

# ORACLE FUSION MIDDLEWARE

## Oracle B2B Technical Note

### Technical Note #007

### Exception Handling

Oracle B2B handles exceptions for inbound and outbound messages. This document described the exception handling behavior, error messages and structures.

#### Contents

<i>Inbound Direction</i> .....	2
Request/Response Message .....	2
Acknowledgment Message .....	2
Exception Message .....	2
Out-of-band Exception (0A1) .....	3
<i>Outbound Direction</i> .....	3
During B2B Startup.....	3
Miscellaneous Notes .....	3
<i>Inbound Exception Handling Sample Scenarios</i> .....	4
Identification of exchange fails or exchange not supported.....	4
Unpacking of message fails .....	4
Decoding of incoming message fails .....	4
Message is duplicated .....	4
Identification of document fails .....	4
Incoming TPA processing fails .....	4
Processing of incoming document fails .....	5
Exception Payload Definition.....	5

## Inbound Direction

### Request/Response Message

In the case where an incoming message is a Request/Response message that results in an exception, the following actions will be taken

- An exception message will be sent to the application. This exception message will be enqueued to the B2B\_IN\_QUEUE and will have the recipient name "b2berroruser". The exception enqueued will be based on ipException.xsd and will have error information such as error text and error code.
- Depending on the exchange that is used, an exception message will be sent back to the TP if the exchange specification mandates it. Note that the exception message will be sent back to the TP only if there is enough information to identify the outgoing TPA. For this purpose, the flag B2BHeader.sendException will be used. The exchange (in most cases) must set this flag to true once enough information is extracted from the incoming message in order to send the exception back.
- The top-level engine code will catch any exceptions thrown by exchange or document layers. It will then check if the B2Bheader.sendException flag is set. If the flag is set, it will process the outgoing TPA, and send an exception back to TP.

### Acknowledgment Message

In the case where an incoming message is an Acknowledgment message that results in an exception, the following actions will be taken

- An exception message will be sent to the application. This exception message will be enqueued to the B2B\_IN\_QUEUE and will have the recipient name "b2berroruser". The exception enqueued will be based on ipException.xsd and will have error information such as error text and error code.
- No exception message will be sent back to the TP

### Exception Message

In the case where an incoming message is an Exception message, the following actions will be taken

- The original message will be updated to be in Error state. The incoming exception will be processed and delivered to the application normally.
- In case the incoming exception message itself results in an exception, an exception message will be sent to the application. This exception message will be enqueued to the B2B\_IN\_QUEUE and will have the recipient name "b2berroruser". The exception enqueued will be based on ipException.xsd and will have error information such as error text and error code. No exception message will be sent back to the TP in this case.

### **Out-of-band Exception (0A1)**

The support for out-of-band exceptions will be implemented later.

## **Outbound Direction**

If there is an exception while sending an outbound message, such as trading partner identification fails, an exception message will be sent to the application. This exception message will be enqueued to the B2B\_IN\_QUEUE and will have the recipient name "b2berroruser". The exception enqueued will be based on ipException.xsd and will have error information such as error text and error code.

### **During B2B Startup**

If there was an exception while starting B2B, an exception message will be enqueued to the B2B\_IN\_QUEUE and will have the recipient name "b2berroruser". The exception enqueued will be based on ipException.xsd and will have error information such as error text and error code. Note that correlation id will not be populated in this case.

### **Miscellaneous Notes**

- When the exception message is sent back to the application, the document type will be "Exception" instead of the original message's document type.
- When the exception message is sent back to the application, the inReplyToMessageId is populated with correlationId value.
- In case of inbound exception handling, a business message will always be created and populated with the available information. It will also point to the corresponding wire message. The wire message will be updated to the Error state. Note that in the outbound direction, only business message will be updated, as the wire message will not exist.
- The error reports will be updated to show only business messages, as we will always have a business message created in the inbound and outbound directions.

## Inbound Exception Handling Sample Scenarios

Depending on when the message results in exception in the incoming case, different steps will be taken as described below.

### Identification of exchange fails or exchange not supported

- Notify middleware
- Update wire message as error
- Create business message in error state for the wire message
- Send transport error to trading partner

### Unpacking of message fails

- Notify middleware
- Update wire message as error
- Create business message in error state for the wire message

### Decoding of incoming message fails

- Notify middleware
- Update wire message as error
- Create business message in error state for the wire message
- Send exception back to Trading Partner. (If sendException flag is set by exchange layer)

### Message is duplicated

- Notify middleware
- Update wire message as "duplicated message error"
- Create business message as "duplicated message error" for the wire message

### Identification of document fails

- Notify middleware
- Update wire message as error
- Create business message in error state for the wire message
- Send exception back to Trading Partner. (If sendException flag is set by exchange layer)

### Incoming TPA processing fails

- Notify middleware
- Update wire message as error
- Create business message in error state for the wire message
- Send exception back to Trading Partner (If sendException flag is set by exchange layer)

## Processing of incoming document fails

- Notify middleware
- Update wire message as error
- Create business message in error state for the wire message
- Send exception back to Trading Partner. (If sendException flag is set by exchange layer)

### Notes:

- An exception will be sent back to TP only for Request messages.
- No exception will be sent back to TP for Response, ACK, IN\_BAND\_EXCEPTION, OUT\_OF\_BAND\_EXCEPTION, FUNCTIONAL\_ACK

## Exception Payload Definition

### ipException.xsd

```
<?xml version="1.0"?>
<xs:schema targetNamespace="http://www.rn.org/Exception"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.rn.org/Exception">
  <xs:element name="Exception">
    <!--xs:complexType name="Exception"-->
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="correlationId" />
        <xs:element ref="b2bMessageId" />
        <xs:element ref="errorCode" />
        <xs:element ref="errorText" />
        <xs:element ref="errorDescription" />
        <xs:element ref="errorSeverity" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="correlationId" type="xs:string" />
  <xs:element name="b2bMessageId" type="xs:string" />
  <xs:element name="errorCode" type="xs:string" />
  <xs:element name="errorText" type="xs:string" />
  <xs:element name="errorDescription" type="xs:string" />
  <xs:element name="errorSeverity" type="xs:string" />
</xs:schema>
```