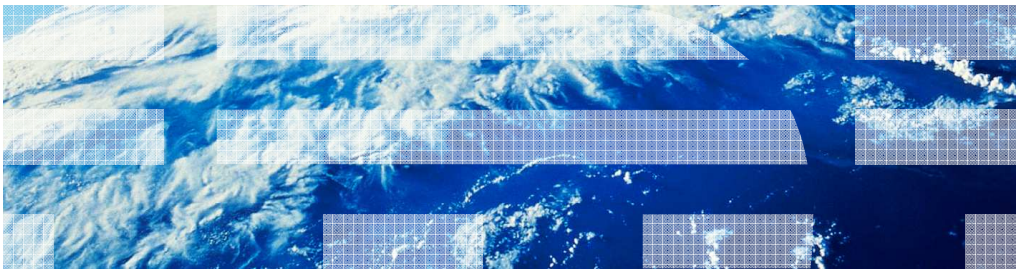
Business Process Management
IBM Business Process Manager V8.0

Introduction to enhanced error handling and termination handling



One of the important considerations when modeling business processes is how to handle errors that might occur when processes are running. IBM Business Process Manager version 8.0 has enhanced it's error handling capabilities. This presentation will give you an introduction to these enhanced capabilities. The presentation will also go over the improvement done in the termination handling functionality in version 8.0.

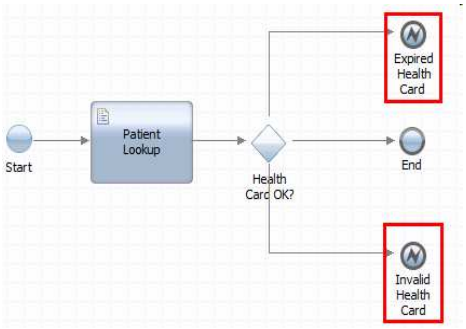
Agenda

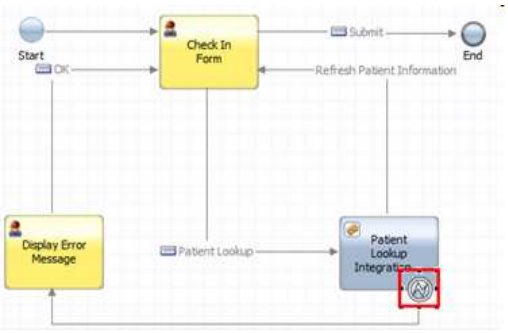
- What is the enhanced error handling behavior?
- How to define the enhanced error handling behavior?
- What is the enhanced termination behavior?
- Summary

You will learn what the enhanced error handling behavior is and how to define the behavior in the Process Designer. You will also learn about what the termination behavior enhancement is compared to versions before 8.0.

Error handling behavior - background

- In Business Process Definitions
  - Error End Event to create an error
  - Boundary Error Intermediate Event to catch an error
  - Error Event Subprocess to catch an error
    - Interrupts the parent process flow
- In Services
  - Error End Event to create an error
  - Boundary Error Intermediate Event to catch an error
  - Flow Error Intermediate Event to catch an error





3 IBM Business Process Manager V8.0 – Enhanced error and termination handling

In a business process definition, you can use an error end event to create an error.

To catch an error, you can attach an error intermediate event to an activity when you want to take some action as a result of an error occurrence and then continue with the process flow normally.

But, if you want to take some action as a result of an error occurrence and end the main process, then you should use an Error Event Subprocess to catch an error. You can model the action to be taken in the Error Event Subprocess. Once the Error Event Subprocess ends, the main process also ends.

In a service, you can use an error end event to throw an error and attach a boundary error intermediate event to a step to catch an error. A service can have an error intermediate event in the flow also.

Error handling behavior

New for V8

- Ability to catch specific errors in business process definitions and in services
  - Based on error code, error data or both
  - Error code/error data can be defined for
    - *Error End Event*
    - *Error Intermediate Event* attached to a boundary
    - *Error Start Event* in an Event Subprocess
- Can attach multiple boundary *Error Intermediate Events* to an activity or a step
  - Each one with unique error code/error data combination (signature)
- Pre 8.0 models will continue to catch all exceptions when imported into 8.0 environment
  - Need to re-create error events to use 'catch specific error' behavior

In versions before 8.0, an error event attached to an activity or a step catches all errors. These attached error events do not provide the level of granularity that allows you to differentiate between different types of errors that can be originated.

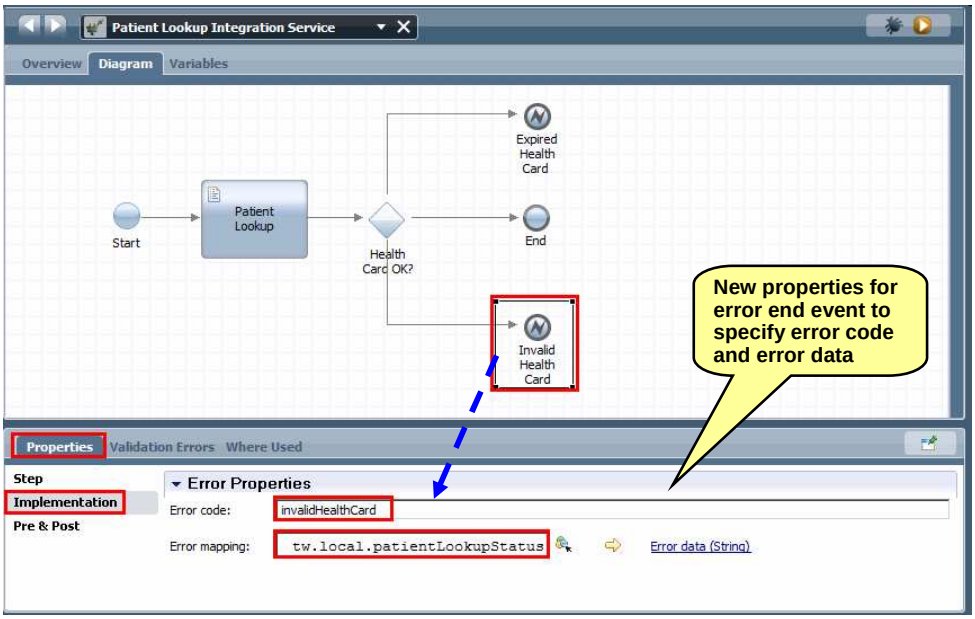
Version 8.0 gives you an ability to catch specific errors in business process definitions and in services. An error can be caught based on an error code, an error data or a combination of both. Error code and error data are new properties that can be defined on an error end event, an error intermediate event attached to an activity or a step and error start event in an event sub process.

An activity in a business process definition or a step in a service can have multiple boundary error intermediate events attached to it. Each of the events attached to a single activity or a step catches a different error. That is, a unique error code, unique error data type or a unique combination of error code and error data type must be specified for each error intermediate event attached to a single activity or a step.

An error intermediate event in a service flow continues to catch all errors.

The next two slides show an example of these events and the new properties.

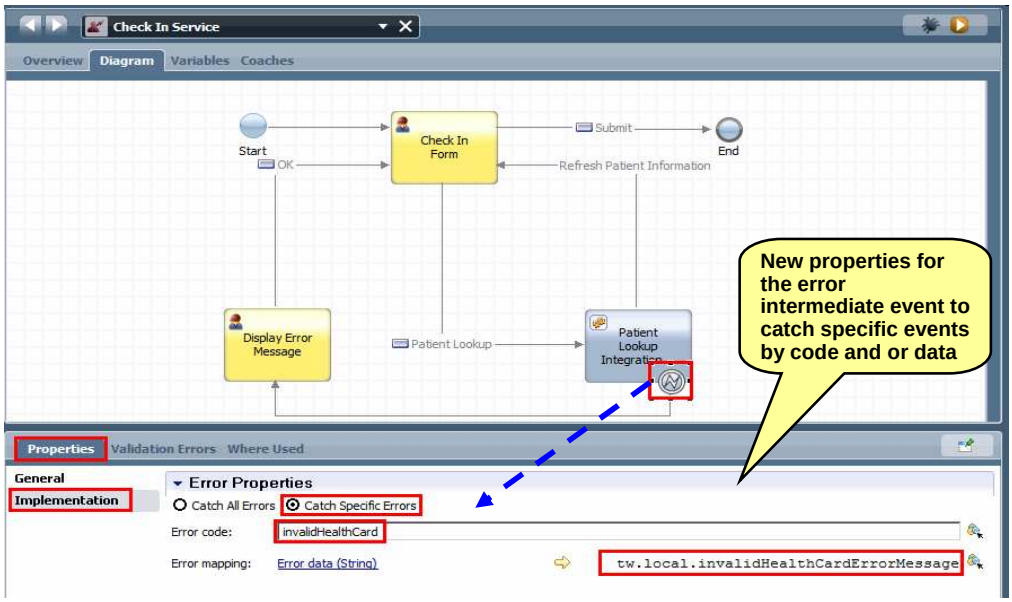
Defining Error End Event – Throw semantics



The slide shows the new properties of an error end event. The Error code is a simple string to identify a specific error.

The Error mapping property allows you to map a local variable to the Error data which is then passed to an error intermediate event that catches this error.

Defining Boundary Error Intermediate Event – Catch semantics



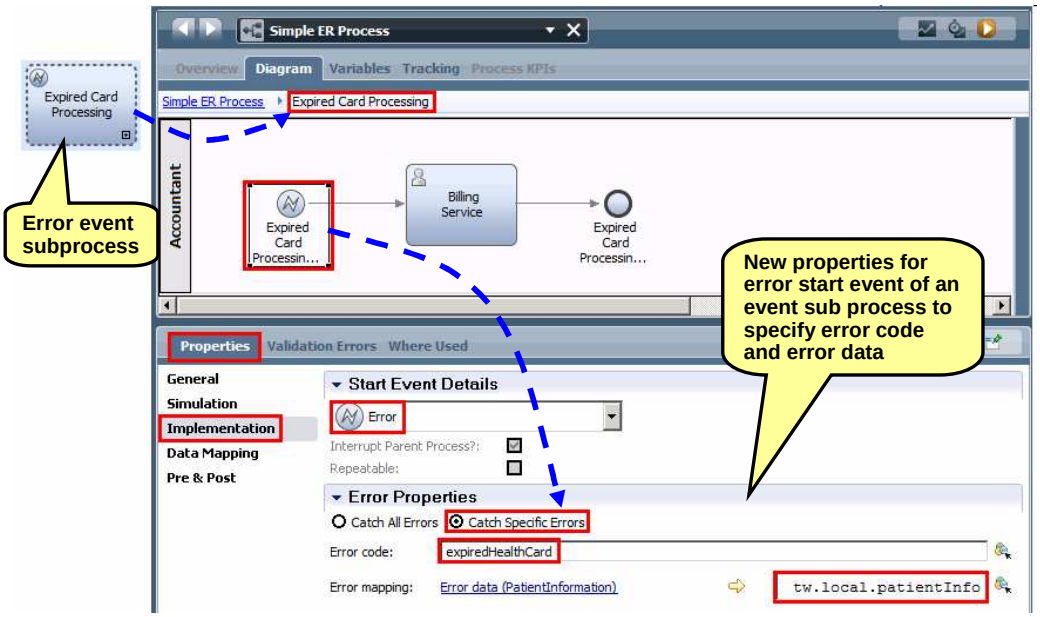
The same properties are available on an error intermediate event that is used to catch an error.

You can select the Catch All Errors option if you do not want to catch a specific error.

When catching an error, a matching combination of error code and error data type takes precedence. The next in order is a matching error code. Matching error data type is considered next.

If no error code and error data type are specified, then the event catches all errors.

Defining Error Event Subprocess – Catch semantics

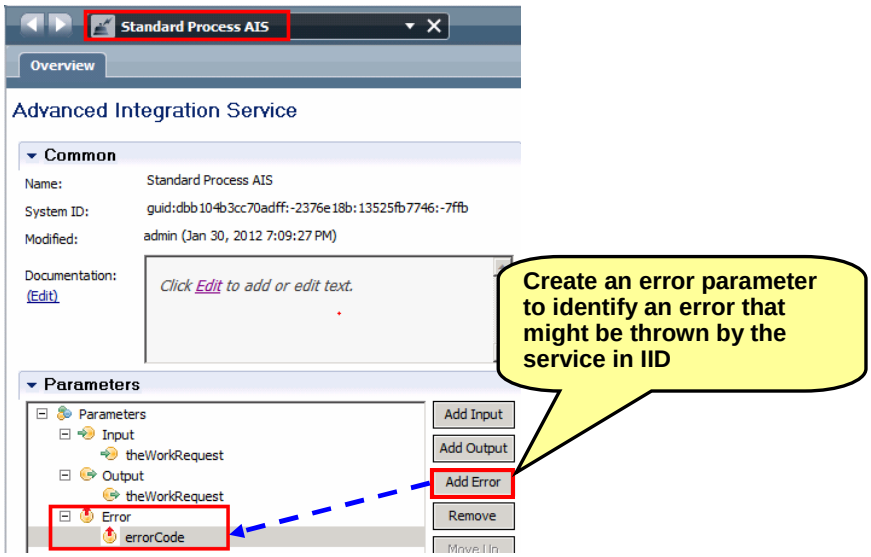


The screen capture on this slide is an example of an event sub process that starts with an error start event. You can define the error code and map the error data to a local variable for the error start event thus allowing the event sub process to catch a specific error.

The error event sub process catches an error in the same order as an error intermediate event on an activity which is described on the previous slide.

Defining errors in Advanced Integration Services

- Ability to define error parameters in the AIS editor



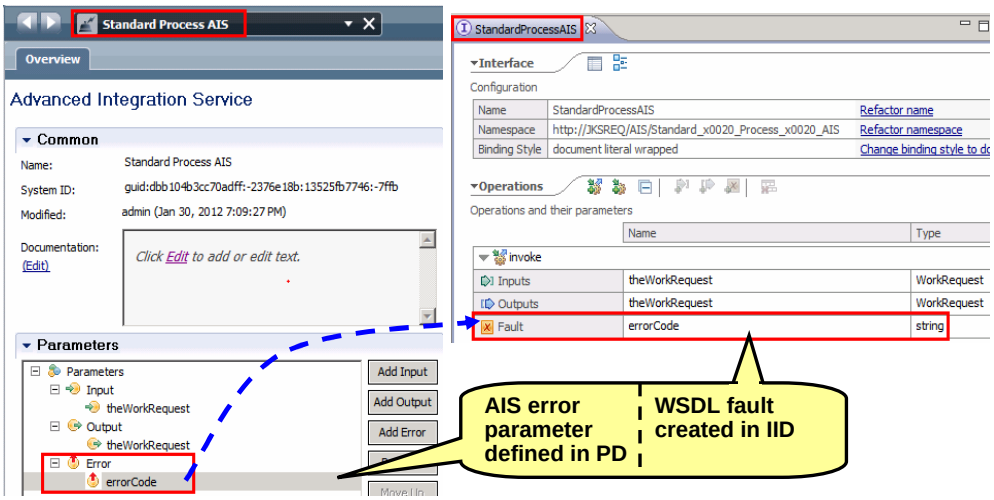
A business process definition can call a service implemented in Integration Designer using an Advanced Integration Service. In version 7.5, Process Designer does not have a very sophisticated support for handling any exceptions or faults that the service in Integration Designer might throw.

Version 8.0 introduces error parameters for Advanced Integration Services that identify errors thrown by the service. You can combine this with the ability to catch specific errors that you learned about on earlier slides. The error parameter name acts as an error code which you can specify for the event that catches the error.

The next two slides will show how these error parameters are handled in the top down and bottom up scenarios.

Defining errors in Advanced Integration Services – Top Down scenario

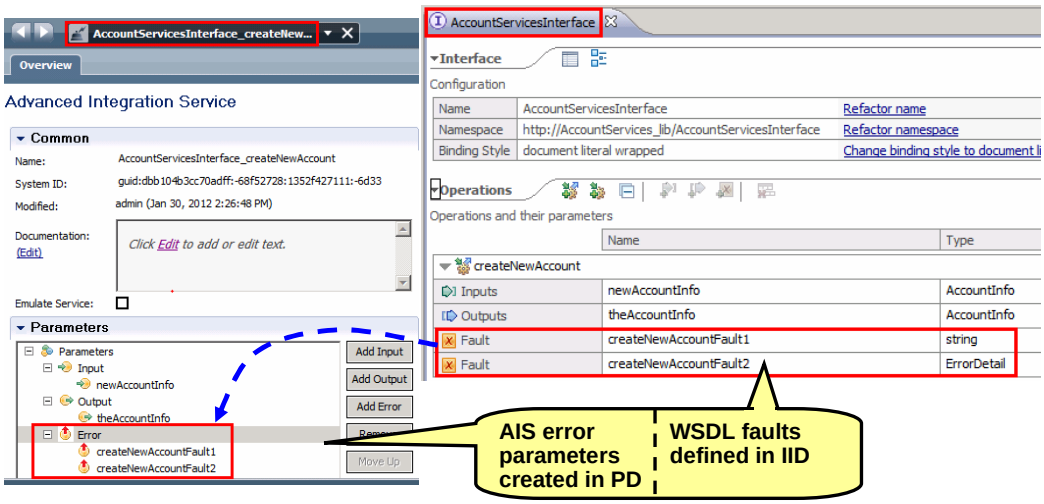
- WSDL faults generated from Advanced Integration Service error parameters



In a top down scenario, when an Advanced Integration Service created in Process Designer with error parameters is brought into Integration Designer, the WSDL that is generated has faults corresponding to the error parameters.

Defining error in Advanced Integration Services – Bottom Up scenario

- No more restrictions imposed by WSDL faults
- Advanced Integration Service error parameters generated from WSDL faults

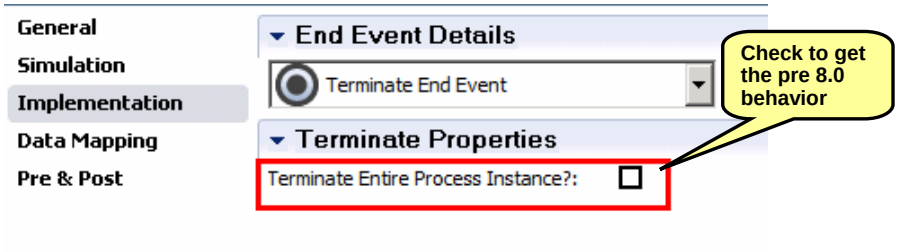


In a bottom up scenario, when an interface with faults is published to the Process Center from Integration Designer, you will see error parameters in the generated Advanced Integration Service in Process Designer. Now that wsdl faults are handled, you will no longer see validation errors in Integration Designer that the faults are not supported.

The slide shows an interface called AccountServicesInterface with two faults in the createNewAccount operation and the corresponding Advanced Integration Service in Process Designer with two error parameters.

Termination behavior

- Ends all activities at the same level and lower level
  - In the case of looping activity, the current iteration is ended
- Ability to select pre 8.0 behavior of ending the entire process instance for compatibility with earlier versions
  - Selected by default for older models
  - Not selected by default for new models



In versions before 8.0, the terminate end event ends the entire process instance irrespective of which level the event is at.

With version 8.0, the process termination semantics allow for ending only the current level activities and the lower level activities instead of ending the entire process instance.

But, if you want the behavior like before, an option is added to the properties view -> implementation tab as shown on the slide. For models imported from older versions, the option is selected by default so that these models continue to behave as they did before.

For processes created in version 8.0, the option is not selected by default enforcing the new semantics.

Summary

- Specific errors can be caught based on error code, error data or a combination of both in business process definitions and services
- Exceptions or faults thrown by a service in Integration Designer are now handled in Process Designer through error parameters of an Advanced Integration Service
- Terminate end event terminates the current level activities and the lower level activities by default.

To summarize this presentation content, IBM Business Process Manager 8.0 offers new error handling capabilities such as catching of specific runtime errors and handling errors thrown by a service in Integration Designer.
It also allows the terminate end event to end the current level activities and the lower level activities instead of ending the entire process instance by default.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_BPM80_ErrorAndTerminationHandling.ppt

This module is also available in PDF format at: ../BPM80_ErrorAndTerminationHandling.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information