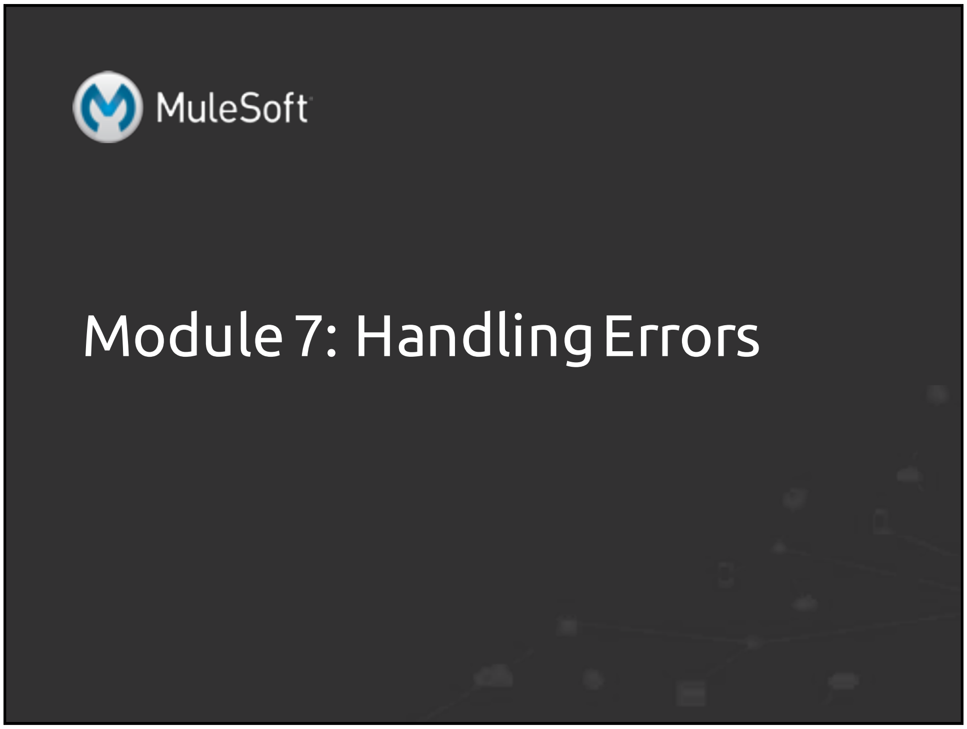
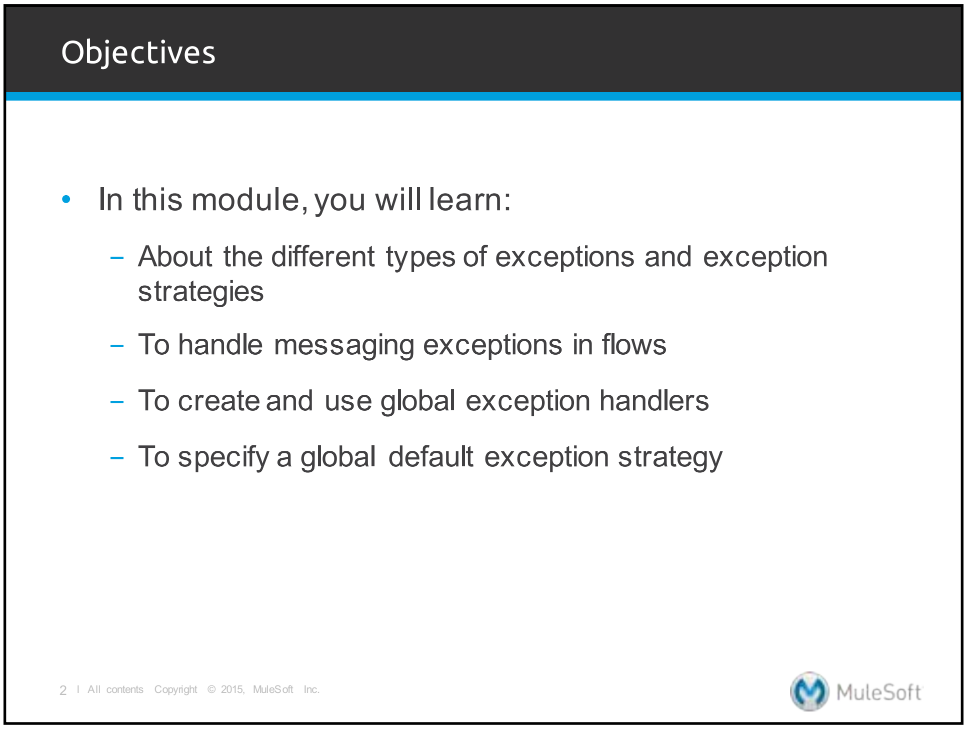


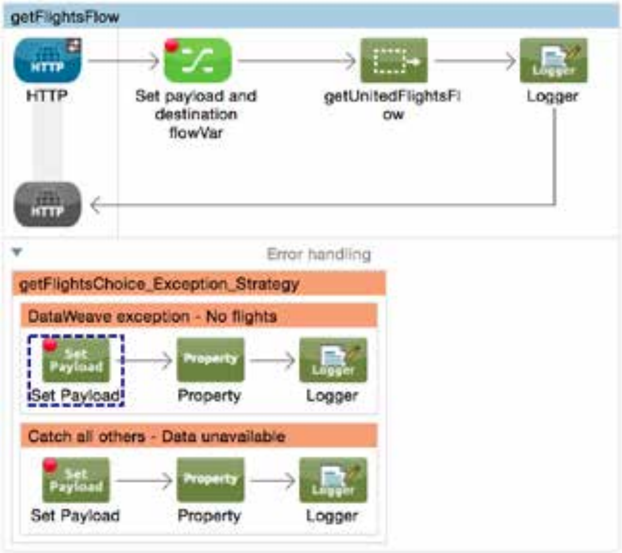
Module 7: Handling Errors



Objectives

- In this module, you will learn:
  - About the different types of exceptions and exception strategies
  - To handle messaging exceptions in flows
  - To create and use global exception handlers
  - To specify a global default exception strategy

Goal

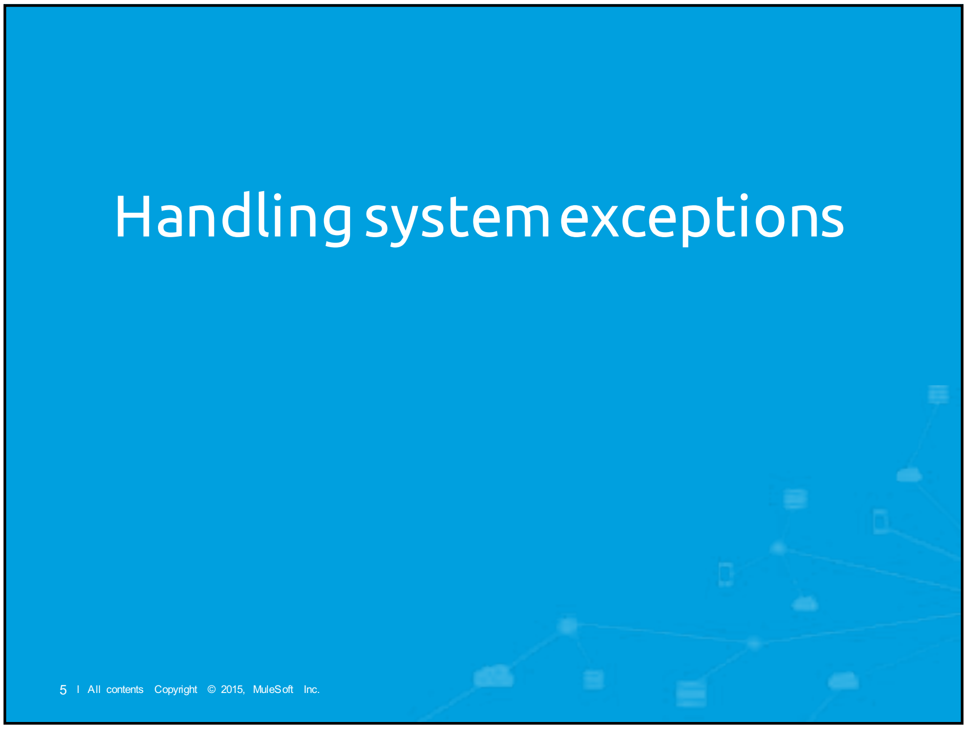

getFlightsFlow
HTTP
Set payload and destination flowVar
getUnitedFlightsFlow
Logger
Error handling
getFlightsChoice_Exception_Strategy
DataWeave exception - No flights
Set Payload
Property
Logger
Catch all others - Data unavailable
Set Payload
Property
Logger

MuleSoft

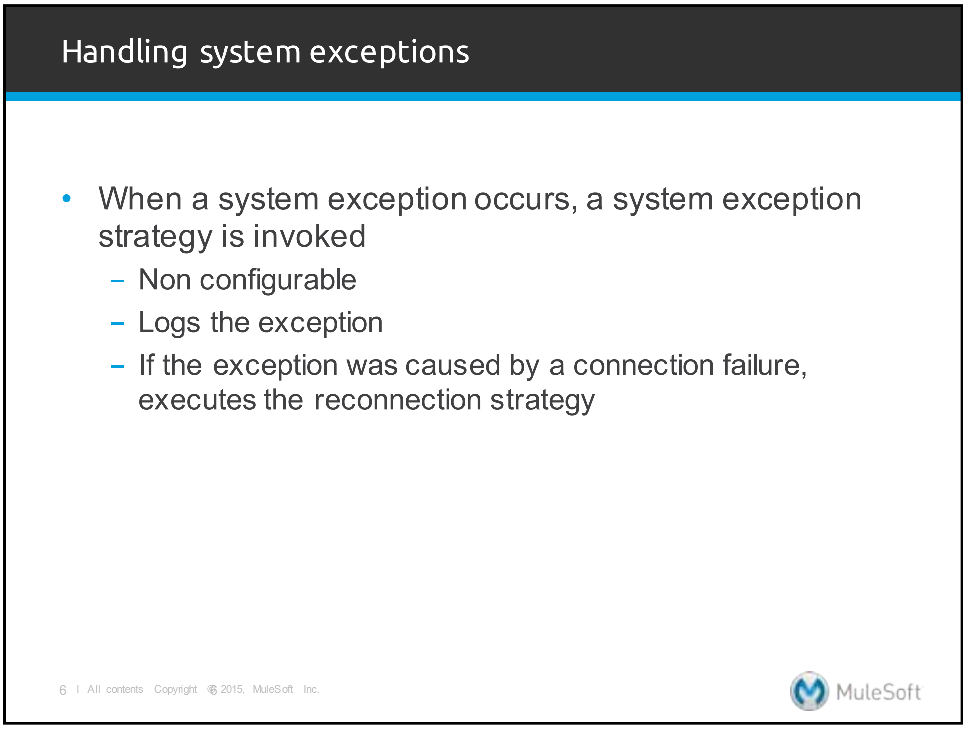
Types of exceptions

- System exceptions
  - Thrown at the system-level when *no* message is involved
  - Exceptions that occur
    - During application start-up
    - When a connection to an external system fails
- Message exceptions
  - Thrown within a flow whenever a message is involved

MuleSoft



Handling system exceptions



Handling system exceptions

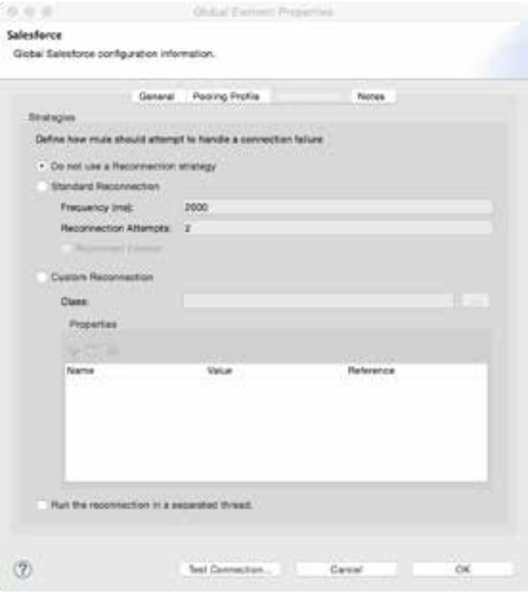
- When a system exception occurs, a system exception strategy is invoked
  - Non configurable
  - Logs the exception
  - If the exception was caused by a connection failure, executes the reconnection strategy

Reconnection strategies

- Set for each connector
- Some connectors have reconnection options in the Global Element properties GUI
- For most others, you set connector properties in XML
  - Set attempt count and frequency in ms
  - Set it to blocking or non-blocking
  - Notify registered listeners

```
<jms:activemq-connector name="AMQConnector”>
  <reconnect count="5” frequency="1000"/>
</jms:activemq-connector>
```

Setting reconnection properties



Global Element Properties

Salesforce

Global Salesforce configuration information.

Strategies

Define how mule should attempt to handle a connection failure

● Do not use a Reconnection strategy

○ Standard Reconnection

Frequency (ms): 2000

Reconnection Attempts: 2

○ Custom Reconnection

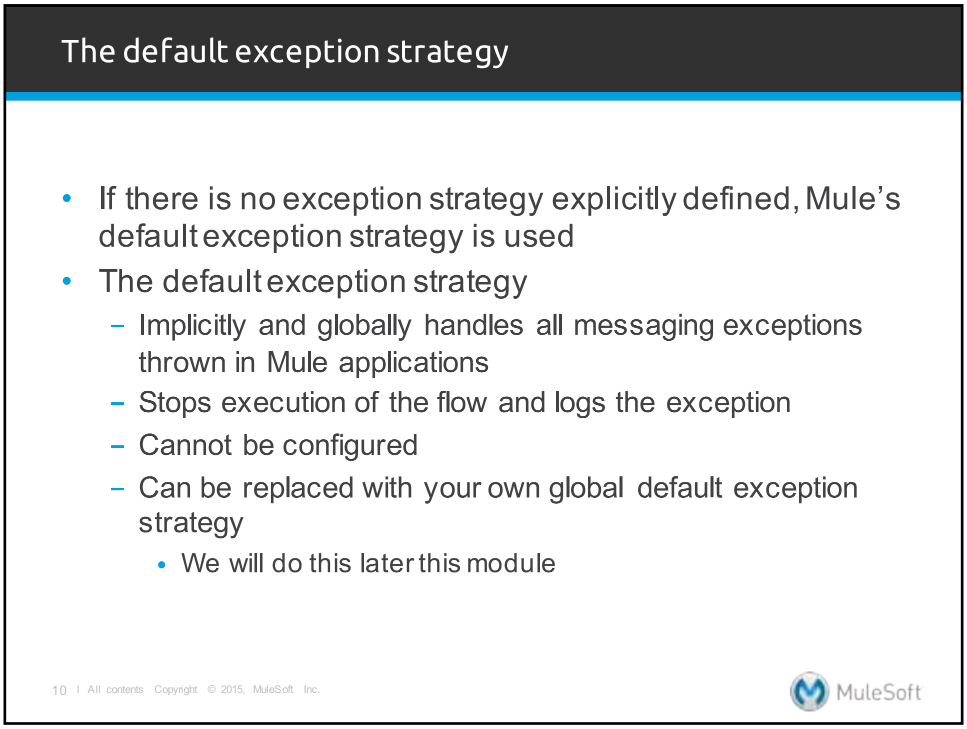
Class:

Properties

Name	Value	Reference

□ Run the reconnection in a separated thread.

Handling messaging exceptions

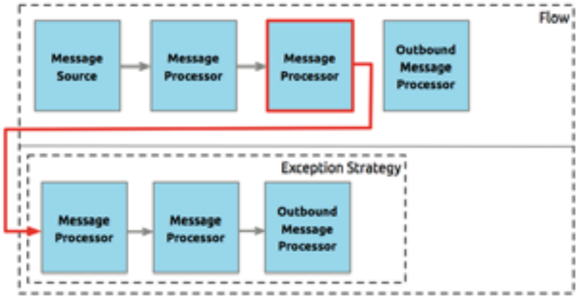


The default exception strategy

- If there is no exception strategy explicitly defined, Mule's default exception strategy is used
- The default exception strategy
  - Implicitly and globally handles all messaging exceptions thrown in Mule applications
  - Stops execution of the flow and logs the exception
  - Cannot be configured
  - Can be replaced with your own global default exception strategy
    - We will do this later this module

Handling messaging exceptions

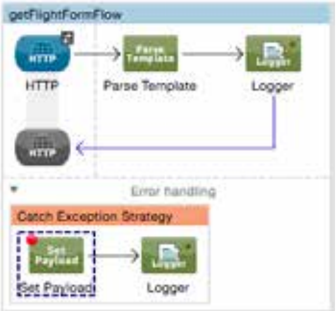
- When a message being processed through a Mule flow throws an exception
  - Normal flow execution stops
  - The message is passed to the first processor in the exception strategy

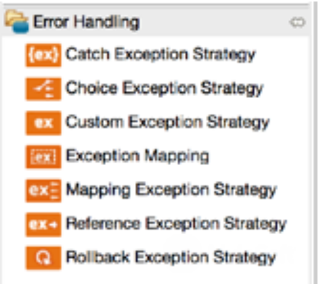


MuleSoft

Defining messaging exception strategies

- Exception strategies are added to the error handling section of a flow
- Each flow can contain only one exception strategy
  - Choice exception strategies can contain one or more catch and/or rollback exception strategies
- Each exception strategy can contain any number of message processors

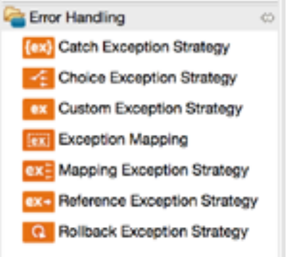



Error Handling
Catch Exception Strategy
Choice Exception Strategy
Custom Exception Strategy
Exception Mapping
Mapping Exception Strategy
Reference Exception Strategy
Rollback Exception Strategy

Referencing the exception inside the strategy

- Inside an exception strategy, you can reference the exception object
  - <logger level="ERROR" message="#[exception]"/>
- Use methods to get different amounts of detail about it
  - exception.getSummaryMessage()
  - exception.getVerboseMessage()
  - exception.getMessageCode()
  - exception.getDetailedMessage()

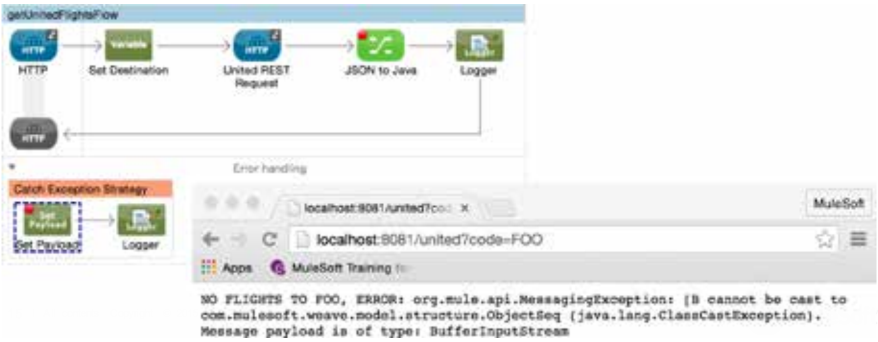
Exception strategies


Error Handling
Catch Exception Strategy
Choice Exception Strategy
Custom Exception Strategy
Exception Mapping
Mapping Exception Strategy
Reference Exception Strategy
Rollback Exception Strategy

- Catch
  - Catches exceptions based on conditions
- Rollback
  - Rolls back message for reprocessing
- Choice
  - Selects one of multiple catch and/or rollback strategies based on conditions
- Reference
  - Lets you reference globally defined strategies
- Mapping
  - Sets HTTP status codes based on exception type
- Custom
  - Lets you specify a custom class for handling the exception

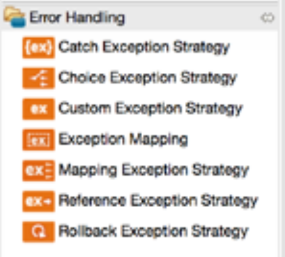
Walkthrough 7-1: Handle a messaging exception

- In this walkthrough, you will:
  - Add a Catch Exception Strategy to a flow
  - Catch the exception and send an error message back
  - Reference the exception object inside an exception handler
  - Create and catch a web service request error



```
NO FLIGHTS TO FOO, ERROR: org.mule.api.MessagingException: [B cannot be cast to
com.mulesoft.weave.model.structure.ObjectSeq (java.lang.ClassCastException).
Message payload is of type: BufferInputStream
```

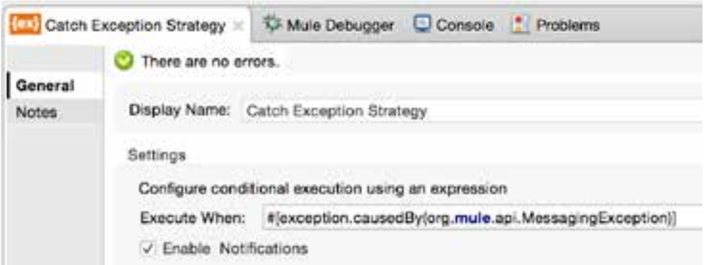
The Choice Exception Strategy


Error Handling
Catch Exception Strategy
Choice Exception Strategy
Custom Exception Strategy
Exception Mapping
Mapping Exception Strategy
Reference Exception Strategy
Rollback Exception Strategy

- The choice exception strategy must contain two or more catch and/or rollback strategies
- The individual catch and/or rollback strategies specify a condition for when they should be evaluated
- The choice strategy routes the message to the *first* exception strategy whose condition evaluates to true
- If none of its exception strategies can handle the error, the message is routed to Mule's default exception strategy

Setting exception strategy conditions

- Can reference the message or the exception
  - exception.causedBy(org.mule.example.ExceptionType)
  - exception.causedExactlyBy(org.mule.example.ExceptionType)
  - exception.causeMatches('org.mule.example.*')



There are no errors.

Display Name: Catch Exception Strategy

Settings

Configure conditional execution using an expression

Execute When: #[exception.causedBy(org.mule.api.MessagingException)]

☑ Enable Notifications

Return status codes

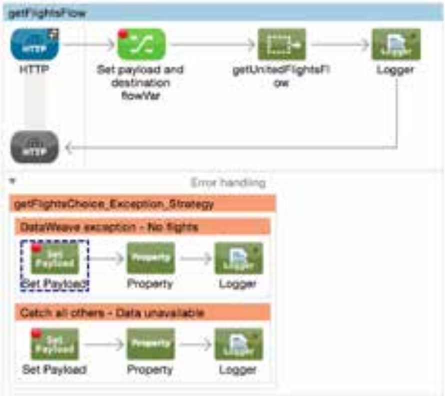
- By default, when a message is handled, an http status code of 200 is set and returned
- You can change this by setting outbound property
  - `message.outboundProperties.'http.status'`
- You can also use the Mapping Exception Strategy

Bubbling exceptions

- All flows can have their own exception strategies
- If a flow does not have an exception strategy, the strategy of the calling flow is used

Walkthrough 7-2: Handle a messaging exception

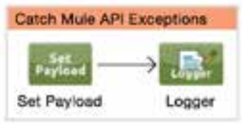
- In this walkthrough, you will:
  - Add and configure a Choice Exception Strategy
  - Set HTTP status codes in the exception handler
  - Let an exception bubble up and be handled by the calling flow



Defining global exception strategies

Defining global exception strategies

- You can reuse exception handling strategies by defining them outside a flow
  - You can drag them out and drop them outside any flow
  - Typically, put them in your global configuration file

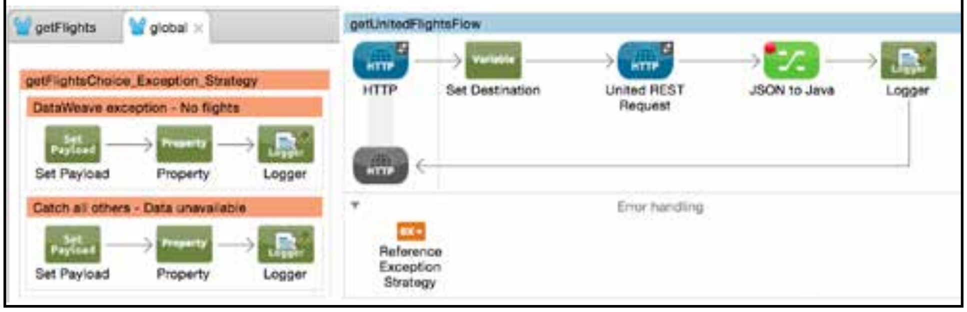


- Reference global exception handlers using the Reference Exception Strategy

EX
Reference Exception Strategy

Walkthrough 7-3: Create and use global exception handlers

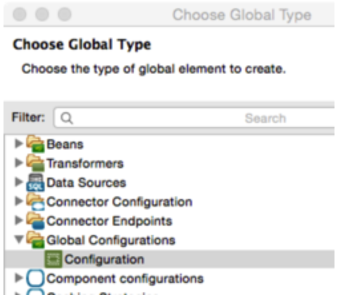
- In this walkthrough, you will:
  - Create a global exception handler
  - Reference and use the global exception handler in flows



Defining a default exception strategy

The default exception strategy

- Recall there is a default exception strategy
  - Implicitly handles all unhandled messaging exceptions
  - Simply logs the exception
  - Cannot be configured
- Can be replaced with your own global default exception strategy
  - Create a global configuration element
  - Specify a default exception strategy in the global configuration element



Choose Global Type

Choose Global Type

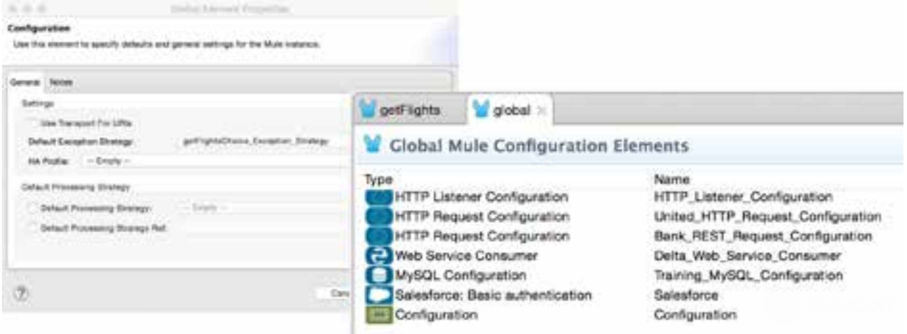
Choose the type of global element to create.

Filter: Search

- ▶ Beans
- ▶ Transformers
- ▶ Data Sources
- ▶ Connector Configuration
- ▶ Connector Endpoints
- ▼ Global Configurations
  - Configuration
- ▶ Component configurations

Walkthrough 7-4: Specify a global default exception strategy

- In this walkthrough, you will:
  - Create a global configuration element in the global.xml file
  - Specify a default exception strategy in the global configuration element
  - Remove the existing exception handling strategies
  - Use the default exception handling strategy



Global Element Properties

Configuration

Use this element to specify defaults and general settings for the Mule instance.

General | Notes

Settings

□ Use Transport For URIs

Default Exception Strategy: getFlightsChoice_Exception_Strategy

HA Profile: -- Empty --

Default Processing Strategy

○ Default Processing Strategy: -- Empty --

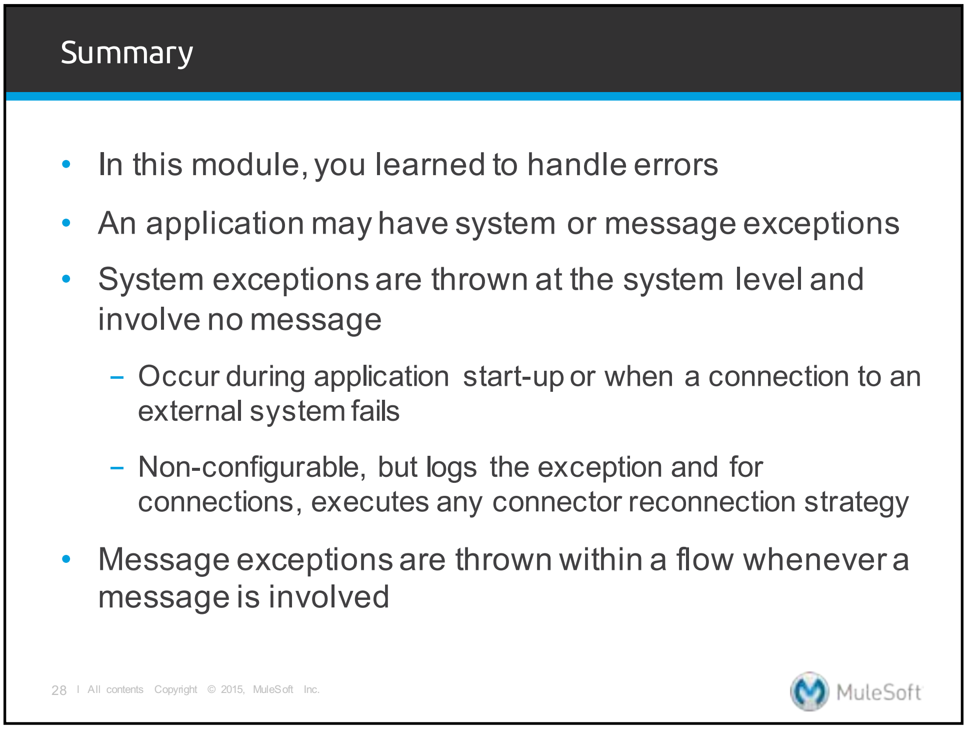
○ Default Processing Strategy Ref:

getFlights | global

Global Mule Configuration Elements

Type	Name
HTTP Listener Configuration	HTTP_Listener_Configuration
HTTP Request Configuration	United_HTTP_Request_Configuration
HTTP Request Configuration	Bank_REST_Request_Configuration
Web Service Consumer	Delta_Web_Service_Consumer
MySQL Configuration	Training_MySQL_Configuration
Salesforce: Basic authentication	Salesforce
Configuration	Configuration

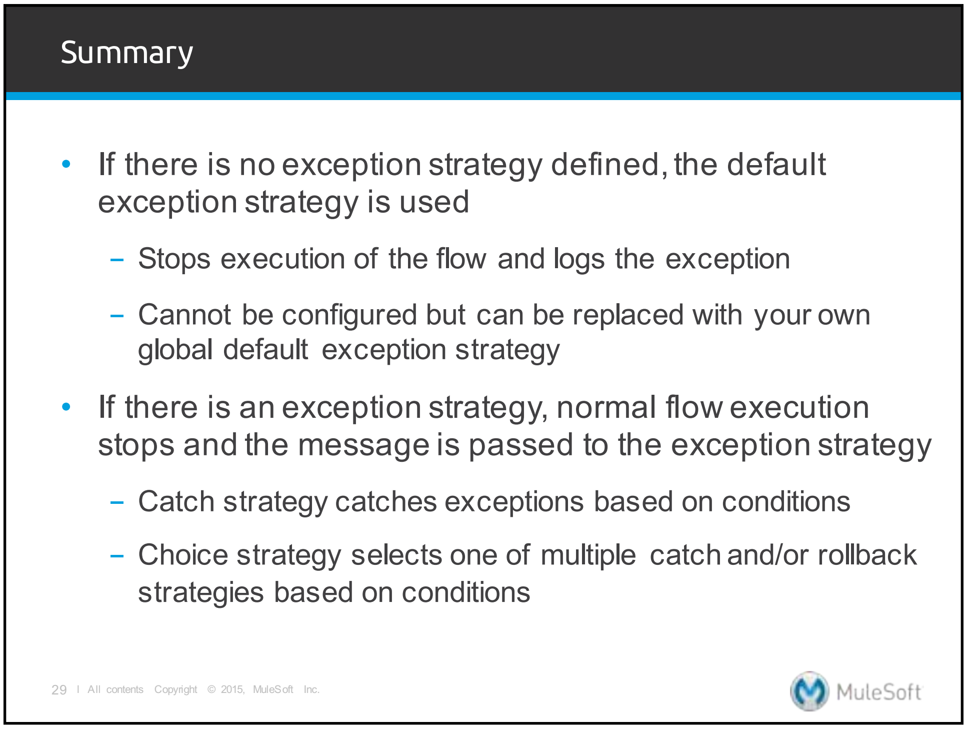
Summary



Summary

- In this module, you learned to handle errors
- An application may have system or message exceptions
- System exceptions are thrown at the system level and involve no message
  - Occur during application start-up or when a connection to an external system fails
  - Non-configurable, but logs the exception and for connections, executes any connector reconnection strategy
- Message exceptions are thrown within a flow whenever a message is involved

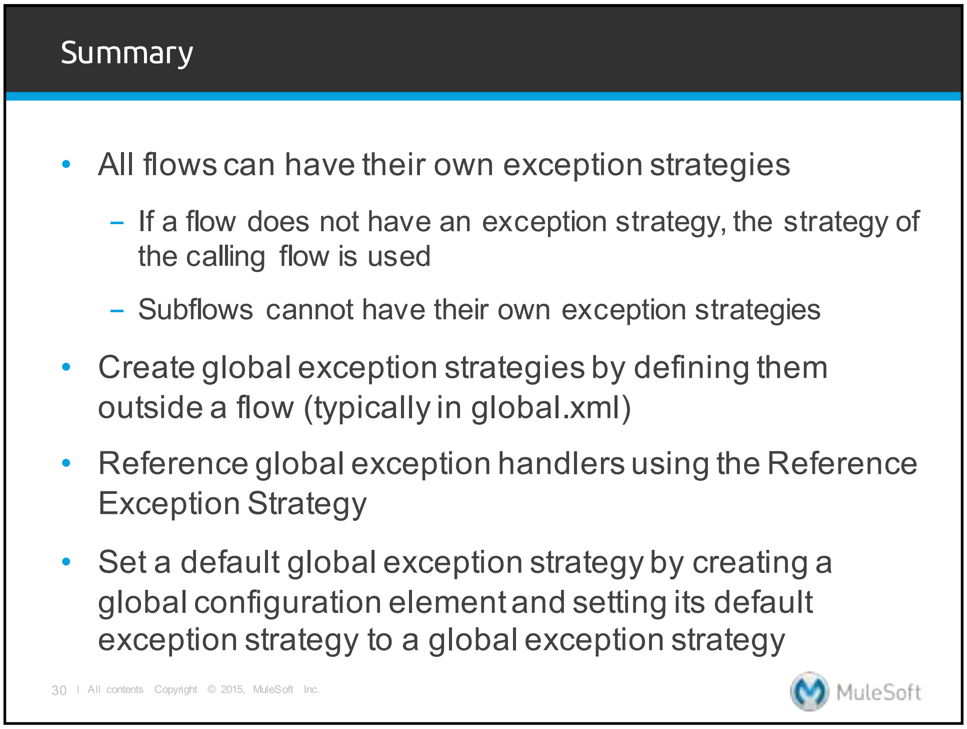
MuleSoft



Summary

- If there is no exception strategy defined, the default exception strategy is used
  - Stops execution of the flow and logs the exception
  - Cannot be configured but can be replaced with your own global default exception strategy
- If there is an exception strategy, normal flow execution stops and the message is passed to the exception strategy
  - Catch strategy catches exceptions based on conditions
  - Choice strategy selects one of multiple catch and/or rollback strategies based on conditions

MuleSoft



Summary

- All flows can have their own exception strategies
  - If a flow does not have an exception strategy, the strategy of the calling flow is used
  - Subflows cannot have their own exception strategies
- Create global exception strategies by defining them outside a flow (typically in global.xml)
- Reference global exception handlers using the Reference Exception Strategy
- Set a default global exception strategy by creating a global configuration element and setting its default exception strategy to a global exception strategy

MuleSoft