

# Validating Messages and Error Handling

# Objectives

After completing this lesson, you should be able to:

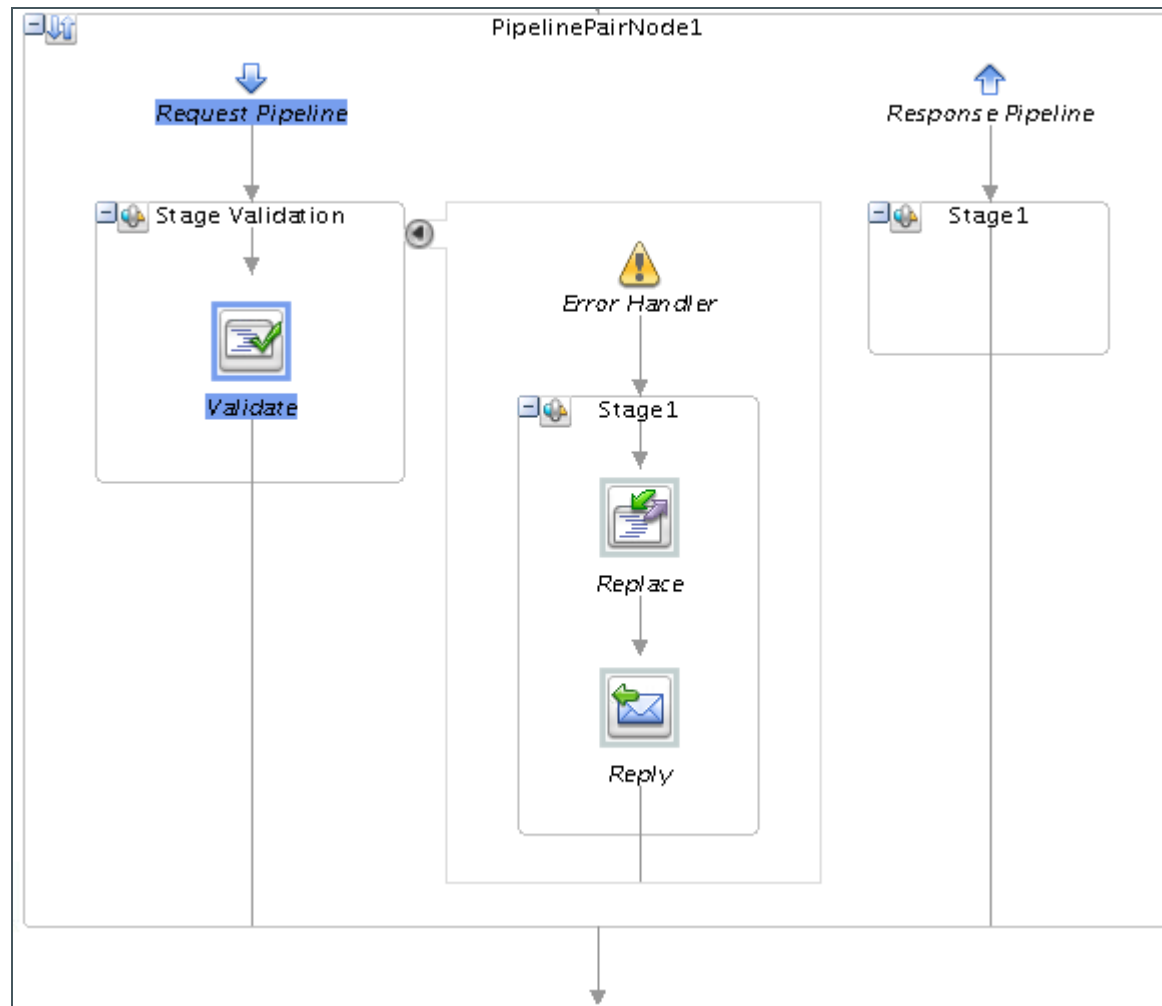
- Validate a message
- Describe basic fault handling and propagation
- Handle errors in Service Bus



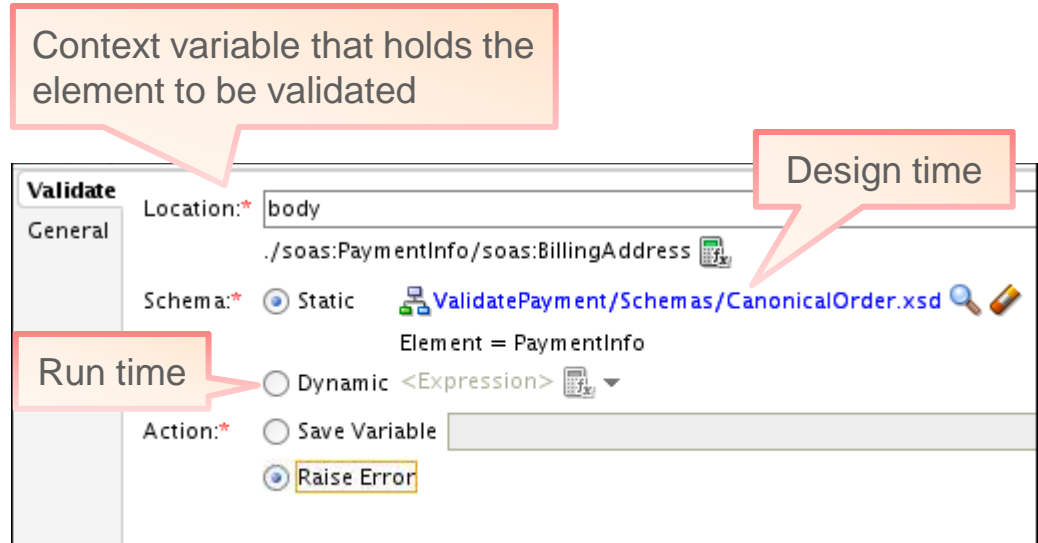
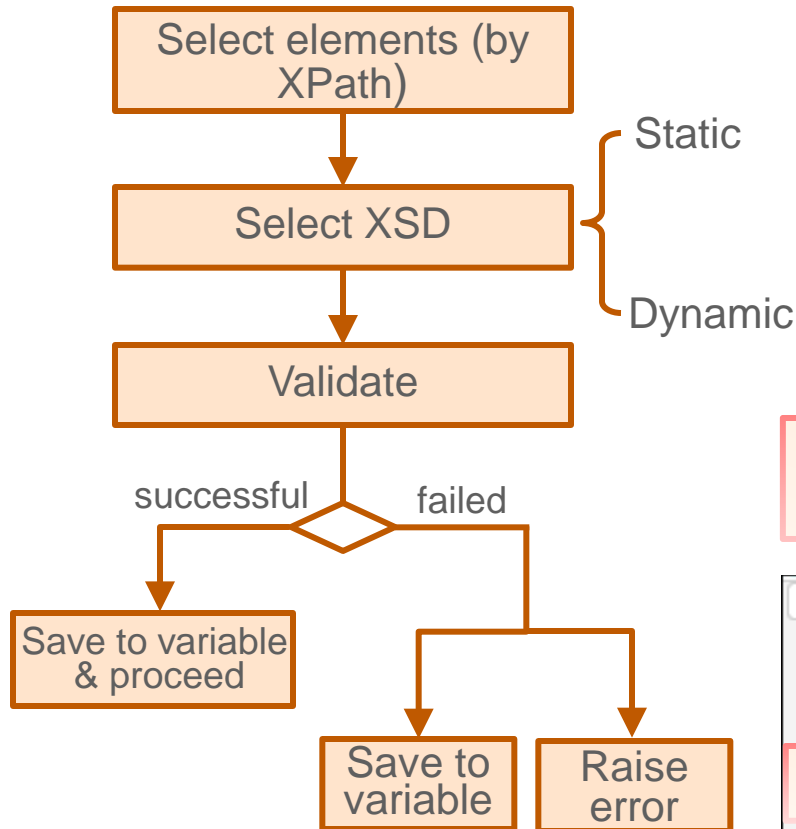
# Agenda

- Message validation
- Fault handling
- Reporting

# Message Validation



# Validate Action



# Dynamic Validation

Allows the user to select the resource (WSDL or schema) to validate against, based on the result of XQuery expression at run time.

- Example of dynamically specifying a schema resource:

```
<validate xmlns="http://www.bea.com/wli/sb/context">  
  <schema>default/MySchema</schema>  
  <schemaElement>  
    <localname>MyElementType</localname>  
  </schemaElement>  
</validate>
```

# Quiz



The Validate action validates elements selected by an \_\_\_\_\_ expression.

- a. XPath
- b. XQuery
- c. XSLT
- d. nXSD

# Agenda

- Message validation
- Fault handling
- Reporting



# Faults: Overview

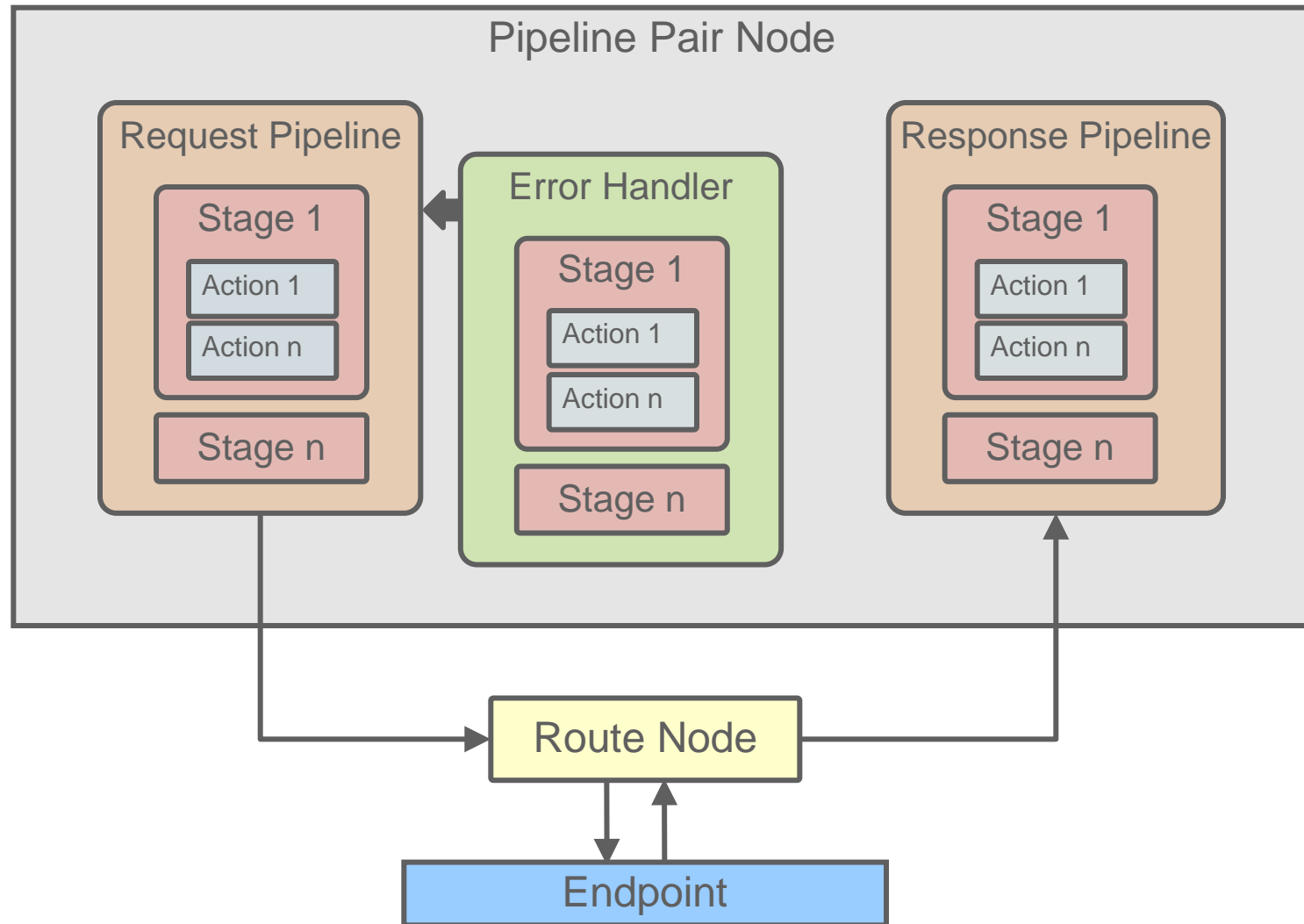
- Faults:
  - Errors encountered during the execution of an application
  - Similar to exceptions in Java
- Types of faults:
  - Business faults:
    - Exceptional business behavior: Caused by a failure to meet a certain business rule
    - Faulty operation by users: Caused by human errors
  - Runtime/technical faults:
    - Software errors: Caused by programming bugs in applications
    - Technical errors: Caused by problems in the infrastructure on which applications run

# Fault-Handling Challenges

The handling and propagation (passing on) of faults is a multi-dimensional problem. Strategies for fault handling must consider the following:

- Fault type
  - Business
  - Runtime (technical)
- Interaction type
  - Synchronous
  - Asynchronous
- Transactionality of the services involved

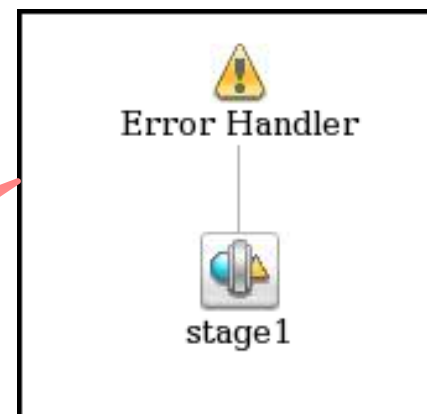
# Handling Errors in OSB



# Error Handlers

- An error handler is a special stage where a set of actions can occur if something goes wrong.
- The following error handler scopes are defined:
  - Stage
  - Route node
  - Pipeline pair (both Request Pipeline and Response Pipeline individually)
  - Pipeline service
- If no error handler is defined, there is a default system error handler.

All configuration for an error handler is in one or more stages contained by the error handler.

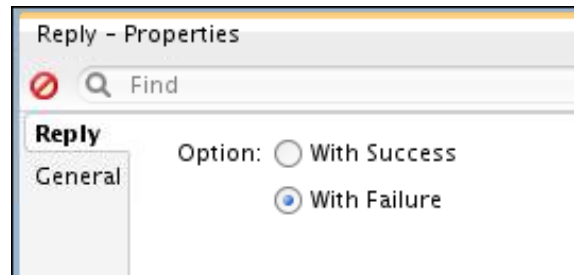


# `$fault` Variable

- A predefined context variable used to hold information about any error that occurs during message processing
- Populated before the appropriate error handler is invoked
- Includes the following subelements:
  - `errorCode`
  - `reason`
  - `details`
  - `location`
  - `java-exception`
  - `stack-trace`

# Reply Action

- Specify that an immediate reply be sent to the invoker
- Can be used in the request, response, or error pipeline
- Can be configured to result in a reply with success or failure



# Generating Error Messages Using Reply Action

The type of error returned depends on the transport used.

Transport Error Returned	
HTTP	HTTP 500
JMS	Sets the <code>JMS_BEA_Error</code> property to <code>true</code>
SOAP	SOAP fault

# Quiz



The smallest error handler scope you can define is at action level.

- a. True
- b. False






# Agenda

- Message validation
- Fault handling
- Reporting

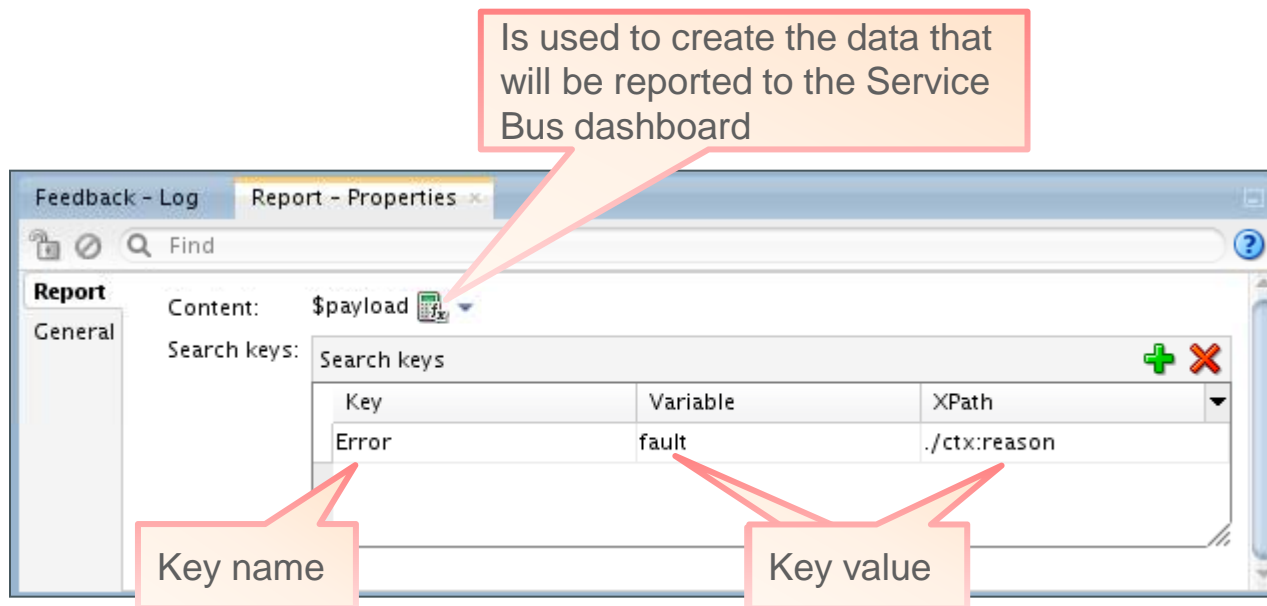
# Reporting Actions

- The Reporting Actions category has actions to log or report errors and generate alerts if required in a message flow in a stage.
- There are three types of actions:

	Report	Enable message reporting for a proxy service.
	Log	Construct a message to be logged.
	Alert	Send an alert notification based on pipeline message context.

# Report Action

The Report action allows you to extract information from each message and write it to the OSB Reporting Data Stream.



# Log Action

- Ensure that global and pipeline logging are enabled in EM.
- Configure log level at OSB and pipeline level.

Configuring logger for pipeline

**Operations**

Type	<input checked="" type="checkbox"/> State	<input type="checkbox"/> Monitoring	Aggregation Interval	<input checked="" type="checkbox"/> SLA Alerts	<input type="checkbox"/> Msg. Tracing	<input checked="" type="checkbox"/> Pipeline Alerts	<input checked="" type="checkbox"/> Logging	<input checked="" type="checkbox"/> Reports
Business Service	<input checked="" type="checkbox"/>	<input type="checkbox"/>	10 Mins	<input checked="" type="checkbox"/> (N)	<input type="checkbox"/> (T)	...	...	...
Pipeline	...	<input type="checkbox"/> (S)	10 Mins	<input checked="" type="checkbox"/> (N)	...	<input checked="" type="checkbox"/> (N)	<input checked="" type="checkbox"/> (D)	<input checked="" type="checkbox"/>
Proxy Service	<input checked="" type="checkbox"/>	<input type="checkbox"/>	10 Mins	<input checked="" type="checkbox"/> (N)	<input type="checkbox"/> (T)	...	...	...

**Tracing**

Execution Tracing ☐ Enabled

**Logging**

Logging ☒ Enabled

Logging Level **Debug** level or above

**Reporting**

**Log - Properties**

Log

General

Content: \$body

Summary:

Severity: Info

Debug

Info

Warning

Error

# Configuring OSB Log Level

The screenshot shows the Oracle Service Bus (OSB) configuration console. In the 'Target Navigation' pane on the left, the tree structure is expanded to 'service-bus (Default Service Bus)'. The 'Logs' option is highlighted in the left pane, and a context menu is open showing 'Log Configuration' as the selected option. The main pane displays the 'Log Levels' configuration page for 'Runtime Loggers'. A table lists various loggers, with the 'oracle.osb' section highlighted by a red box.

Logger Name	Oracle Diagnostic Logging Level (Java Level)	Log File	Persistent Log Level
oracle.integration	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb.configfwk	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb.debug.alert-manager	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb.debug.bpel	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb.debug.configfwk.component	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb.debug.configfwk.core	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb.debug.configfwk.deployment	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb.debug.configfwk.persistence	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb.debug.configfwk.security	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb.debug.configfwk.transaction	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	
oracle.osb.debug.configfwk.validation	WARNING:1 (WARNING) [Inherited from ...]	odl-handler	

# Alert Action

- Alert action is for business use or reporting errors, and not for monitoring system health.
- Pipeline alerting should be enabled for the service or at the domain level.

The image shows two overlapping windows from a management console. The background window is titled "Alert - Properties" and has a "General" tab selected. It contains the following fields:

- Content:** \$fault (with a small icon)
- Summary:** Business Service is down or under duress. Please check fault for details.
- Severity:** Critical
- Destination:** Shared58/AlertDestination.alert (with a magnifying glass icon)

A blue callout arrow points from the "Destination" field to a foreground window titled "Alert Destination". This window has a green warning icon and the title "Alert Destination". Below the title is the text "Create an Alert Destination". It contains the following fields:

- Description:** (empty text box)
- SNMP Trap:** ☐ Yes ☒ No
- Reporting:** ☐ Yes ☒ No
- Alert Logging:** ☒ Yes ☐ No
- E-Mail Recipients:** (empty text box with a green plus icon and a grey X icon)
- JMS Destinations:** (empty text box with a green plus icon and a grey X icon)

# Summary

In this lesson, you should have learned how to:

- Validate a message
- Describe basic fault handling and propagation
- Handle errors in Service Bus



# Practice 5: Overview

- 5-1: Importing template resources and creating a business service
- 5-2: Creating the pipeline that validates messages using the Pipeline template