. | The *capacityContractTypeCode.timescale* Cross-border Scheduling gate is not opened for the given Border *bordeDirectionCode.borderCode.name* and Delivery Day *deliveryDay*.  A57 |
| **Sender is authorized to submit non-commercial schedules** | In case of operational or compensation cross-border schedules, the system checks that the sender is authorized to submit this type of schedules. | Sender is not authorized to enter non-commercial schedules.  A05 |
| **Schedules for given parameters are submitted with the correct time series identification** | In case the cross-border schedule already exists for the submitted parameters, the system checks that the time series identification is equal to the previously stored one. If this validation passes, the cross-border schedule is updated. | Timeseries *list.timeSeriesIdentification* is provided for the same set of parameters as already existing timeseries *crossBorderSchedule.timeSeriesIdentification*. Update the already existing timeseries in order to change nominations.  A55 |
| **Schedules for given parameters are submitted with a higher version** | In case the cross-border schedule already exists for the submitted parameters, the system checks that the time series version is higher than the previously stored one. | Timeseries *list.timeSeriesIdentification* already exist in the system with version *crossBorderSchedule.timeSeriesVersion*, provided version is *list.timeSeriesVersion*.  A50 |
| **Capacity Contract Types belong to the same timescale** | The system checks that the submitted capacity contract type for all submitted cross-border schedules belong to the same timescale. | Capacity Contract Types must belong to the same timescale.  999 |
| **Time Intervals are consistent** | The system checks for each cross-border schedule separately that the time interval is fully within the time interval of the Schedule Message. | Time intervals in message header and time series header must be consistent.  A04 |
| **Sender is authorized to submit cross-border schedules** | If the cross-border schedules are not submitted by SCH Manager, the system checks that each declared domestic BRP is equal to the sender. | Sender which is submitting schedules is not Domestic BRP.  A05 |
| **Number of decimal places correspond with the configuration** | The system checks that all cross-border schedule values are provided with up to number of decimal places as defined in cross-border scheduling configuration. | Values for each schedule should contain maximum *decimalPlaces* decimal places.  A42 |
| **Resolution corresponds with the configuration** | The system checks that all cross-border schedule values are provided in resolution as defined in cross-border scheduling configuration. | The values of schedule must be the same within each hour.  A42 |
| **Domestic BRP has rights for given Border** | The system checks for each cross-border schedule separately that the domestic BRP is assigned to the given border. | Selected Domestic BRP has no rights to submit schedules for border *borderDirectionCode.border.name*.  A05 |
| **Counterpart has rights for given Border** | The system checks for each cross-border schedule separately that the counterpart is assigned to the given border. | Selected Counterpart has no rights to submit schedules for border *borderDirectionCode.border.name*.  A22 |
| **BRP Couple is correct** | The system checks for each cross-border schedule separately that the domestic BRP and counterpart couple is assigned to the given border and capacity contract type.  This validation is skipped in case of the M:N scheduling principle. | The BRP Couple is not defined for the Border Direction *borderDirectionCode.name*and Capacity Contract Type *capacityContractTypeCode.name*.  A58 |
| **Cross-border Schedules do not exceed transmission rights** | In case the pre-matching is performed, the system checks that the cross-border schedules do not exceed the transmission rights capacity. | Schedule for direction *list.borderDirectionCode.name* contract type *capacityContractTypeCode.name* and CAI *agreementIdentification*has exceeded transmission rights.  (warning) |

### Submit Balancing Bids

This service enables submitting/modifying balancing bids in Damas for all Market Participants registered as BSP.

#### Description

This data flow enables uploading balancing bid Timeseries in the XML format. The standard *CIM – Reserve Bid Document v7r2* is used. The balancing bid Timeseries are declared for the balancing energy product.

The authorized user(s) are allowed to submit balancing bids between the gate opening and the gate closure for the given balancing energy product and trading period.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | BAL\_BIDS\_AND\_NEEDS\_IN |  |
| **XML** | XmlParam | XML *Reserve Bid Document* with balancing bid time series. | The XML file may contain multiple Balancing Bid Timeseries provided for the balancing energy product. |

#### Output Parameters

Users receive an acknowledgement message as response confirming data delivery and describing processing results. Please see the chapter [*7.2.1 – Acknowledgment Document*](#_Acknowledgement_Document) for further information.

#### Business Validations

The following business validations are performed when uploading Balancing Bids. An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes. The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Please see the table below.

|  |  |  |
| --- | --- | --- |
| **Validation** | **Description** | **Error text / ACK Reason code** |
| **Balancing Bidding Gate is Open** | The balancing bidding gate must be open for balancing bids entering. | Gate is already closed for trading period *commonMessage.radingPeriodStart*. |
| **BSP is authorized to submit bids for the given power unit** | The system checks for each bid separately that the BSP belongs to the declared power unit. | Unit *commonMessage.unit* doesn't belong to the BSP *commonMessage.bsp*. |
| **Divisible bids are submitted for balancing energy product where divisibility is allowed** | The system checks for each bid separately that the divisibility is allowed for the given balancing energy product. | Minimum Offered Volume *commonMessage.minimumOfferedVolume* is not allowed for the Energy Product *commonMessage.product* in the Balancing Bid *commonMessage.internalId*.  Maximum Offered Volume *valuecommonMessage.maximumOfferedVolume*is not allowed for the Energy Product *commonMessage.product*. |
| **Minimum and maximum offered volume submitted for divisible bid do not exceed allowed min and max** | The system checks for each bid separately that the minimum and maximum offered volume do not exceed minimum and maximum defined for the given balancing energy product. | Minimum Offered Volume *valuecommonMessage.minimumOfferedVolume* is not allowed for the Energy Product *commonMessage.product* in the Balancing Bid *commonMessage.internalId*.  Maximum Offered Volume *valuecommonMessage.maximumOfferedVolume* is not allowed for the Energy Product *commonMessage.product*. |
| **Minimum and maximum offered price does not exceed allowed min and max** | The system checks for each bid separately that the minimum and maximum offered price does not exceed minimum and maximum defined for the given balancing energy product. | Price *commonMessage.price* value is incorrect in the Balancing Bid *commonMessage.internalId*. |
| **Currency corresponds to the default currency defined for the balancing energy product** | The system checks that the currency is equal to the currency defined for the given balancing energy product. | Currency *commonMessage.currency* is incorrect in the Balancing Bid *commonMessage.internalId* for the Energy Product *commonMessage.product*. |
| **Direction must be allowed for the balancing energy product** | The system checks for each bid separately that the submitted direction is allowed for the given balancing energy product. | Direction *commonMessage.direction* is not allowed for the Energy Product *commonMessage.product*. |
| **Activation type must be allowed for the balancing energy product** | The system checks for each bid separately that the submitted activation type is allowed for the given balancing energy product. | Activation Type *commonMessage.activationType* is not allowed for the Energy Product *commonMessage.product*. |
| **Auction must correspond with the balancing energy product** | The system checks for each bid separately that the submitted auction is equal to the auction defined for the given balancing energy product. | Auction *commonMessage.auction* is not *balancingProduct.auction*. |
| **Linked bids belong to the same balancing energy product as the current bid** | The system checks that all linked bids belong to the same balancing energy product as the current bid. | The Balancing Energy Product in the linked bid of *commonMessage.internalId* must be the same. |
| **Linked bids belong to the same BSP as the current bid** | The system checks that all linked bids belong to the same BSP as the current bid. | The BSP in the linked bid of *commonMessage.internalId* must be the same. |
| **Linked bids are submitted for the same trading period as the current bid** | The system checks that all linked bids are submitted for the same trading period as the current bid. | The trading period in the linked bid of *commonMessage.internalId* is incorrect. |
| **Linked bids must be simple for the simple current bid** | The system checks that all linked bids are simple in case the current bid is simple. | The linked bid of *commonMessage.internalId* must be also simple. |
| **Linked bids must be multipart for the multipart current bid** | The system checks that all linked bids are multipart in case the current bid is multipart. | The linked bid of *commonMessage.internalId* must be also complex. |
| **Linked bids must be exclusive for the exclusive current bid** | The system checks that all linked bids are exclusive in case the current bid is exclusive. | The linked bid of *commonMessage.internalId* must be also complex. |
| **Simple linked bids must be unique** | The system checks for each simple linked bid that it is unique per trading period. | The linked bid of *commonMessage.internalId* must be linked only once. |
| **Complex linked bids must belong to the same complex group** | The system checks for complex bids that all linked bids belong to the same group per trading period. | The linked bid of *commonMessage.internalId* must be only part of one complex group. |
| **Bids linked to a multipard group must have different price** | The system checks for multipart bids that all linked bids have different price. | The Price in the linked bids of *commonMessage.internalId*must be different. |
| **Exclusive bids do not have any conditional link** | The system checks for exclusive bids that no conditional link is specified. | The Conditional Link for *commonMessage.internalId* is not allowed |
| **Power unit bids do not exceed the upward or downward limit (based on the direction)** | The system checks that the offered volume does not exceed the limit as defined for the given balancing energy product for the given power unit.  The validation is performed only if the total offered volume is higher than 0. | "Prequalified volume exceeded" |

### Submit Ancillary Services Bids

This service enables submitting/modifying AnS bids in Damas for all Market Participants registered as AnS Manager on behalf of AnS Participant or AnS Participant Writer.

#### Description

This data flow enables uploading AnS bids Timeseries in the XML format. The standard *Damas type – Reserve Bid Document v6r0* is used. The AnS bids Timeseries are declared for the Tender.

The authorized user(s) are allowed to submit AnS bids between the gate opening and the gate closure for the given tender.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | ANS\_BID\_IN |  |
| **XML** | XmlParam | XML *Reserve Bid Document* with domestic schedule time series. | The XML file may contain multiple AnS Bids Timeseries provided for one AnS Participant and Tender. |

#### Output Parameters

Users receive an acknowledgement message as response confirming data delivery and describing processing results. Please see the chapter [*7.2.1 – Acknowledgment Document*](#_Acknowledgement_Document) for further information.

#### Business Validations

The following business validations are performed when uploading AnS Bids. An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes. The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Please see the table below.

|  |  |  |
| --- | --- | --- |
| **Validation** | **Description** | **Error text / ACK Reason code** |
| **Tender must exist** | Tender for which AnS bid(s) are submitted must exist in Damas. | The tender *tenderCode* does not exist.  A74 |
| **Tender Bidding Gate must be open** | The tender must be in state opened for bidding. | The bidding gate for tender *tenderCode* is closed.  A74 |
| **Sender must be authorized to submit bids for the subject party** | Sender must be authorized to submit AnS bids on behalf of the subject party or must be the subject party. | Sender *senderMarketParticipantId.name* (*senderMarketParticipantId.EIC*) is not authorized to submit Tender Bids on behalf of Subject Party *subjectANSParticipantCode.marketParticipant.name* (*subjectANSParticipantCode.marketParticipant.EIC*).  A05 |
| **AnS Participant is not suspended** | Subject AnS Participant must not be suspended | The ANS Participant *subjectANSParticipantCode.marketParticipant.name* (*subjectANSParticipantCode.marketParticipant.EIC*) is suspended.  A05 |
| **AnS Bid is submitted for AnS service defined in the Tender** | The system checks for each bid separately that the bid is submitted for AnS service defined in the given tender. | The bid *tenderBidIdentification* was submitted for the service *serviceType* which is not included in the tender *tenderCode*.  999 |
| **AnS Participant must be authorized to submit bids for the given AnS service** | The system checks for each bid separately that the AnS Participant is allowed to submit bids for the given AnS service. | ANS Participant *subjectANSParticipantCode.marketParticipant.name* (*subjectANSParticipantCode.marketParticipant.EIC*) is not authorized to submit Tender Bids for service se*rviceType*.  A05 |
| **Number of bids does not exceed the maximum specified for the given AnS service** | The system checks that the number of bids per trader is equal to or lower than the maximum defined for the given AnS service. | The maximum number of Tender Bids per service *serviceType* is *maxTenderBidsNumber*.  A59 |
| **Indivisible bids are submitted for AnS service where indivisibility is allowed** | The system checks for each bid separately that the indivisibility is allowed for the given AnS service in the given tender. | The bid *tenderBidIdentification* is marked as indivisible. Indivisibility is not allowed for the tender *tenderCode* and service type *serviceType*.  A59 |
| **Offered power is not below the minimum allowed for aFRR service type** | The system checks for each bid separately that the offered power values are equal to or higher than the minimum defined for the given AnS service.  The validation is skipped in case the offered power value is 0. | The bid *tenderBidIdentification* offered power (*offeredPower*) is lower than minimum quantity (*ansService.minSelectedQty*).  A59 |
| **Offered power is not below the minimum allowed for non-aFRR service type** | The system checks for each bid separately that the offered power values are equal to or higher than the minimum defined for the given AnS service.  The validation is skipped in case the offered power value is 0. | The bid *tenderBidIdentification* offered power (*offeredPower*) is lower than minimum quantity (*ansService.minQty*).  A59 |
| **Offered power does not exceed the maximum allowed for aFRR service type** | The system checks for each bid separately that the offered power values are equal to or lower than the maximum defined for the given AnS service. | The bid *tenderBidIdentification* offered power (*offeredPower*) is higher than maximum quantity (*ansService.maxSelectedQty*).  A59 |
| **Offered power does not exceed the maximum allowed for non-aFRR service type** | The system checks for each bid separately that the offered power values are equal to or lower than the maximum defined for the given AnS service. | The bid *tenderBidIdentification* offered power (*offeredPower*) is higher than maximum quantity (*ansService.maxQty*).  A59 |
| **Offered price does not exceed the maximum allowed for the given AnS service** | The system checks for each bid separately that the offered price values are equal to or lower than the maximum defined for the given AnS service. | The bid *tenderBidIdentification* offered price (*offeredPrice*) is higher than maximum quantity (*ansService.maxPrice*).  A59 |
| **Offered price must be non-zero for a non-zero offered power** | The system checks for each bid separately that the offered price values are non-zero for the non-zero submitted offered power. | The bid *tenderBidIdentification* offered price must be higher than 0 for entered offered power (*offeredPower*).  A59 |
| **The bid time interval must be consistent with the tender period** | The system checks that the bids are submitted for the whole time interval of the tender period. | The Tender Bid time interval must be equal to the Tender Period.  A04 |
| **The individual bid time intervals must be within the tender period** | The system checks for each bid separately that the bid is submitted for the time interval within the tender period. | All Tender Bids must be provided within the Tender Period.  A04 |
| **Offered power does not exceed the demand power** | The system checks for each bid separately that the offered power is equal to or lower than the demand power for the given AnS service in the given tender. | Offered Power of the bid *tenderBidIdentification* exceeds the Demand Power for the tender *tenderCode* and service *serviceType*.  A59 |

### Submit AnS Demand File

This service enables submitting/modifying AnS demand in Damas for all Market Participants registered as AnS Manager.

#### Description

This data flow enables uploading AnS demand Timeseries in the XML format. The standard *Damas type – Reserve Requirement Document v6r0* is used. The AnS demand Timeseries are declared for the Tender.

The authorized user(s) are allowed to submit AnS demand until the Tender is published.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | TEN\_DEM\_IN |  |
| **XML** | XmlParam | XML *Reserve Requirement Document* with AnS demand time series. | The XML file may contain multiple AnS Demand Timeseries provided for the tender and AnS service. |

#### Output Parameters

Users receive an acknowledgement message as response confirming data delivery and describing processing results. Please see the chapter [*7.2.1 – Acknowledgment Document*](#_Acknowledgement_Document) for further information.

#### Business Validations

The following business validations are performed when uploading AnS Demand. An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes. The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Please see the table below.

|  |  |  |
| --- | --- | --- |
| **Validation** | **Description** | **Error text / ACK Reason code** |
| **Tender must exist** | Tender for which AnS demand is submitted must exist in Damas. | The tender *list.tenderCode* does not exist.  A74 |
| **Sender is authorized to submit AnS demand** | Sender must be authorized to submit AnS demand. | Sender must be equal to Domestic TSO.  A05 |
| **Tender must be in proper business state** | Tender must be in state “Scheduled”. | Tender *tender.tenderCode* is not in state for entering Demanded Power.  A74 |
| **The AnS demand time interval must be consistent with the tender period** | The system checks that the demand is submitted for the whole time interval of the tender period. | The Tender Demand time interval (*tenderDemandCommonMesssage.timeInterval*) must be the same the Tender Period (*tender.timeInterval*).  A04 |
| **The individual AnS demand time intervals must be within the tender period** | The system checks for each AnS service separately that the demand is submitted for the time interval within the tender period. | All Tender Demand values must be provided within the Tender Period.  A04 |
| **AnS demand is submitted for AnS service defined in the Tender** | The system checks for each AnS service separately that the demand is submitted for AnS service defined in the given tender. | The requested service *list.serviceCode* is not included in the list of services for the tender *tender.tenderCode*. Possible services are (*tender.tenderServiceList.serviceCode*).  999 |

### Submit AnS Contract File

This service enables submitting/modifying AnS contracts in Damas for all Market Participants registered as AnS Manager and AnS Participant Writer. AnS Manager may submit regulated type of AnS contracts. AnS Participant Writer may submit transfer type of AnS contracts.

#### Description

This data flow enables uploading AnS contract Timeseries in the XML format. The standard *Damas type – Contract Anex Document v6r0* is used. The AnS contract Timeseries are declared for the AnS service, subject AnS participant and contract type (i.e., type regulated or transfer). In case of transfer, the particular contract annex identification must be specified as the source contract.

The authorized user(s) are allowed to submit AnS contracts of type regulated at any time. AnS contracts of type transfer may be submitted before the predefined deadline as set in the system for the given AnS service.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | ANS\_CON\_IN |  |
| **XML** | XmlParam | XML *Contract Anex Document* with AnS contract time series. | The XML file may contain multiple AnS Contract Timeseries provided for the AnS service, subject AnS participant and contract type. |

#### Output Parameters

Users receive an acknowledgement message as response confirming data delivery and describing processing results. Please see the chapter [*7.2.1 – Acknowledgment Document*](#_Acknowledgement_Document) for further information.

#### Business Validations

The following business validations are performed when uploading AnS Contracts. An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes. The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection.

Please see the table below.

|  |  |  |
| --- | --- | --- |
| **Validation** | **Description** | **Error text / ACK Reason code** |
| **AnS contract of type tender cannot be created or updated** | The system checks for each AnS contract separately that the contract type is different from tender. | Tender type of AnS Contract cannot be created nor updated. |
| **AnS contract cannot be updated in state other that "created"** | In case of update, the system checks for each AnS contract separately that the existing contract is in state “Created”. | AnS Contract *contractIdentification* cannot be updated in state *ansContract.state*. |
| **Regulated AnS contract can be created/updated only by AnS Manager** | The system checks that the regulated type of AnS contract is created/updated by a user assigned to the AnS Manager user role. | The sender *senderMarketParticipantId.name* (*senderMarketParticipantId.EIC*) is not authorized to create or update regulated type of AnS Contract. |
| **Transfer AnS contract can be created/updated only by the contract owner** | The system checks that the transfer type of AnS contract is created/updated by the owner of the given contract. | The sender *senderMarketParticipantId.name* (*senderMarketParticipantId.EIC*) is not the owner of the original ANS Contract (*originalAnsContract.contractIdentification*). |
| **Transferor must be different from Transferee** | The system checks for each AnS contract separately that the transferee is different from the transferor. | Transferee *list.subjectANSParticipantCode.marketParticipant.name* (*list.subjectANSParticipantCode.marketParticipant.EIC*) must be different from Transferor s*enderMarketParticipantId.name* (*senderMarketParticipantId.EIC*).  A22 |
| **The transfer AnS contract time intervals must be within the time interval of the original contract** | The system checks for each contract separately that the transfer AnS contract is submitted for the time interval within the time interval of the original contract. | The time interval of the transfer (*list.timeInterval*) is not within the time interval of the original ANS Contract (*originalAnsContract.timeInterval*).  A04 |
| **Original AnS contract must be confirmed prior creating a new transfer** | If a transfer is created from a contract of type transfer, the system checks that the original contract is in state "Confirmed". | The original ANS Contract Transfer *originalAnsContract.contractIdentification*must be Confirmed prior creating new Transfer.  999 |
| **Original AnS contract must not be cancelled** | If a transfer is created from a contract of type regulated or tender, the system checks that the original contract is not in state "Cancelled". | The original ANS Contract *originalAnsContract.contractIdentification* must not be cancelled.  999 |
| **Transfer service must be equal to original AnS contract service** | The system checks for each contract separately that the AnS service is equal to AnS service in the original contract. | The ANS Transfer service type (*list.serviceType*) must be equal to the Original ANS Contract service type (*originalAnsContract.serviceType*).  999 |
| **Transfer must be submitted until defined deadline** | The system checks that the transfer is received before the predefined deadline as defined for the given AnS service. | The Transfer for the time interval *timeInterval* was submitted after deadline.  A57 |
| **Transfer contracted price must be equal to the original AnS contracted price** | The system checks for each contract separately that the contracted price is equal to the contracted price in the original contract. | The Transfer Contract Price (*itemList.contractedPrice*) differs from the Contracted Price of Original ANS Contract (or*iginalAnsContractContractedPriceItemList*).  999 |
| **Subject AnS participant must be authorized to have AnS contract for the given AnS service** | The system checks for each contract separately that the AnS participant is assigned to the given AnS service. | The ANS Participant *subjectANSParticipantCode.marketParticipant.name* (*subjectANSParticipantCode.marketParticipant.EIC*) is not authorized to have ANS Contract for service *serviceType*.  A05 |
| **New Available Power is not negative** | The system checks for each contract separately that the new available power will not be negative. | New contracted power does not cover all related transfers. Recalculated available power is lower than 0.  999 |

## Data Flows for Data Download

### Download (Internal) Domestic Schedules

This data flow is used for downloading (Internal) Domestic Schedules for the requested Time interval (i.e., Date from – Date to) by authorized user(s).

#### Description

The data flow is intended for downloading the domestic schedules in detailed form. The format of the XML file is based on the *Schedule Message v2r3*.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | DOMESTIC\_SCHEDULES\_MANUAL\_DOWNLOAD\_XML\_OUT | Mandatory parameter |
| **Date from** | DateParam | Business date from for which the data should be downloaded in format YYYY-MM-DD | Mandatory parameter |
| **Date to** | DateParam | Business date to for which the data should be downloaded in format YYYY-MM-DD | Mandatory parameter |

Example of the input parameters:

<Input>

<FID>**DOMESTIC\_SCHEDULES\_MANUAL\_DOWNLOAD\_XML\_OUT**</FID>

<Parameters>

<DateParam Name="**DateFrom**">**2018-04-24**</DateParam>

<DateParam Name="**DateTo**">**2018-04-25**</DateParam>

</Parameters>

</Input>

#### Input Validations

The following validations are being performed for input parameters:

|  |  |
| --- | --- |
| **Validation** | **Error text** |
| **No configuration in master data found (for requested dataflow)** | No configuration for this data flow was found in master data. |
| **Time interval (Date from – Date to) is invalid** | Time interval is longer than 31 days. |
| **Sender is not authorized** | User with uuID *uuId* is not authorized to download *dataFlowCode.name* data. |
| **No data found** | No data for this data flow was found. |

#### Output Parameters

(Internal) Domestic Schedules are received in the XML file.

### Download Cross-border Schedules (Nominations)

This data flow is used for downloading Cross-Border Schedules for the requested Time interval (i.e., Date from – Date to) by authorized user(s).

#### Description

The data flow is intended for downloading the cross-border schedules in detailed form. The format of the XML file is based on the *Schedule Message v2r3*.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | CROSS\_BORDER\_SCHEDULES\_MANUAL\_DOWNLOAD\_XML\_OUT | Mandatory parameter |
| **Date from** | DateParam | Business date from for which the data should be downloaded in format YYYY-MM-DD | Mandatory parameter |
| **Date to** | DateParam | Business date to for which the data should be downloaded in format YYYY-MM-DD | Mandatory parameter |

Example of the input parameters:

<Input>

<FID>**CROSS\_BORDER\_SCHEDULES\_MANUAL\_DOWNLOAD\_XML\_OUT**</FID>

<Parameters>

<DateParam Name="**DateFrom**">**2018-04-24**</DateParam>

<DateParam Name="**DateTo**">**2018-04-25**</DateParam>

</Parameters>

</Input>

#### Input Validations

The following validations are being performed for input parameters:

|  |  |
| --- | --- |
| **Validation** | **Error text** |
| **No configuration in master data found (for requested dataflow)** | No configuration for this data flow was found in master data. |
| **Time interval (Date from – Date to) is invalid** | Time interval is longer than 31 days. |
| **Sender is not authorized** | User with uuID *uuId* is not authorized to download *dataFlowCode.name* data. |
| **No data found** | No data for this data flow was found. |

#### Output Parameters

Cross-Border Schedules are received in the XML file.

### Download Auction Results (LT, DA, ID auctions)

This data flow is used for downloading results for individual auction per Auction Participant by authorized user(s).

#### Description

The data flow is intended for downloading the auction results in detailed form. The format of the XML file is based on the *Allocation Results Document v4r0*.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | AUC\_RES\_TRADERS\_OUT | Mandatory parameter |
| **AuctionID** | StringParam | Auction identification of the individual auction | Mandatory parameter |
| **AuctionParticipant** | StringParam | EIC code of the Auction Participant | Mandatory parameter |

Example of the input parameters:

<Input>

<FID>**AUC\_RES\_TRADERS\_OUT**</FID>

<Parameters>

<StringParam Name="**AuctionID**">MDRO-D-15012024-000292</StringParam>

<StringParam Name="**AuctionParticipant**">**10X--TRADER01---**</StringParam>

</Parameters>

</Input>

#### Input Validations

The following validations are being performed for input parameters:

|  |  |
| --- | --- |
| **Validation** | **Error text** |
| **No configuration in master data found (for requested dataflow)** | No configuration for this data flow was found in master data. |
| **Time interval (Date from – Date to) is invalid** | Time interval is longer than 31 days. |
| **Sender is not authorized** | User with uuID *uuId* is not authorized to download *dataFlowCode.name* data. |
| **No data found** | No data for this data flow was found. |

#### Output Parameters

Total Allocation Results are received in the XML file.

### Download Dispatch Orders

This data flow is used for downloading Dispatch Orders for the requested Time interval (i.e., Date from – Date to) by authorized user(s).

#### Description

The data flow is intended for downloading the dispatch orders in detailed form. The format of the XML file is based on the *Reserve Bid Document v7r2*.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | DISPATCH\_ORDERS\_MANUAL\_DOWNLOAD\_XML\_OUT | Mandatory parameter |
| **Date from** | DateParam | Business date from for which the data should be downloaded in format YYYY-MM-DD | Mandatory parameter |
| **Date to** | DateParam | Business date to for which the data should be downloaded in format YYYY-MM-DD | Mandatory parameter |

Example of the input parameters:

<Input>

<FID>**DISPATCH\_ORDERS\_MANUAL\_DOWNLOAD\_XML\_OUT**</FID>

<Parameters>

<DateParam Name="**DateFrom**">**2018-04-24**</DateParam>

<DateParam Name="**DateTo**">**2018-04-25**</DateParam>

</Parameters>

</Input>

#### Input Validations

The following validations are being performed for input parameters:

|  |  |
| --- | --- |
| **Validation** | **Error text** |
| **No configuration in master data found (for requested dataflow)** | No configuration for this data flow was found in master data. |
| **Time interval (Date from – Date to) is invalid** | Time interval is longer than 31 days. |
| **Sender is not authorized** | User with uuID *uuId* is not authorized to download *dataFlowCode.name* data. |
| **No data found** | No data for this data flow was found. |

#### Output Parameters

Dispatch Orders are received in the XML file.

### Download Generation Schedules

This data flow is used for downloading Generation Schedules for the requested Time interval (i.e., Date from – Date to) by authorized user(s).

#### Description

The data flow is intended for downloading the generation schedules in detailed form. The format of the XML file is based on the *Schedule Message v2r3*.

#### Input Parameters

List of the input parameters is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Description** | **Note** |
| **FID** |  | GENERATION\_SCHEDULES\_MANUAL\_DOWNLOAD\_XML\_OUT | Mandatory parameter |
| **Date from** | DateParam | Business date from for which the data should be downloaded in format YYYY-MM-DD | Mandatory parameter |
| **Date to** | DateParam | Business date to for which the data should be downloaded in format YYYY-MM-DD | Mandatory parameter |

Example of the input parameters:

<Input>

<FID>**GENERATION\_SCHEDULES\_MANUAL\_DOWNLOAD\_XML\_OUT**</FID>

<Parameters>

<DateParam Name="**DateFrom**">**2018-04-24**</DateParam>

<DateParam Name="**DateTo**">**2018-04-25**</DateParam>

</Parameters>

</Input>

#### Input Validations

The following validations are being performed for input parameters:

|  |  |
| --- | --- |
| **Validation** | **Error text** |
| **No configuration in master data found (for requested dataflow)** | No configuration for this data flow was found in master data. |
| **Time interval (Date from – Date to) is invalid** | Time interval is longer than 31 days. |
| **Sender is not authorized** | User with uuID *uuId* is not authorized to download *dataFlowCode.name* data. |
| **No data found** | No data for this data flow was found. |

#### Output Parameters

Generation Schedules are received in the XML file.

# XSD SCHEMAS

This part of the document provides detailed technical description of Nomination XSD schemas used in Damas. Each XSD description contains model of the XSD schema structure, detailed description of the schema and explanation of the meaning of all XSD elements. Examples of the XML files are attached in the Appendix.

## List of the XSD Schemas

### ENTSO-E XSD Schemas

|  |  |  |
| --- | --- | --- |
| **Web Service ID** | **Description** | **Name of XSD Schema** |
| **AUC\_BID\_IN** | ECAN – Bid Document v4r0 used for uploaded Auction Bids. | bid-document.xsd |
| **DOM\_SCH\_IN** | ESS – Schedule Message v3r1 used for uploading Domestic Schedules. | schedule-xml.xsd |
| **GEN\_SCH\_IN** | ESS – Schedule Message v3r1 used for uploading Generation Schedules. | schedule-xml.xsd |
| **AVA\_DEC\_IN** | Damas type – Availability Declaration Document v6r0 used for uploading Availability Declaration Schedules. | availability-declaration-document.xsd |
| **CBS\_IN** | ESS – Schedule Message v3r1 used for uploading Cross-Border Schedules. | schedule-xml.xsd |
| **BAL\_BIDS\_AND\_NEEDS\_IN** | CIM – Reserve Bid Document v7r2 used for uploading Balancing Bids. | iec62325-451-7-reservebiddocument\_v7\_2.xsd |
| **ANS\_BID\_IN** | Damas type – Reserve Bid Document v6r0 used for uploading AnS Bids. | reserve-bid-document.xsd |
| **TEN\_DEM\_IN** | Damas type – Reserve Requirement Document v6r0 used for uploading AnS Demand. | reserve-requirement-document.xsd |
| **ANS\_CON\_IN** | Damas type – Contract Anex Document v6r0 used for uploading AnS Contract. | contract-anex-document.xsd |
| **Result of Input WS** | CIM Acknowledgement document is used for acknowledging receptions of the incoming data | iec62325-451-1-acknowledgement\_v8\_1 |
| **Result of Input WS** | ECAN Acknowledgement document is used for acknowledging receptions of the incoming data | urn-entsoe-eu-wgedi-acknowledgement-acknowledgementdocument-6-0.xsd |

## Description of ENTSO-E XSD Schemas

All XSD Schemas described below are based on the ENTSO-E standards.

### Acknowledgement Document

#### Acknowledgement Document Description

The Acknowledgement document is sent as feedback to the sender. The Acknowledgment document confirms reception of the submitted document and provide information about status of processing. In case of rejection, the Acknowledgement document notifies recipient of errors identified during processing of the document. Acknowledgement message is generated according to the ENTSO-E CIM Acknowledgement Document v8r1 or ECAN Acknowledgement Document v6r0.

The Acknowledgement document header contains document identification, current date and time, identification of the document sender and recipient, type and process type of the original document. Except for this, received document identification and version are included in the elements defining the ID and version of the received document.

An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes (element code). The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Such Reason Code specifies the validation rule that is not passed.

#### Acknowledgement Document Specification

##### Specification of AcknowledgementMarketDocument Elements (CIM)

A list of the XML elements included in the AcknowledgementMarketDocument element are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Description** | **Values** | **Applicability** |
| **mRID** | Unique identification of the acknowledgement of the document that has been received. | The naming convention is: ACK\_<FID>\_<DAMAS\_REQUEST\_ID>  NOTE: if needed, the string is trimmed according to the XSD specification | Mandatory |
| **createdDateTime** | Date and time of the transmission of the acknowledgement. | The time must be expressed in UTC as YYYY-MM-DDTHH:MM:SSZ. | Mandatory |
| **sender\_MarketParticipant.mRID** | Identification of the party that is originator of the acknowledgement. | EIC Party Code of the sender. A01 coding scheme. | Mandatory |
| **sender\_MarketParticipant.marketRole.type** | Identification of the role that is played by sender. | A04 (System Operator) | Mandatory |
| **receiver\_MarketParticipant.mRID** | Identification of the party who is recipient of the acknowledgement. | EIC Party Code of the receiver. A01 coding scheme. | Mandatory |
| **receiver\_MarketParticipant.marketRole.type** | Identification of the role played by receiver. | Receiver role, e.g. A04 (System Operator) | Mandatory |
| **received\_MarketDocument.mRID** | ID of the document that is acknowledged. | String | Mandatory |
| **received\_MarketDocument.revisionNumber** | Version of the document received. | Number equal or greater than 1. | Mandatory |
| **received\_MarketDocument.type** | Type of the document received. | A01 | Mandatory |
| **received\_MarketDocument.process.processType** | Process Type of the original document. | Process type | Mandatory |
| **received\_MarketDocument.title** | NOT USED | | |
| **received\_MarketDocument.createdDateTime** | Creation Date and Time of the document received. | In format YYYY-MM-DDTHH:MM:SSZ | Mandatory |
| **Reason** | Description of the errors discovered in received document. |  | Mandatory |

A list of the XML elements included in the Reason element are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Description** | **Values** | **Applicability** |
| **code** | Code providing the acknowledgement status. | Reason code | Mandatory |
| **text** | Textual description of the rejection. | Reason description | Optional |

An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes (element code). The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Such Reason Code specifies the validation rule that is not passed.

##### Acknowledgement Document (CIM) - Example

The following example represents an Acknowledgement Document generated after submission of the data flow BAL\_BIDS\_AND\_NEEDS\_IN (Upload of Balancing Bids). The File is rejected by Damas (Reason Code A02). The rejection is caused by submission of bids by BSP who is not assigned to the given power unit (Reason Code A82).

<?xml version="1.0" encoding="UTF-8"?>

<Acknowledgement\_MarketDocument xmlns="urn:iec62325.351:tc57wg16:451-1:acknowledgementdocument:8:1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:iec62325.351:tc57wg16:451-1:acknowledgementdocument:8:1 iec62325-451-1-acknowledgement\_v8\_1.xsd">

<mRID>BAL\_BIDS\_AND\_NEEDS\_IN\_3242344</mRID>

<createdDateTime>2018-07-13T14:10:02Z</createdDateTime>

<sender\_MarketParticipant.mRID codingScheme="A01">10X1001A1001A58S</sender\_MarketParticipant.mRID>

<sender\_MarketParticipant.marketRole.type>A04</sender\_MarketParticipant.marketRole.type>

<receiver\_MarketParticipant.mRID codingScheme="A01">10X--TRADER01---</receiver\_MarketParticipant.mRID>

<receiver\_MarketParticipant.marketRole.type>A30</receiver\_MarketParticipant.marketRole.type>

<received\_MarketDocument.mRID>20180713\_A19\_10X--TBDL\_NLGB</received\_MarketDocument.mRID>

<received\_MarketDocument.revisionNumber>1</received\_MarketDocument.revisionNumber>

<received\_MarketDocument.process.processType>A19</received\_MarketDocument.process.processType>

<received\_MarketDocument.createdDateTime>2018-07-13T14:05:02Z</received\_MarketDocument.createdDateTime>

<Reason>

<code>A02</code>

<text>Message fully rejected</text>

</Reason>

<Reason>

<code>A82</code>

<text>provider\_MarketParticipant.mRID must be an existing BSP registered in Damas and must be assigned to the respective Power Unit (registeredResource.mRID)</text>

</Reason>

</Acknowledgement\_MarketDocument

##### Specification of AcknowledgementDocument Elements ECAN

A list of the XML elements included in the AcknowledgementDocument element are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Description** | **Values** | **Applicability** |
| **DocumentIdentification** | Unique identification of the acknowledgement of the document that has been received. | The naming convention is: ACK\_<FID>\_<DAMAS\_REQUEST\_ID>  NOTE: if needed, the string is trimmed according to the XSD specification | Mandatory |
| **DocumentDateTime** | Date and time of the transmission of the acknowledgement. | The time must be expressed in UTC as YYYY-MM-DDTHH:MM:SSZ. | Mandatory |
| **SenderIdentification** | Identification of the party that is originator of the acknowledgement. | EIC Party Code of the sender. A01 coding scheme. | Mandatory |
| **SenderRole** | Identification of the role that is played by sender. | A04 (System Operator) | Mandatory |
| **ReceiverIdentification** | Identification of the party who is recipient of the acknowledgement. | EIC Party Code of the receiver. A01 coding scheme. | Mandatory |
| **ReceiverRole** | Identification of the role played by receiver. | Receiver role, e.g. A04 (System Operator) | Mandatory |
| **ReceivingDocumentIdentification** | ID of the document that is acknowledged. | String | Mandatory |
| **ReceivingDocumentVersion** | Version of the document received. | Number equal or greater than 1. | Mandatory |
| **ReceivingDocumentType** | Type of the document received. | A01 | Mandatory |
| **ReceivingPayloadName** | NOT USED | | |
| **DateTimeReceivingDocument** | Creation Date and Time of the document received. | In format YYYY-MM-DDTHH:MM:SSZ | Mandatory |
| **Reason** | Description of the errors discovered in received document. |  | Mandatory |

A list of the XML elements included in the Reason element are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Description** | **Values** | **Applicability** |
| **code** | Code providing the acknowledgement status. | Reason code | Mandatory |
| **text** | Textual description of the rejection. | Reason description | Optional |

An Acknowledgement contains information related to Acceptance/Rejection by the means of Reason Codes (element code). The main Reason Code is either A01 for Acceptance, or A02 for Rejection. Supportive Reason Codes are followed in case of Rejection. Such Reason Code specifies the validation rule that is not passed.

##### Acknowledgement Document (ECAN) - Example

The following example represents an Acknowledgement Document generated after submission of the data flow AUC\_BID\_IN (Upload of Auction Bids). The File is rejected by Damas (Reason Code A02). The rejection is caused by submission of non-existing Border Direction (Reason Code A82).

<?xml version="1.0" encoding="UTF-8"?>

<AcknowledgementDocument xmlns ecc="urn:entsoe.eu:wgedi:components" xmlns:xsd=" http://www.w3.org/2001/XMLSchema " xlmns=" urn:entsoe.eu:wgedi:acknowledgement:acknowledgementdocument:6:0">

<DocumentIdentification>ACK\_AUC\_BID\_IN\_3242344</DocumentIdentification>

<DocumentDateTime>2018-07-13T14:10:02Z</DocumentDateTime>

<SenderIdentification codingScheme="A01">10X1001A1001A58S</SenderIdentification>

<SenderRole>A04</SenderRole>

<ReceiverIdentification codingScheme="A01">10X--TRADER01---</ReceiverIdentification>

<ReceiverRole>A30</ReceiverRole>

<ReceivingDocumentIdentification>20180713\_A19\_10X--TBDL\_NLGB</ReceivingDocumentIdentification>

<ReceivingDocumentVersion>1</ReceivingDocumentVersion>

<ReceivingDocumentType>A19</ReceivingDocumentType>

<DateTimeReceivingDocument>2018-07-13T14:05:02Z</DateTimeReceivingDocument>

<Reason>

<code>A02</code>

<text>Message fully rejected</text>

</Reason>

<Reason>

<code>A82</code>

<text>OutArea and InArea must be an existing Border Direction registered in Damas and must be assigned to the respective Border (Domain)</text>

</Reason>

</AcknowledgementDocument

# WSDL File

