


Slide 1

WebSphere Education

IBM

Handling errors in a service policy



© Copyright IBM Corporation 2013
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Slide 2

WebSphere Education

IBM

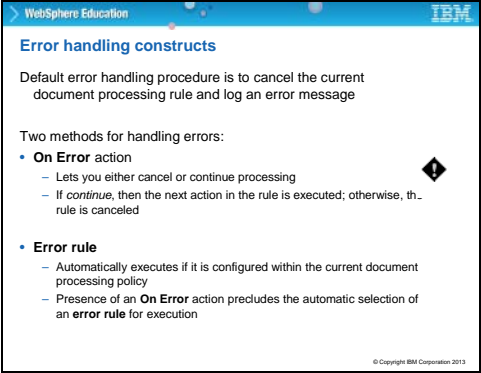
Unit objectives

After completing this unit, you should be able to:

- Configure an On Error action in a service policy
- Configure an Error rule in a service policy
- Describe how On Error actions and Error rules are selected during error handling

© Copyright IBM Corporation 2013

Slide 3



WebSphere Education

Error handling constructs

Default error handling procedure is to cancel the current document processing rule and log an error message

Two methods for handling errors:

- **On Error action**
 - Lets you either cancel or continue processing
 - If *continue*, then the next action in the rule is executed; otherwise, the rule is canceled
- **Error rule**
 - Automatically executes if it is configured within the current document processing policy
 - Presence of an **On Error** action precludes the automatic selection of an **error rule** for execution

© Copyright IBM Corporation 2013

Error handling constructs

If you do not give any indication as to how you want errors to be handled, the default action is to stop processing. You can modify the processing in two ways, which are not exclusive. You can add an action to a rule, or you can write a rule that is designated as an error rule. Remember that there are four types of rule: request, response, both directions, and error. The priority is for the error action. If you have one set in a rule, the error rule is not called unless the action makes the call specifically.

Slide 4

WebSphere Education

Configure an On Error action

The **On Error** action is used to control what happens when an error is encountered within the rule

- Optional: execute a named rule to handle the error condition

Configure the following within an **On Error** action:

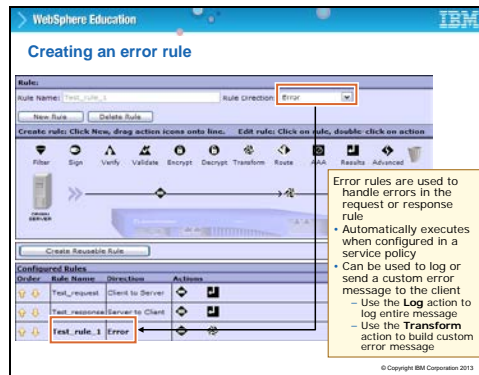
- Error mode:
 - Cancel**: stop executing the current rule
 - Alternative**: invoke an alternative processing rule
 - Continue**: continue with the next sequential action
- The **Processing Rule** fields specify either:
 - An error rule to execute
 - A custom variable for the processing rule
 - Use the Var Builder to create a custom variable

© Copyright IBM Corporation 2013

Configure an On Error action

So how do you configure the On Error action? An On Error is one of the 'Advanced' actions. You drag the icon to the rule bar, double-click it, and choose the action that you want. When you select 'On Error', the dialog you see on the slide opens. Your first task is to decide what mode you require. 'Cancel' is the equivalent of stop processing and does stop all processing. 'Continue' says ignore the error and keep going. You would use the processing for circumstances where the error was anticipated and of no consequence. The most interesting mode is 'Alternative'. Alternative allows you to specify another rule to call if an error occurs. You can choose the rule in the second field, 'Processing Rule', and then indicate the context for the input (that is to say, what the input variable is) and likewise for the output. You can choose the context that is called OUTPUT for the error output, which returns the error message.

Slide 5



Creating an error rule

The diagram shows how to build an error rule. You click 'New Rule', give the rule a name, and choose the direction to be 'Error'. You can then specify a style sheet to create an error message by adding a transform action, or add a log action that can take the entire input message and write it out. You can define many error rules for different types of error by setting the match action to accept different error codes. You might have a single error code to match on, or a bunch of codes. The next slide looks at how to call a style sheet.

Slide 6

WebSphere Education

Configure Transform action in error rule

• Use the **Transform** action to build custom error messages in an error rule

- Transforms error messages that are generated by the appliance into custom error messages

Configure Transform Action

Basic | Advanced

Input: INPUT

Transform

Use Document Processing Instructions

☐ Use XSLT specified in this action on a non-XML message
☒ Use XSLT specified in this action
☐ Use XSLT specified in XML document processing instruction

Processing Control File: local:///custom-error.xml

URL Resource Policy: None

Asynchronous: on


Output: OUTPUT

© Copyright IBM Corporation 2013

Configure Transform action in error rule

A transform action is added to the error rule line. Specify where the input is coming from, what transformation you want to do, and where the output is going. On the screen capture on the slide, the input is the INPUT context variable, in other words, what ever message came into the rule. The effected transformation might be on an XML message or a non-XML message. If you choose this second alternative, you might want to go out to WebSphere Transformation Extender to build the output, and so extra fields are added to the dialog that allow you to choose a WebSphere Transformation Extender map file.

You have several alternatives for the output. The one shown in the slide indicates that you want the result of this action that passed back to the client. You can also specify Null, where nothing is output (you might log the error), or a temporary variable that can be passed to another action. You might specify PIPE, where the output of this action is the input to the next. More on PIPE is presented later.

WebSphere Education


Style sheet programming that use error variables

- Output log messages with log priority that use `<xsl:message>`

```

<xsl:message
  dp:type='ws-proxy' dp:priority='error'>
  Error: <xsl:value-of select="$errtest"/>
</xsl:message>

```
- The following DataPower variables are useful when generating a custom error message
 - `var://service/error-code` DataPower error code
Example: Dynamic execution error
 - `var://service/error-subcode` DataPower suberror code
Example: Schema validation error
 - `var://service/error-message`
Error message sent to client
 - `var://service/transaction-id`
ID used to correlate transactions in the DataPower system logs
 - `var://service/client-service-address`
Address of the calling client

© Copyright IBM Corporation 2013

Style sheet programming by using error variables

When you invoke a style sheet to build an error message output, you get something like the error. DataPower gives you variables to add a type to the message (in this case, the type is defined as 'ws-proxy') and a priority. Then, you can pick up the value of a variable here the example is a variable that is called 'errtest'). There are also a number of service variables. (You saw service variables in the previous unit.) From the variables, you can pick up the error code and message, the ID of the transaction, or the client service address.

Here is an important point to note: if you have a schema validation error in a rule, and that rule invokes an error rule, the schema validation error code becomes the error subcode. The error code is, for example, a dynamic execution error code. You can then pick up the schema validation error code through the service variable.

Slide 8

WebSphere Education

IBM

Example custom error style sheet

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:dp="http://www.datapower.com/extensions"
  extension-element-prefixes="dp" exclude-result-prefixes="dp">

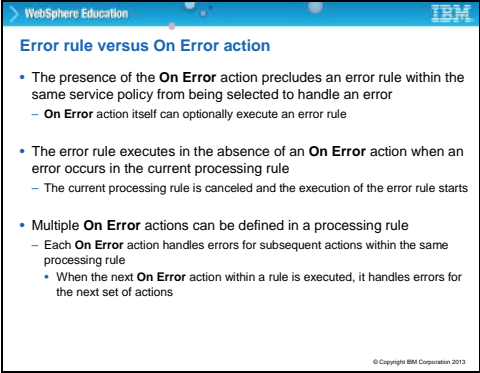
  <xsl:template match="/">
    <!-- Get the error codes set by DP. -->
    <xsl:variable name="dpErrorCode" select=
      "dp:variable('var://service/error-code')"/>
    <xsl:variable name="dpErrorSubcode" select=
      "dp:variable('var://service/error-subcode')"/>
    <xsl:variable name="dpErrorMessage" select=
      "dp:variable('var://service/error-message')"/>
    <xsl:variable name="dpTransactionId" select=
      "dp:variable('var://service/transaction-id')"/>

    <!-- Build custom SOAP fault message -->
    <env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
      <env:Body>
        <env:Fault (details omitted) ... </env:Fault>
      </env:Body>
    </env:Envelope>
  </xsl:template>
</xsl:stylesheet>
```

© Copyright IBM Corporation 2013

Example custom error style sheet

This slide shows how the error style sheet might be constructed. The template match is the root of the document to which the style sheet is being applied. The values of the four service variables you saw on the previous slide are dropped into local variables. You are not required drop the values into local variables, but it is clearer coding if you do. You then build a SOAP message and use the variables in the Fault tag, probably together with some text to explain what each value refers to.



WebSphere Education

Error rule versus On Error action

- The presence of the **On Error** action precludes an error rule within the same service policy from being selected to handle an error
 - **On Error** action itself can optionally execute an error rule
- The error rule executes in the absence of an **On Error** action when an error occurs in the current processing rule
 - The current processing rule is canceled and the execution of the error rule starts
- Multiple **On Error** actions can be defined in a processing rule
 - Each **On Error** action handles errors for subsequent actions within the same processing rule
 - When the next **On Error** action within a rule is executed, it handles errors for the next set of actions

© Copyright IBM Corporation 2013

Error rule versus On Error action

Finally, here are a few guidelines to be aware of when writing error handling in rules. Error rules are invoked either if there is no on error action in a request or a response rule, or if the on error action itself calls the error rule as an 'Alternative' processing path. If there is no 'On Error' action that is defined, when an error occurs current processing terminates and the process looks for an error rule that matches the error that occurred. If it finds one, it runs the rule; otherwise, the processing terminates. Finally, an 'On Error' action is only current after it is defined. The definition means that if you have a series of actions {A, 'On Error', B, C}, the 'On Error' applies to B and C. If there is an error in action A, processing terminates. The 'On Error' action is valid from the point it is defined until another 'On Error' is defined. So, for example, in the action sequence {A, 'On Error1', B, 'On Error2', C}, 'On Error1' applies only to action B. After that, 'On Error2' overrides 'On Error1'.

Slide 10

WebSphere Education

IBM

Unit summary

Having completed this unit, you should be able to:

- Configure an On Error action in a service policy
- Configure an Error rule in a service policy
- Describe how On Error actions and Error rules are selected during error handling

© Copyright IBM Corporation 2013

Slide 11

WebSphere Education

IBM

Checkpoint questions

1. True or False: When a rule with an **On Error** action encounters an error, the rule is always terminated.
2. True or False: An error rule is unidirectional.
3. A service policy has an error rule and a request rule with an **On Error** action. How does the firmware select the error-handling option?
 - A. The **On Error** radio button is selected from the admin setup page.
 - B. The firmware does not select the error-handling option. Selecting the error-handling option is an off-appliance function.
 - C. If the **On Error** action is already encountered, error processing goes the **On Error** action. If the **On Error** action is not encountered, the error rule gets control.
 - D. None of items A, B, and C
 - E. All of items A, B, and C

© Copyright IBM Corporation 2013

Slide 12

WebSphere Education

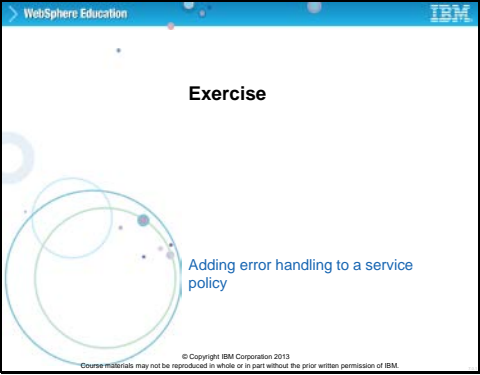
IBM

Checkpoint answers

1. **False.** Continuation of the current rule depends on the setting of Error Mode.
2. **False.** An error rule is active for both request and response rules.
3. **C.** A service policy has an error rule and a request rule with an **On Error** action. How does the firmware select the error-handling option?
 - A. The **On Error** radio button is selected from the admin setup page.
 - B. The firmware does not select the error-handling option. Selecting the error-handling option is an off-appliance function.
 - ✓ **C.** If the **On Error** action is already encountered, error processing goes to the **On Error** action. If the **On Error** action is not encountered, the error rule gets control.
 - D. None of items A, B, and C
 - E. All of items A, B, and C

© Copyright IBM Corporation 2013

Slide 13



WebSphere Education

IBM

Exercise

Adding error handling to a service policy

© Copyright IBM Corporation 2013
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

Slide 14

WebSphere Education

IBM

Exercise objectives

After completing this exercise, you should be able to:

- Configure a service policy with an On Error action
- Configure a service policy with an Error rule

© Copyright IBM Corporation 2013

Slide 15

