



TO: ROSETTANET MEMBERS

FROM: LINDA YORK

**SUBJECT: ROSETTANET IMPLEMENTATION FRAMEWORK,
V1.1: TECHNICAL ADVISORY #1**

In our continuing effort to support members implementing PIPs for 2.2.2000, we are publishing this Technical Advisory to prevent differing interpretations of items not explicitly addressed in the current specifications and provide helpful clarification. Although we had not planned to issue a series of amendments to the specifications, where member needs indicate that this is the best course, we will do so.

This Technical Advisory explains how Exception Handling occurs during the PIP-interchange process. This had not been documented in one place until now. In the next full release of the RosettaNet Implementation Framework (RNIF), this will be incorporated as a separate section of the documentation set. At that time, additional work will be completed on Exception Handling in general.

Additionally, this Technical Advisory specifies the handling of the DTD name in the document type declaration. This will be incorporated as part of the current section 3.1.1 in the next release of the RNIF documentation set.

1 ROSETTANET EXCEPTION HANDLING PROCESS

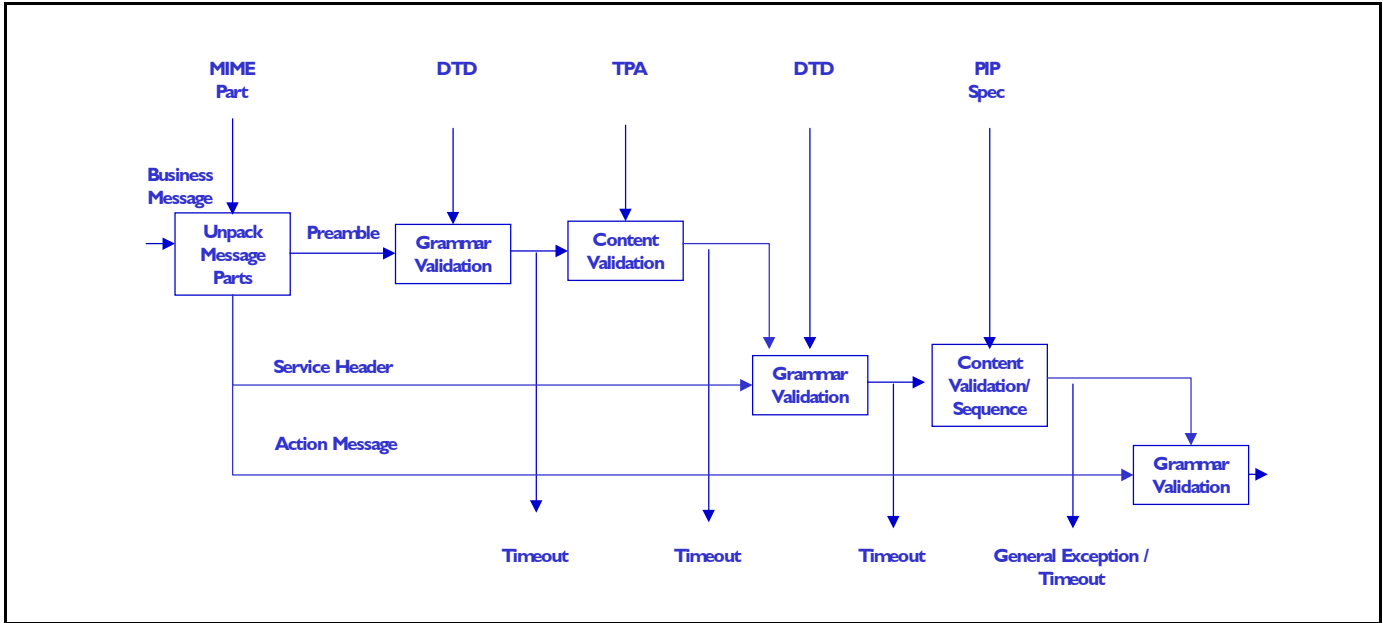
Errors are prone to happen at any point in a PIP interchange. For instance, a message received from a partner may not be compliant with the RosettaNet standard format, it may fail DTD validation, or it may be out of sequence. A message can be either a Business Message or a Business Signal. Business Signals contain Acknowledgments and Exception messages; Business Messages contain Action Messages (e.g., a Purchase Order Request). Exception handling during message processing depends on whether the message is a business message or a business signal. This section specifies how these errors should be handled at the various stages that a business message or a business signal flows through.

1.1 EXCEPTION HANDLING FOR BUSINESS MESSAGES

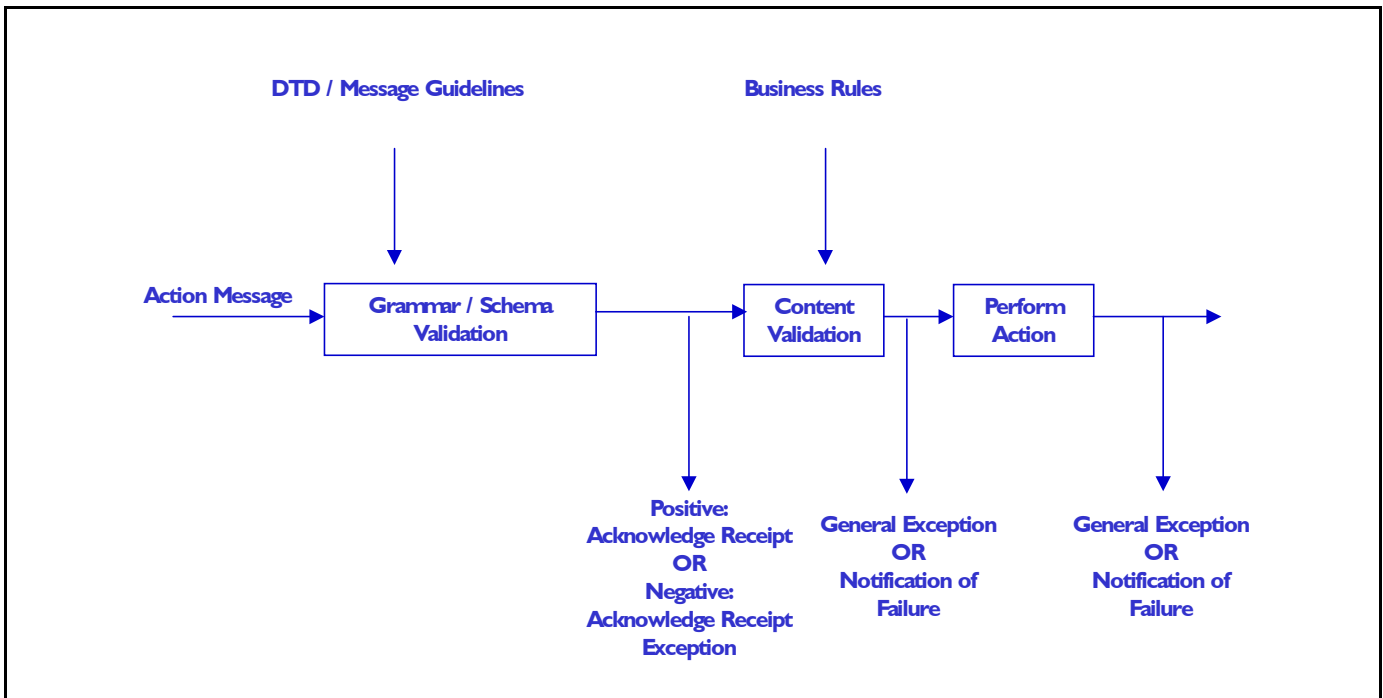
A RosettaNet Business Message goes through the following steps that are executed in a distributed fashion (i.e., not all steps are executed at the same trading partner's side). Both trading partners send and receive Business Messages during the execution of a PIP; hence each trading partner goes through each of the following series of steps, depending on the stage of PIP execution.

1. Sender: Pack & Send Message
2. Recipient: Unpack Message into Preamble, Service Header and Service Content (Action Message)
3. Recipient: Validate Grammar and Content of Preamble
4. Recipient: Validate Grammar of Service Header
5. Recipient: Validate Content of Service Header and Sequence of the Message
6. Recipient: Validate Grammar and Schema of Action Message
7. Recipient: Validate Content of Action Message
8. Recipient: Perform Action

These steps from 1 to 8 are repeated until all the actions specified in the PIP are completed. Exceptions may occur or errors may be detected in any of these steps and should be handled as specified below.



Business Message Validation (part 1)



Business Message Validation (part 2)

1.1.1 PACK & SEND MESSAGE

Packing: An error may occur during packing, either in the first message in the process or in subsequent messages. If the error occurs in packing the first message, then (as the process has not

started yet) the error would be processed internally within the sending partner's organization. RosettaNet does not specify how this should be done, as this does not affect interoperability. If the message is not the first message, then a General Exception should be sent to the receiving Partner with a reason text in the Exception message containing a descriptive reason for this failure.

Sending: Again, an error may occur in sending either the first message in the process or in subsequent messages. If the error occurs while sending the first message, then the error should be handled internally. If the message is not the first message, and if the error does not get resolved in a couple of retries, it may not be possible to do anything. The only alternative is for the sender of the message to give up, expecting the receiver to time out.

1.1.2 UNPACK MESSAGE INTO PREAMBLE, SERVICE HEADER AND ACTION MESSAGE

Any error in unpacking would mean that there is not sufficient information to respond with an exception. Hence the most feasible thing to do is for the receiver of the message to give up, expecting the sender to time out waiting for a Receipt Acknowledgment.

1.1.3 VALIDATE GRAMMAR AND CONTENT OF THE PREAMBLE

If the preamble is not valid, there would not be enough information to send back an exception. Hence the most feasible thing to do is for the receiver of the message to give up, expecting the sender to time out waiting for a Receipt Acknowledgment.

1.1.4 VALIDATE GRAMMAR OF THE SERVICE HEADER

If the Service Header is not valid, there would not be enough information to send back an exception. Hence the most feasible thing to do is for the receiver of the message to give up, expecting the sender to time out waiting for a Receipt Acknowledgment.

1.1.5 VALIDATE CONTENT OF SERVICE HEADER AND VALIDATE SEQUENCE OF THE MESSAGE

If the content validation of the service fails, and if there is sufficient information to be able to send a General Exception, then a General Exception should be sent.

If the information is not sufficient, then it should be expected that the sender would time out.

If the message is out of sequence (detected during validation of the content of the Service Header), then a General Exception message should be sent from the recipient to the sender of the message.

1.1.6 VALIDATE GRAMMAR AND SCHEMA OF ACTION MESSAGE

If the Action Message does not pass grammar or schema validation, then a Receipt Acknowledgment Exception should be sent to the sender of the message. If the Action Message passes the validation, then a Receipt Acknowledgment should be sent.

1.1.7 VALIDATE CONTENT

If the Action message content is not valid, and the current message is not the last action message in the PIP interchange, a General Exception should be sent to inform the sender that the content validation failed.

However, if the Action Message content is not valid, and if the current message is the last action message, then there is no way to inform the sender of the error within the same process instance. Hence, a Notification of Failure PIP should be initiated.

If the content is valid, proceed to the next step of Performing Action.

1.1.8 PERFORM ACTION

If an error is encountered while performing the action, and if this message is not the last Action Message in the PIP, then a General Exception should be sent back.

If an error is encountered, and if this message is the last Action Message in the PIP, then there is no way to inform the message sender of the error within this same process. Hence, on such an error, a Notification of Failure PIP should be initiated.

1.2 EXCEPTION HANDLING FOR BUSINESS SIGNALS

Business Signals include Receipt Acknowledgment, Acceptance Acknowledgment and Exceptions. The rules for handling exceptions during processing of signals are different from those for business messages. When a business signal is received and there are errors validating the Preamble or Service Header for grammar, content or sequence, then no exception should be sent to the sender of these signals, nor should the Notification of Failure PIP be initiated. This means that if an error is encountered in an Acknowledgment signal (for example), it should be treated similarly to the case where the acknowledgment was not received on time. (Timeout may eventually occur causing initiation of Notification of Failure; see “Handling Timeouts” below.)

1.3 HANDLING TIMEOUTS WHILE SENDING A BUSINESS MESSAGE

If all the timeouts and retries are exhausted while sending a business message, the Notification of Failure PIP should be initiated.

2 DTD PATHNAME SPECIFICATION

All XML documents should reference the associated DTD by specifying the doctype element. The name of the DTD file as published by RosettaNet should be specified. The doctype element should not specify any additional URL qualifiers that refer to a specific location where the DTD file exists. Recipients of RosettaNet XML messages are responsible for configuring their systems to find the appropriate DTD file.

For example, the following example is valid:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ServiceHeader SYSTEM "ServiceHeaderGuideline_v1_1.dtd">
<ServiceHeader>
  <ProcessControl>
```

The following examples are not valid:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ServiceHeader SYSTEM
"http://www.rosettanet.org/ServiceHeaderGuideline\_v1\_1.dtd">
<ServiceHeader>
  <ProcessControl>
```

and

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ServiceHeader SYSTEM
"file://localhost/c:/dtd/ServiceHeaderGuideline\_v1\_1.dtd">
<ServiceHeader>
  <ProcessControl>
```