

NEMHANDEL

OIOUBL BIS Message Level Response

Danish Business Authority

Version 3.0.1.1e7815bf

NEM

Table of Contents

[200]OIOUBL BIS Message Level Response	1
1. Document Structure	1
2. Business Principles and Prerequisites	1
2.1. Message Level Response General	1
2.2. Scope	2
2.3. Parties and roles	4
3. Business Process and Typical Use Cases	4
3.1. Typical use cases	5
4. Data types	9
4.1. Primitive data types	9
4.2. Semantic data types	10
5. OIOUBL 3 Identifiers	14
5.1. Profiles and messages	14
5.2. Customization and Profile identifiers	14
5.3. Namespaces	14
6. Description of Selected Elements in the Message Level Response	14
6.1. Document response	15
6.2. Line response	16
6.3. Document reference	16
6.4. Party identification	16
7. OIOUBL License and Statement of Copyright	17
8. Links to main site of documentation	17

v3.0.0 RC

The Business Interoperability Specification, called the “OIOUBL BIS” from here on, has been developed by the Danish Business Authority.

Chapter 1. Document Structure

This document is structured as follows:

- **Chapter 2** Provides general information on the business principles and processes
- **Chapter 3** Describes the business processes for typical use cases
- **Chapter 4** Describes the primitive and semantic data types
- **Chapter 5** Provides information about identifiers, e.g. customization and profile identifiers
- **Chapter 6** Provides examples of selected parts of the Message Level Response
- **Chapter 7** Statement of license and copyright
- **Chapter 8** Link to the main site for additional documentation

Chapter 2. Business Principles and Prerequisites

This chapter describes the principles and assumptions that underlie the use of the OIOUBL 3 Message Level Response.

A Messaging Level Response can be used where there is a requirement to inform the sending party about the technical validity (or not) of a received business document. It informs the sender about the results of the receiver’s validation and, in case of negative results, the details of the error(s).

The error information allows the sender party of the original business document to take appropriate action when required.

2.1. Message Level Response General

Once an electronic business message is delivered to the recipient, there may be a need to inform involved parties about the message’s status or validation results. These response types are different in nature, and for the purpose of this document they can be divided into the following main groups:

Transport acknowledgements

This message is exchanged within the transport network(s), to inform on the status of a message which is being exchanged between parties. These responses may inform a counterparty about the status of a delivery (success/failure/in process), and may contain details about relevant issues, such as why a delivery was not successful. The key feature of these responses is that they do not inform on results of validation or

processing of the of the payload content that is being transported. These response messages are commonly called “acks”.

Message Level Responses

When a message has reached a given point in a data exchange, its content may be validated according to agreed specifications. Such specifications may be syntactical and/or semantic. The outcome of these validations may be reported to a relevant up-stream party. These outcomes include the success or failure of the validation, along with additional explanatory details on errors. An example could be an order message that is rejected after receipt, because the subtotals do not add up to the order total (syntax error), or because a free text field is used to provide information for which the BIS prescribes a specific data structure (semantic error). Message level responses do not report on the business content of the communication, only on syntactic and semantic (in)validity.

Business Level Responses

A message that has been received and accepted for processing may call for an action on the receiver’s behalf. That receiver’s action may need to be reported back up-stream to a relevant party. An example is where a technically correct order may be received but the receiver decides to reject the order for a business reason such as an out-of-stock situation, expired contract etc. The key feature of these responses is that they report a business decision that is made on the message instance which is received.

This specification is only concerned with the Message Level Responses.

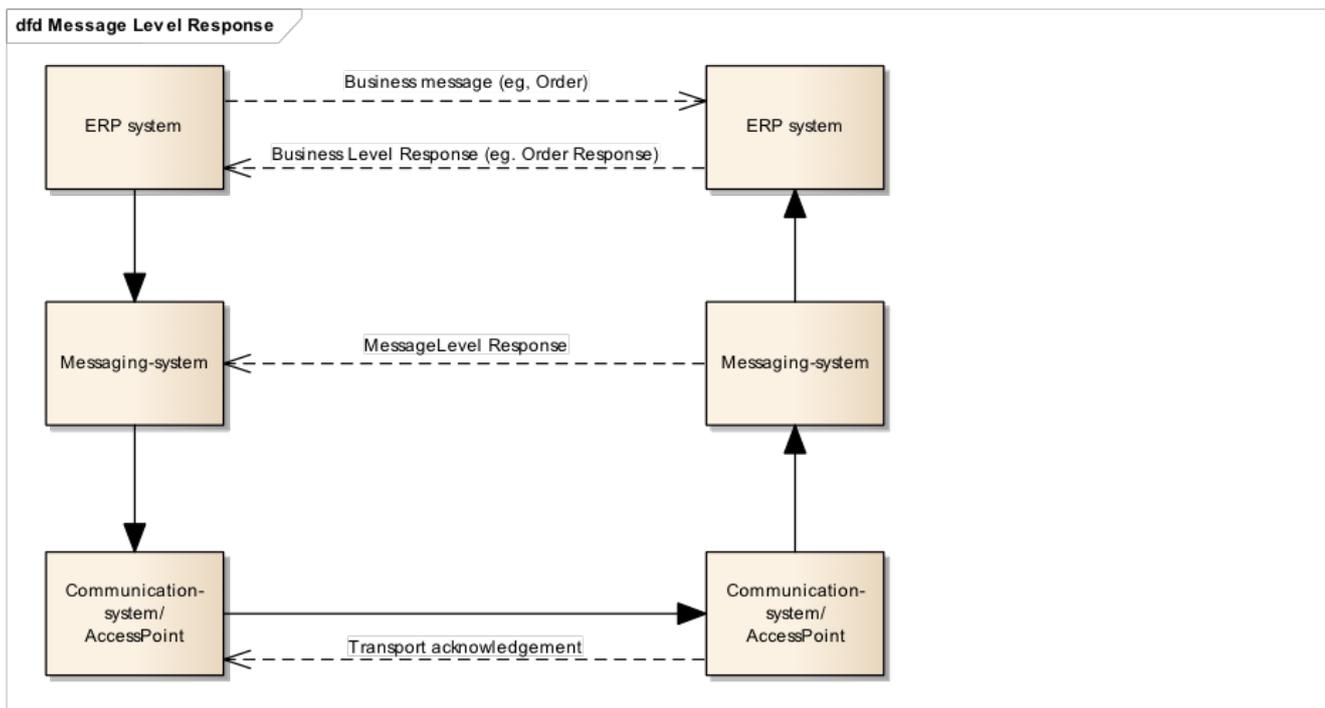


Figure 1. Flow of different response messages

2.2. Scope

The Message Level Response message is intended to inform the issuer of the following situations

1. The received message contained errors according to the relevant conformance rules.

- The message will not be processed any further.
- 2. The received message passed the validation of conformance rules without any fatal errors.
 - The message will be processed further.
- 3. The received message is not validated for conformance but the receiver acknowledges that it has been received and identified as a business message.
 - The message will be processed further.

2.2.1. The following errors are within the scope for a negative/rejecting Message Level Response:

- XML schema validation errors.
- Standard Compliance. violations (e.g. empty elements not allowed in UBL 2.1).
- Validation error of type fatal error.
- Validation error of type warning. Warnings alone must NOT cause rejection of the business document (but they may be reported in addition to fatal errors).
- Wrong version of business document (Will be handled like validation error of type fatal error).

2.2.2. The following errors are outside the scope of the Message Level Response:

- Unknown sender (in scope of transport acknowledgement).
- Unknown receiver (in scope of transport acknowledgement).
- Wrong version of envelope (in scope of transport acknowledgement).
- XML schema validation error – envelope (in scope of transport acknowledgement).
- XML not well formed (in scope of transport acknowledgement).
- Non supported encoding (in scope of transport acknowledgement).
- Wrong value (after database look-up) in reference fields (in scope of Business Level Response).

□ Examples:

- Wrong order number in invoice.
- Wrong project or customer reference in invoice.
- Wrong contract ID reference in catalogue.
- Business error or dispute (in scope of Business Level Response).

□ Examples:

- Invoiced amount does not match delivered amount.
- Ordered amount does not meet minimum orderable amount.
- Payment terms not as agreed.

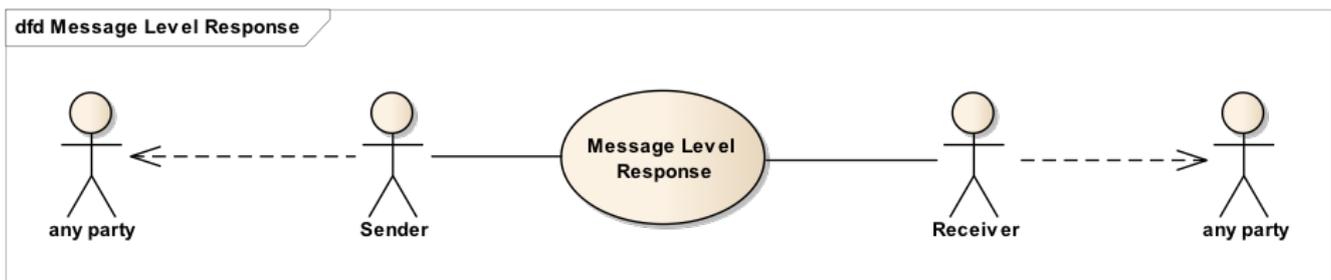
Note: Schema validation errors should be caught already in the Nemhandel eDelivery transport layer, but if this fails schema validation errors can be raised to the sender in a

message level response.

2.3. Parties and roles

The table below gives the definitions of the parties and roles in the Message Level Response process. The sender and receiver of the Message Level Response message should be extracted from the envelope, i.e. the Trading parties.

Business partners	Description
Sender	The party sending an electronic Message Level Response back to the sending party of the business document.
Receiver	The party, an electronic Message Level Response was addressed to, and who is supposed to process the Message Level Response. This is the same party as the sender of the original business document to which the Message Level Response is responding.



Chapter 3. Business Process and Typical Use Cases

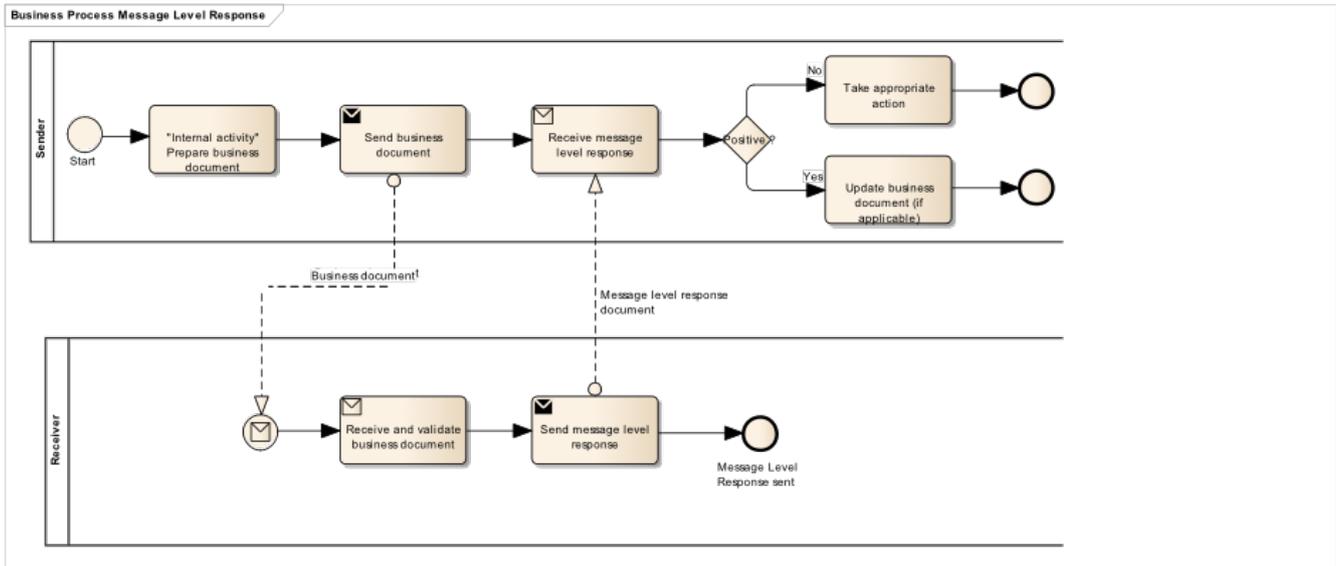
The process starts when a Business Document Sender is preparing an electronic business document and then sends it. The Business Document Receiver receives the business document and potentially validates syntax and business rules.

- If the Business Document Sender has not requested a Message Level Response, the Business Document Receiver shall either:**
 - Validate the business document and return a 'reject' if fatal errors are found.
 - Not respond with a Message Level Response if no fatal errors are found.
- If the Business Document Sender has requested an Message Level Response back, the Business Document Receiver shall either:**
 - Validate the business document and based on the result returns either an 'accept' (no fatal errors) or a 'reject' (fatal errors found)
 - Not validate the business document and send a Message Level Response with a response code indicating the message is acknowledged.



This specification does not prescribe how a request for Message Level Response is made or communicated.

If a Message Level Response is returned, the Message Level Response Receiver must be notified and take appropriate action. If the response is positive, the Message Level Response Receiver may update the status of the business document or simply ignore the Message Level Response.



3.1. Typical use cases

3.1.1. Parties/Roles:

In the use case descriptions below the following terms are used:

Business Document Sender	Sender party in the role of sending a business document
Business Document Receiver	Receiving party in the role of receiving a business document
Message Level Response Sender	Sender party in the role of sending a Message Level Response document
Message Level Response Receiver	Receiver party in the role of receiving a Message Level Response document
Service Provider (on behalf the Business Document Receiver)	Receiving party in the role of receiving a business document



In all the use cases the Business Document Sender is the same physical party as the Message Level Response Receiver and the Business Document Receiver is the same physical party as the Message Level Response Sender.

3.1.2. Use Case 1 – Negative response – violation of syntax and business rules

This use case is a Message Level Response containing errors and warnings due to violation of syntax rules.

Use Case number	1
Use Case Name	Negative response – violation of business rules, business rule warnings and violation of syntax
Use Case Description	This use case is a Message Level Response to a business document that contains errors and warnings, due to violation of business rules and syntax.
Parties involved	Business Document Sender Message Level Response Receiver Service Provider (on behalf of the Business Document Receiver) Message Level Response Sender
Assumptions	<ol style="list-style-type: none">1. The Service Provider (on behalf of the Business Document Receiver) has received an electronic business document from the Business Document Sender.2. The Service Provider (on behalf of the Business Document Receiver) has validated the business document from the Business Document Sender.3. The result of the validation is not OK due to violation of business rules and/or syntax.
The flow	<ol style="list-style-type: none">1. The Business Document Sender has prepared and sent an electronic business document to the Service Provider (who is working on behalf of the Business Document Receiver).2. The Service Provider (on behalf of the Business Document Receiver) has received the business document.3. The Service Provider (on behalf of the Business Document Receiver) has validated and rejected the business document.4. The Message Level Response Sender sends a Message Level Response back to the Message Level Response Receiver (using the Message Level Response Network document profile).5. The Message Level Response Receiver receives and processes the message level response message and performs appropriate action due to the rejection.

Use Case number	1
Result	The Message Level Response provides the Business Document Sender's Service Provider with a confirmation that the business document was received and rejected by the Business Document Receiver's Service Provider (on behalf of the Business Document Receiver), due to an error which should have been detected by the Business Document Sender's Service Provider prior to sending (schema or syntax error). The Business Document Sender's Service Provider must take appropriate action to correct and resend the business document, or follow up with the Business Document Sender to correct invalid data. The Business Document Sender's Service Provider should also investigate the root cause of their failure to properly validate the document prior to sending, and take appropriate corrective action.

3.1.3. Use Case 2 – Positive response

This use case is a Message Level Response containing no errors, ie a positive response.

Use Case number	2
Use Case Name	Positive response
Use Case Description	This use case is a Message Level Response based on a business document with no errors, ie a positive response.
Parties involved	Business Document Sender Message Level Response Receiver Business Document Receiver Message Level Response Sender
Assumptions	<ol style="list-style-type: none"> 1. The Business Document Receiver has received an electronic business document from the Business Document Sender. 2. The Business Document Receiver has validated the business document from the Business Document Sender. 3. The result of the validation is OK, no fatal errors.

Use Case number	2
The flow	<ol style="list-style-type: none"> 1. The Business Document Sender has prepared and sent an electronic business document to the Business Document Receiver. 2. The Business Document Receiver has received the business document. 3. The Business Document Receiver has validated the business document. 4. The Message Level Response Sender sends a Message Level Response back to the Business Document Sender (using the Message Level Response document profile). 5. The Message Level Response Receiver receives and processes the message level response message.
Result	The Message Level Response provides the Business Document Sender a confirmation that the business document was received and validated with no errors by the Business Document Receiver.

3.1.4. Use Case 3 – Negative response – violation of business rules

This use case is a Message Level Response containing errors due to violation of semantic rules, which are not captured by the syntax validation.

Use Case number	3
Use Case Name	Negative response – violation of business rules
Use Case Description	This use case is a Message Level Response based on a business document containing errors due to violation of the semantic data model.
Parties involved	Business Document Sender Message Level Response Receiver Business Document Receiver Message Level Response Sender
Assumptions	<ol style="list-style-type: none"> 1. The Business Document Receiver has received an electronic business document from the Business Document Sender. 2. The Business Document Receiver has validated the business document from the Business Document Sender. 3. The result of the validation is not OK due to a semantic error.

Use Case number	3
The flow	<ol style="list-style-type: none"> 1. The Business Document Sender has prepared and sent an electronic business document to the Business Document Receiver. 2. The Business Document Receiver has received the business document. 3. The Business Document Receiver has validated and rejected the business document. 4. The Message Level Response Sender sends a message level response message back to the Business Document Sender (using the Message Level Response document profile). 5. The Message Level Response Receiver receives and processes the message level response message and takes appropriate action due to the rejection.
Result	The Message Level Response communicated to the Business Document Sender a confirmation that the business document was received and rejected by the Business Document Receiver. The Business Document Sender must take appropriate action to correct and resend the business document.

Chapter 4. Data types

Semantic data types are used to bridge the gap between the semantic concepts expressed by the information elements and the technical implementation. The semantic data types define the allowed value domain for the content, and any additional information components (attributes) needed in order to ensure a precise interpretation of the content.

4.1. Primitive data types

Semantic data type content may be of the following primitive types. These primitive types were taken from [ISO 15000-5:2014](#), Annex A.

Primitive type	Definition
Binary	A set of finite-length sequences of binary digits.
Date	Time point representing a calendar day on a time scale consisting of an origin and a succession of calendar ISO 8601:2004 .
Decimal	A subset of the real numbers, which can be represented by decimal numerals.
String	A finite sequence of characters.

4.2. Semantic data types

The semantic data types are described in the tables below, where various features such as attributes, format, and decimals as well as the basic type are defined for each semantic data type. They are based on [ISO 15000-5:2014](#).

When used in an instance document, each data element will contain data. In the tables below this is identified as the “content”. Whenever a business term is used this term shall **always** have content and therefore the content is always mandatory.

4.2.1. Amount

An amount states a numerical monetary value. The currency of the amount is defined as a separate business term.



Amount is floating up to two fraction digits.

Component	Use	Primitive Type	Example
Content	Mandatory	Decimal	10000.25

4.2.2. Unit Price Amount

A unit price amount states a numerical monetary amount value for data elements that contain item prices that may be multiplied by item quantities. The currency of the amount is defined as a separate business term.



Unit price amount does not set restrictions on number of decimals, as contrast to the Amount type

Component	Use	Primitive Type	Example
Content	Mandatory	Decimal	10000.1234

4.2.3. Percentage

Percentages are given as fractions of a hundred (per cent) e.g. the value 34.78 % in percentage terms is given as 34.78.



No restriction on number of decimals for percentages.

Component	Use	Primitive Type	Example
Content	Mandatory	Decimal	34.7812

4.2.4. Quantity

Quantities are used to state a number of units such as for items. The code for the Unit of Measure is defined as a separate business term.



No restriction on the number of decimals for quantities.

Component	Use	Primitive Type	Example
Content	Mandatory	Decimal	10000.1234

4.2.5. Code

Codes are used to specify allowed values in elements as well as for lists of options. Code is different from Identifier in that allowed values have standardized meanings and gives clarity to the recipient.



Codes shall be entered exactly as shown in the selected code list

Component	Use	Primitive Type	Example
Content	Mandatory	String	Abc123

4.2.6. Identifier

Identifiers (IDs) are keys that are issued by the sender or recipient of a document or by a third party.



The use of the attributes is specified for each information element.

Component	Use	Primitive Type	Example
Content	Mandatory	String	abc:123-DEF
Scheme identifier	Conditional	String	0088
Scheme version identifier	Conditional	String	1.0

4.2.7. Date

Dates shall be in accordance to the 'Calendar date complete representation' as specified by [ISO 8601:2004](#), format **YYYY-MM-DD**.



Dates shall **not** include timezone information.

Table 1. EN 16931_ Date. Type

Component	Use	Primitive Type	Example
Content	Mandatory	Date	2024-12-01

4.2.8. Time

Time shall be in accordance with the 'Extended time format' as specified by [ISO 8601:2004](#), format [hh]:[mm]:[ss].



Time shall **not** include time zone information. Decimal fraction on seconds SHALL not be used.

Table 2. EN 16931_ Time. Type

Component	Use	Primitive Type	Example
Content	Mandatory	Time	09:30:12

4.2.9. Document Reference

Document Reference Types are identifiers that were assigned to a document or document line.

Table 3. Document Reference. Type

Component	Use	Primitive Type	Example
Content	Mandatory	String	abc:123-DEF

4.2.10. Short String

Text is the actual wording of anything written or printed. Line breaks in the text must not be present. Maximum length of short string is 64 characters.

Short String shall contain only characters supported by the UTF-8 character set.

Component	Use	Primitive Type	Example
Content	Mandatory	String	Product Name

4.2.11. Long String

Free text field for longer descriptions or annotations. Line breaks in the text may be present, and any line breaks must be preserved and respected by the receiver's system.

Maximum length of a Long String is 1024 characters.

Long String shall contain only characters supported by the UTF-8 character set.

Component	Use	Primitive Type	Example
Content	Mandatory	String	5% allowance when paid within 30 days

4.2.12. Text

Text is the actual wording of anything written or printed. Line breaks in the text may be present, and any line breaks should be preserved and respected by the receiver's system.

Text fields have no length limits.

Text data type should not be used in OIOUBL, but is included here for completeness/compatibility reasons.

Component	Use	Primitive Type	Example
Content	Mandatory	String	5% allowance when paid within 30 days

4.2.13. Binary objects

Binary objects can be used to describe files which are transmitted together with the business document. Attachments shall be transmitted together with the business document. The binary object has two supplementary components: a Mime Code, which specifies the Mime type of the attachment and a Filename that is provided by (or on behalf of) the sender of the business document.

Component	Use	Primitive Type	Example
Content	Mandatory	Binary	QmFzZTY0IGNvbmlbnRlb nQgZXhhbXBsZQ==
Mime Code	Mandatory	String	image/jpeg
Filename	Mandatory	String	drawing5.jpg

4.2.14. Boolean Indicators

Boolean indicators are used to specify the two allowed values, *true* or *false*. All elements of datatype Boolean, must have either *true* or *false* as their value.

Component	Use	Primitive Type	Example
Content	Mandatory	String	true

Chapter 5. OIOUBL 3 Identifiers

OIOUBL 3 has defined how to use identifiers in the documents exchanged across the Nemhandel eDelivery infrastructure (Based on the Peppol identifier policy).

It also introduces principles for any identifiers used in the Nemhandel eDelivery environment. The policies that apply to this OIOUBL 3 BIS at the follows:

5.1. Profiles and messages

All messages contains ProfileID and CustomizationID. ProfileID identifies what business process a given message is part of, and CustomizationID identifies the kind of message and the rules which are applied.

Profiles are connected to one business process, and may contain multiple document types. Valid document instances shall contain corresponding ProfileID and CustomizationID.

5.2. Customization and Profile identifiers

Type	Element <code>cbc:CustomizationID</code>	Element <code>cbc:ProfileID</code>
Message Level Response	<code>urn:fdc:peppol.eu:poacc:trns:mlr:3@urn:fdc:oiubl.dk:trns:message_level_response:3.0</code>	<code>urn:fdc:oiubl.dk:bis:message_level_response:3</code>
Message Level Response Network	<code>urn:fdc:peppol.eu:poacc:trns:mlr:3@urn:fdc:oiubl.dk:trns:message_level_response:3.0</code>	<code>urn:fdc:oiubl.dk:bis:message_level_response_network:3</code>

5.3. Namespaces

All document types in OIOUBL BIS Message Level Response are bound to the UBL 2.1. The target namespaces are:

Type	Namespace
Message Level Response	<code>urn:oasis:names:specification:ubl:schema:xsd:ApplicationResponse-2</code>

Chapter 6. Description of Selected Elements in the Message Level Response

6.1. Document response

The document response is used to indicate the result of business document validation.

To be able to define more correct use of document response codes, two different document profiles are defined as a part of this OIOUBL BIS Message Level Response. See the details in section for OIOUBL 3 identifiers.

Table 4. Valid document response codes in the two different document profiles.

Document response code name	Document response code id	Profile Message Level Response	Profile Message Level Response Network
Rejected	RE	Default	Default - and only allowed value
Message acknowledgement	AB	Allowed	Not allowed
Accepted	AP	Allowed	Not allowed

Exactly one `cac:DocumentResponse` element MUST be present. The element `cac:DocumentResponse/cac:Response/cbc:ResponseCode` MUST contain the overall result code.

6.1.1. Rejected

In case the document is rejected, the `cbc:Description` element MUST be set.

Example for rejection:

```
<cac:Response>
  <cbc:ResponseCode>RE</cbc:ResponseCode>
  <cbc:Description>The document was rejected because it failed
validation</cbc:Description>
</cac:Response>
```

6.1.2. Accepted

Example for acceptance:

```
<cac:Response>
  <cbc:ResponseCode>AP</cbc:ResponseCode>
</cac:Response>
```

6.1.3. Acknowledgement

Example for acknowledging:

```
<cac:Response>
  <cbc:ResponseCode>AB</cbc:ResponseCode>
  <cbc:Description>The document was accepted without any validations being
performed.</cbc:Description>
</cac:Response>
```

6.2. Line response

In case of a negative response (rejection), the `cac:LineResponse` element is used to specify the errors in the business document. The `cbc:LineID` element is mandatory in UBL but not relevant since the line responses are listing errors on both header level and line level from the original document and should have the value: **NA** (“Not Applicable”, must be all uppercase).

Example:

```
<cac:LineResponse>
  <cac:LineReference>
    <cbc:LineID>NA</cbc:LineID>
  </cac:LineReference>
  <cac:Response>
    <cbc:Description>BR-CO-13: Invoice tax exclusive amount MUST equal the sum of
lines plus allowances and charges on header level</cbc:Description>
    <cac>Status>
      <cbc>StatusReasonCode>BV</cbc>StatusReasonCode>
    </cac>Status>
  </cac:Response>
</cac:LineResponse>
```

6.3. Document reference

The document reference is used to provide a reference to the envelope of the business document on which the Message Level Response is based. The Message Level Response message may only cover exactly one business document. The element `cac:DocumentResponse/cac:DocumentReference/cbc:ID` MUST contain the instance identifier of the envelope of the original business document.

Example for document references:

```
<cac:DocumentReference>
  <cbc:ID>EnvelopeID-12456789</cbc:ID>
</cac:DocumentReference>
```

6.4. Party identification

6.4.1. Sender Party

The element `cac:SenderParty/cbc:EndpointID` MUST contain the party identification of the receiver of the original envelope. If the response is a Message Level Response, this is the original business recipient (corner 4). If the response is a Message Level Response - Network, this is the technical endpoint of the business recipient’s service provider (corner 3).

Example:

```
<cac:SenderParty>  
  <cbc:EndpointID schemeID="0184">98754325</cbc:EndpointID>  
</cac:SenderParty>
```

6.4.2. Receiver Party

The element `cac:ReceiverParty/cbc:EndpointID` MUST contain the party identification of the sender of the original envelope. If the response is a Message Level Response, this is the EndpointID of the sending Party per the relevant BIS (for invoices this would be AccountingSupplierParty). If the response is a Message Level Response - Network, this is the technical endpoint of the sending party's service provider, which is found in the message envelope, per the Nemhandel eDelivery specification.

Example:

```
<cac:ReceiverParty>  
  <cbc:EndpointID schemeID="0088">7315458756328</cbc:EndpointID>  
</cac:ReceiverParty>
```

Chapter 7. OIOUBL License and Statement of Copyright

[OIOUBL 3 Licens og ophavsret](#)[OIOUBL 3 License and Statement of Copyright]

Chapter 8. Links to main site of documentation

[OIOUBL 3 Dokumentationssite](#)[OIOUBL 3 Main documentation site]