Regulation F1:

EDI communication with the DataHub in the electricity market

Appendix report 4:

DataHub - Web service interface

February 2011

Version 2.0

Effective from 1 October 2012

1.0		16-6-2010	24-6-2010	29-6-2010		DATE
		BOO	MBN	JHH		NAME
2.0		1-3-2011	1-3-2011	25-3-2011	14-6-2012	DATE
		FSD	CCO	JHH	JHH	NAME
REV.	DESCRIPTION	PREPARED	CHECKED	REVIEWED	APPROVED	
		63360-10				
		DOC. NO.				

Table of contents

1. Introduction

This appendix report describes the technical specifications and rules for the DataHub web service interface. Procedures and rules are defined in the relevant BRSs; see 'Business processes for the Danish electricity market'.

1.1 Purpose and target group

The purpose of this document is to specify the technical requirements to be met by the players when communicating via the DataHub web service interface.

1.2 XML structure descriptions (XSD)

All XML Schema Definitions relevant for messages affected by business transactions described in this document can be found at the following address:

http://edi.energinet.dk/schemas

1.2.1 Versioning for XML schema

XML schemas developed for communication between Energinet.dk and its external partners use a target namespace designed in the following way:

For messages covered by bilateral agreements:

http://www.energinet.dk/schemas/<subnamespace>/<document>/v<version>

For messages part of the ebIX standard:

prefix:EEM-DK_<NavnPåForretningsTransaktion>

The example below shows how the naming of a namespace can look like for the XML schema concerning notification of change of supplier:

http://www.energinet.dk/schemas/datahub/RequestChangeOfSupplier/v1

The XML schema version is written in the file name. The file name thus consists of the name of the XML schema's root element combined with the version number. The two parts are separated by a - (hyphen) as shown below:

<organisation>_<RootElementName>-<version>.xsd

The example below shows the naming of the first version of an XML schema where the root element is named "RequestChangeOfSupplier":

RequestChangeOfSupplier-1.xsd

The attribute 'version' in the schema element consists of a *major* version and a *minor* version separated by a full stop, and *revision*. The following example applies to major version 2, minor version 4, revision 0:

Version="2.4.0"

Changes that are not backwards compatible will result in changes in the major version no. This means removal of non-optional elements, changing the name of elements or attributes and changes to the element structure.

Changes that are backwards compatible will result in changes in the minor version no. only. This concerns the addition of optional elements, changes to rules of attributes content (provided that these changes do not have a limiting effect) and the like.

Editorial changes, such as comments, result in changes at revision level.

It is thus possible to apply several versions of an XML schema at the same time. Upon deployment of a new version of an XML schema, Energinet.dk can choose to stop supporting one or more previous versions.

2. Web service description

2.1 Communication pattern

Communication between players and the Datahub can take place *synchronously* or *asynchronously*. Asynchronous communication is applied in connection with the performance of business processes (described in the BRSs), while synchronous communication is applied when players make ad hoc requests to the DataHub (requests for old messages or the like).

2.1.1 Asynchronous communication

Asynchronous communication involves a loose connection between the player and the DataHub. The player communicates with the DataHub via a queue system that contains messages for the player.

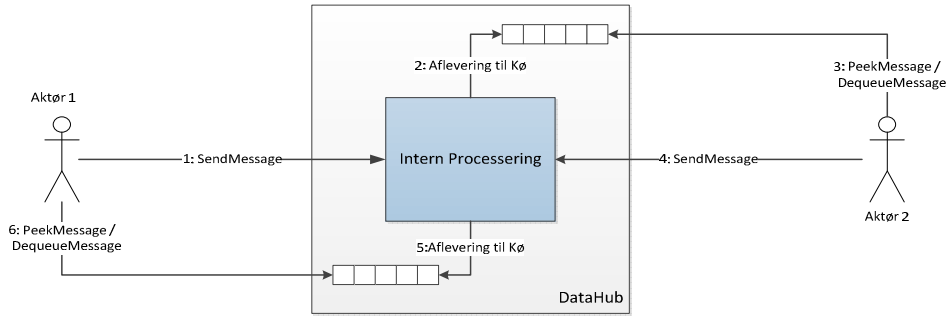


Figure 1 Asynchronous communication pattern in the DataHub

Figure 1 describes the calling process in an imagined business process in which data is exchanged between two players.

The process is initiated when Player 1 sends a message to the DataHub (1:SendMessage). The DataHub concludes that Player 2 must be notified, after which a message is generated and placed in the outgoing queue reserved for Player 2 (2: Handover to queue). Player 2 retrieves the message from the queue (3:PeekMessage/DequeueMessage) and processes it in accordance with the business process specified. Any response from Player 2 is returned to the DataHub and processed in a similar manner (4: SendMessage, 5: Handover to queue, 6: PeekMessage/DequeueMessage).

All players have their own message queue and are responsible for retrieving and removing any messages from the queue. The DataHub guarantees that messages in the queue are in the correct order (PeekMessage always returns the oldest message until it is 'removed' by calling DequeueMessage).

2.1.2 Synchronous communication

Synchronous communication involves an open connection (session) between a player and the DataHub throughout the communication process. The player waits while the message is being processed and may receive a reply when the session is closed.

2.2 Service definitions

The following *asynchronous* methods are available via the web service:

Method	Description
MessageId SendMessage (XmlDocument message)	Used for submitting business messages as defined in BRS. *Message*: The message to be submitted. Note that the message is always an XML document - EDIFACT messages as well as XML messages are packed in a common XML wrapper (see chapter 2.4). *Return value*: Message ID.
XmlDocument PeekMessage	*Return value*: The next message in the queue. If the queue is empty, Null is returned.
void DequeueMessage(MessageID)	Removes the oldest message in the queue with the ID stated.

The following synchronous methods are available via the web service:

Method	Description
XmlDocument GetMessage (MessageId id)	Retrieves the message with the given message ID. *ID:* ID of the wanted message. *Return value*: The wanted message. If no message exists with the given ID, Null is returned.
MessageId[] GetMessageIds (dateTime utcFrom, dateTime utcTo)	Retrieves IDs for a given time interval. *utcFrom, utcTo*: Definition of time interval. *Return value*: List of IDs of messages sent to the player during the given time interval.
XmlDocument QueryData(XmlDocument params)	Generic data requesting method. Reserved for future use.

The web interface supports Web Service Description Language (WSDL).

2.3 Data types

The web interface applies the following data types:

Data type	Description
MessageId	UUID as a 32-character string. The string may consist of the following 16 characters: '0123456789abcdef'

	Example: **550e8400e29b41d4a716446655440000**
BusinessProcess	String
MessageType	String, enumeration {"XML", "EDIFACT"}
dateTime	Time indication. Example: 2002-12-10T17:00:00Z (At 17:00 on 10 Cecember 2002)

2.4 Message structure

All messages communicated via the DataHub web interface are XML messages consisting of:

1. A MessageHeader containing metainformation used for controlling the underlying business process. The metainformation includes:
    - Identification of the individual message and its contents
    - Identification of the business process that is to be used for processing the message
2. One business message in either XML or EDIFACT format.

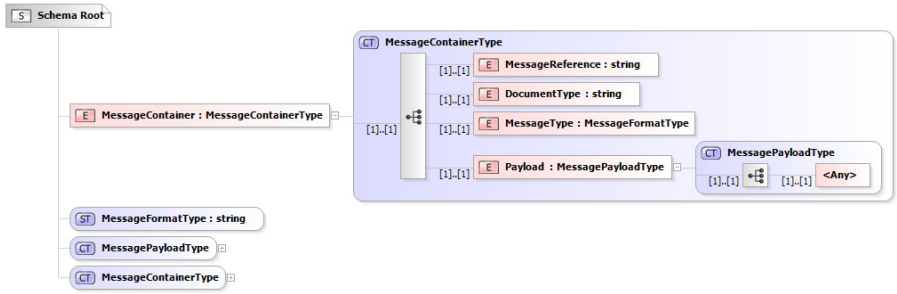


Figure 2 XML message structure

The actual business document is inserted instead of the <Any> element in Figure 2. An EDIFACT message is thus an XML document containing an EDIFACT business message sent as a single string in a <CData> element.

2.5 Handling of operators and queues

Each individual legal unit in the market has its own queue for messages from the DataHub and is identified by its GLN number. Communication with the DataHub can be divided into ingoing and outgoing communication.

Communication to the DataHub

A player can either communicate metered data with the DataHub itself or leave it to another player. Responsibility for other messages cannot be assigned to other parties.

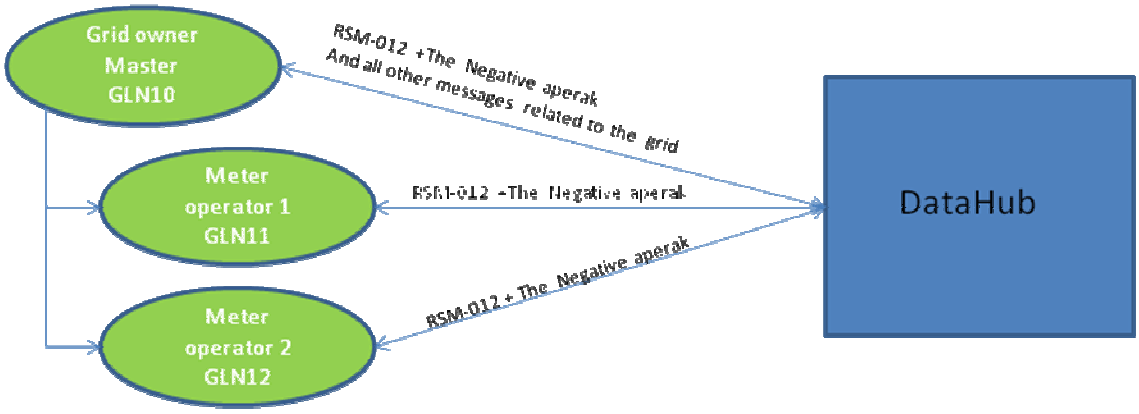
The grid company is responsible for metered data but can assign the responsibility for the collection, validation and exchange of data to a metering point administrator.

A grid company may have several metering point administrators associated to a grid area.

The fact that the authorisation is performed at grid area level means that a metering point administrator can submit metered data for metering points in a grid area where the metering point administrator is the delegated authority.

Only the following RSMs can be delegated for submission to the DataHub:

- RSM-011: Consumption for profile-settled metering point
- RSM-012: Submission of metered data for metering point



All metering point administrators can communicate with the DataHub if they are a delegated authority.
After having sent a message, the sender (metering point administrator) will receive a direct answer from Web Service (approved/rejected). The only message a metering point administrator can receive in addition to this reply, is a negative acknowledgement (RSM-009). The DataHub will always send an RSM-009 message to the physical sender.
Please note that emails not relating to EDI exchange will always be sent to the grid company (including reminder emails).

As mentioned earlier, all market players have their own queue identified by the player's GLN number. This means that if a metering point administrator submits data on behalf of several grid companies, all messages to the metering point administrator will be placed in one single queue.
If a player uses several different systems for sending messages, the player is responsible for routing these messages internally.

Communication from the DataHub
Choice of the correct recipient of a message from the DataHub takes place at RSM level. It is possible to choose another recipient than the party responsible for each RSM message.

Responsibility for the following messages cannot be delegated:
- RSM-001: Start of supply
- RSM-002: Cancel start of supply
- RSM-004: Notify on change of electricity supplier
- RSM-005: End of supply from electricity supplier
- RSM-006: Request master data at metering point
- RSM-007: Master data for metering point

The receipt of the following messages can be delegated:
- RSM-009: Acknowledgement
- RSM-011: Consumption for profile-settled metering point
- RSM-012: Submission of metered data for metering point
- RSM-013: Submit proportional figures
- RSM-014: Submit calculated time series

The functionality for exchanging RSMs-011/014 is provided by updating the player's data, and the name and GLN number of the RSMs which the player does not want to receive must be stated. There can only be one recipient per RSM.

RSM-015: Metered data request:
This RSM is used when a player (grid company, electricity suppler, metering point administrator, balance-responsible player) wants to request metered data, and it can therefore be used by the player in those situations when the RSM has been delegated to another player. The recipient is found on the basis of a request and not via RSM.

2.6 Handling of business process

The MessageHeader includes a field (known as 'DocumentType') to identify all messages interchanged via the DataHub in one business process.
This field identifies the logical business process that is closely related to the individual transactions as described in Appendix Report 1 (EDI guide - RSMs).

2.7 Validation of messages

When a player sends a message to the DataHub by means of the SendMessage() method, the 'letter-head' structure, the MessageHeader, will be validated first. When this is done, it can be decided whether the business document is based on an XML or an EDIFACT format.

2.7.1 *Processing of incoming EDIFACT messages*

When the EDIFACT string's syntax has been checked, the message is considered as having been received by the DataHub and a MessageID is returned to the system sending the message. Any semantic errors will be reported back to the supplier via an error acknowledgement, RSM-009.

2.7.2 *Processing of incoming XML messages*

The XML message structure is validated by means of the schema associated with the logical business process. When the validation is finished, a MessageID is returned to the system that sends the message. Any semantic errors will be reported back to the player via an error acknowledgement, RSM-009.

2.8 Safety

The B2B web service is accessed via an encrypted connection (HTTPS) based on certificates. Connections without valid certificate will be rejected on the transport layer.

2.9 Message sizes

Business documents can be sent together as long as they are of the same DocumentType (ie business process). However, the overall size of messages in XML format must not exceed 50 MiB[1], while messages in EDIFACT format must not exceed 10MiB.

[1] Mebibyte corresponds to 1,048,576 bytes.