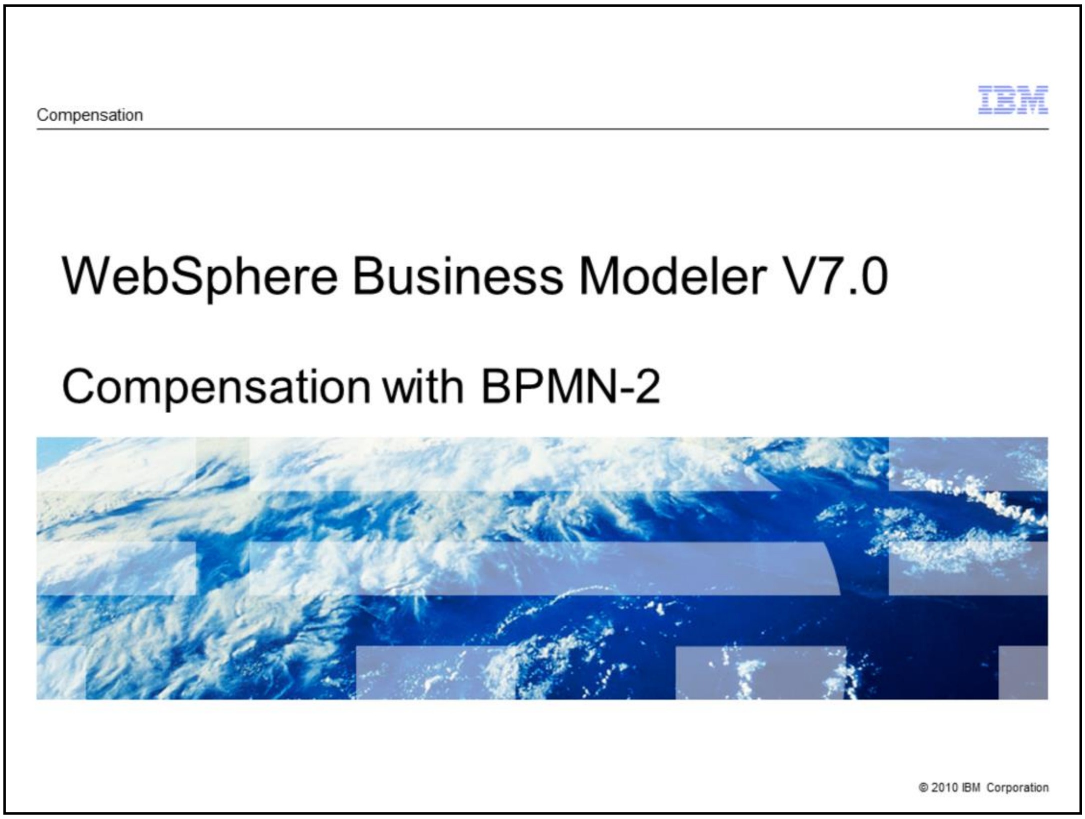
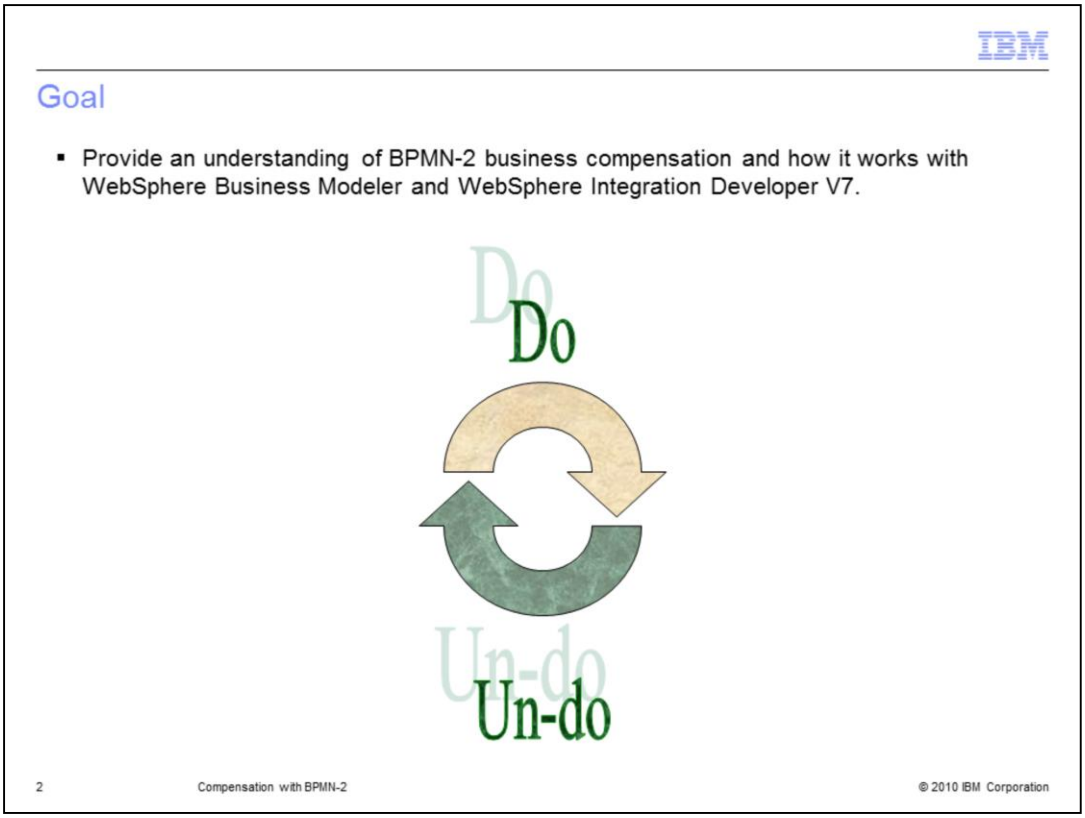


Compensation

WebSphere Business Modeler V7.0

Compensation with BPMN-2

This presentation provides an introduction to business compensation modeling in WebSphere Business Modeler V7



Goal

- Provide an understanding of BPMN-2 business compensation and how it works with WebSphere Business Modeler and WebSphere Integration Developer V7.

2 Compensation with BPMN-2 © 2010 IBM Corporation

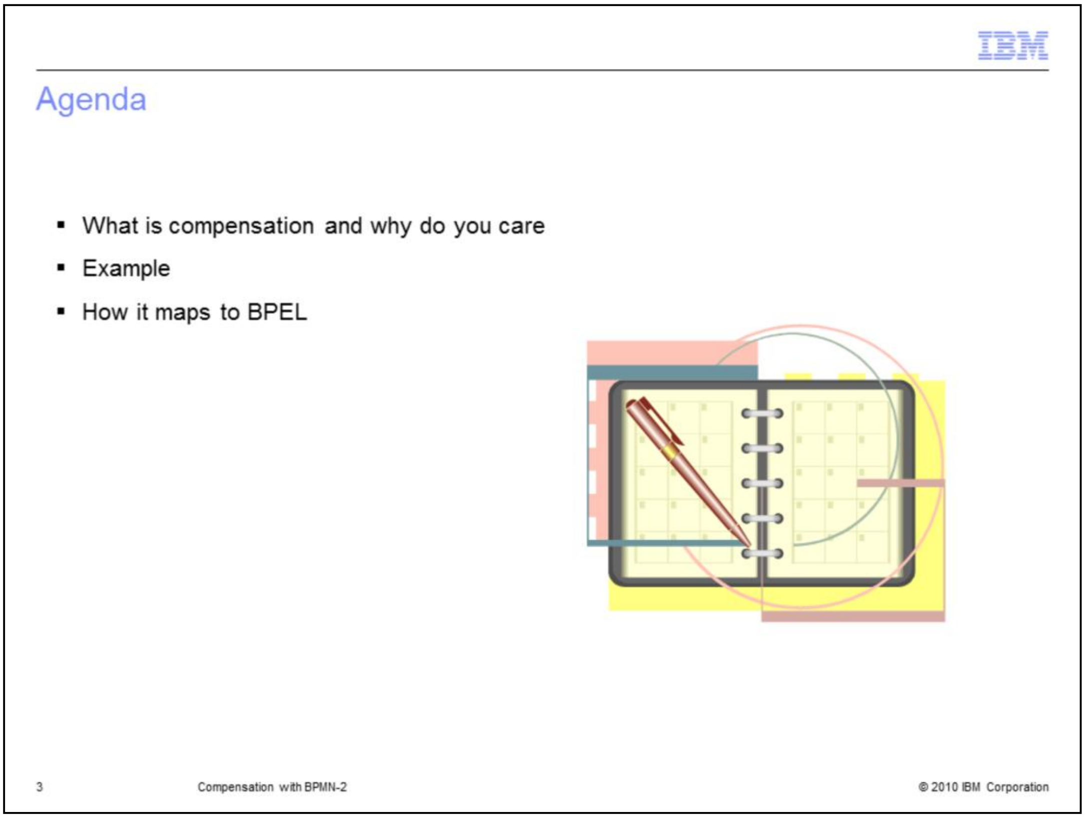
Compensation is often an overlooked or misunderstood topic.

It has been available for several versions of WebSphere Integration Developer in a couple of forms.

With version 7, it is now available for the business modeler using WebSphere Business Modeler and,

when modeling for execution, is transformed to the latest BPEL compensation pattern and constructs.

The goal of this presentation is to provide an understanding BPMN-2 business compensation and how it works with WebSphere Business Modeler and WebSphere Integration Developer V7.



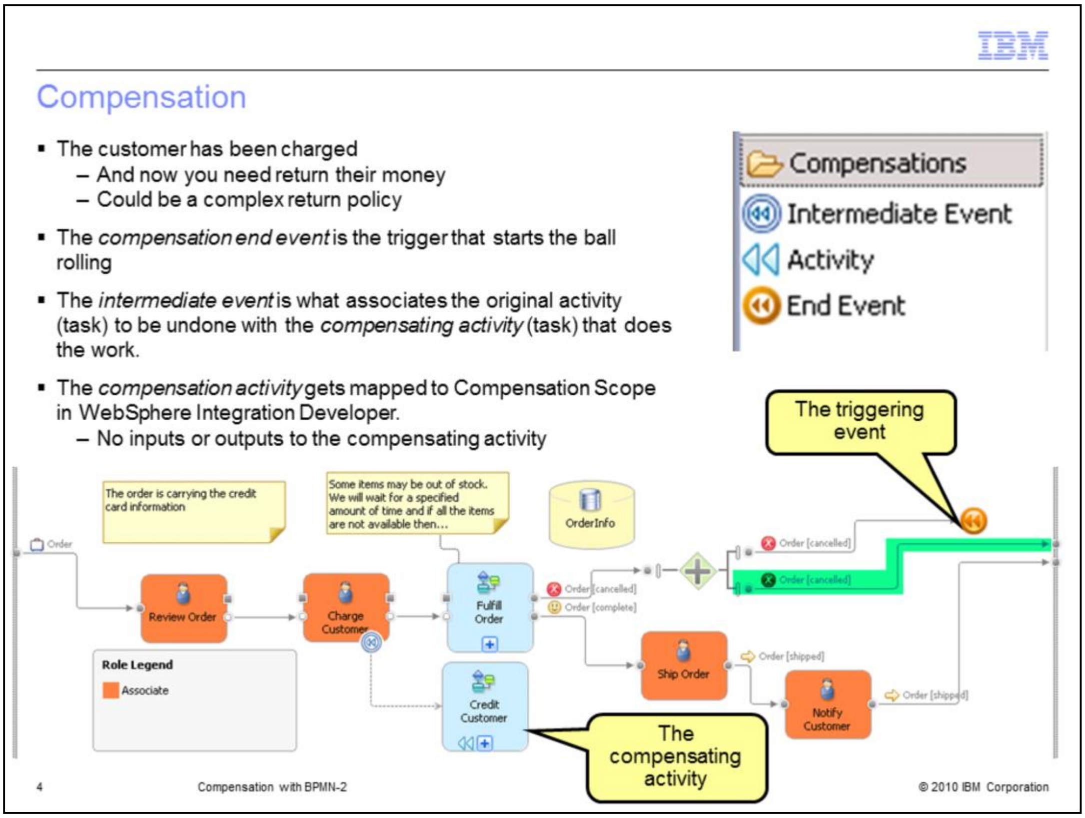
Agenda

- What is compensation and why do you care
- Example
- How it maps to BPEL

In this presentation you will learn about the concept of business compensation and learn why it is important to you.

This is followed by a detailed example that will walk you through some of the nuances specific to this modeling technique.

Then finally we’ll show how the BPMN-2 compensation constructs map to BPEL, when imported into WebSphere Integration Developer V7.



Compensation

- The customer has been charged
  - And now you need return their money
  - Could be a complex return policy
- The *compensation end event* is the trigger that starts the ball rolling
- The *intermediate event* is what associates the original activity (task) to be undone with the *compensating activity* (task) that does the work.
- The *compensation activity* gets mapped to Compensation Scope in WebSphere Integration Developer.
  - No inputs or outputs to the compensating activity

The triggering event

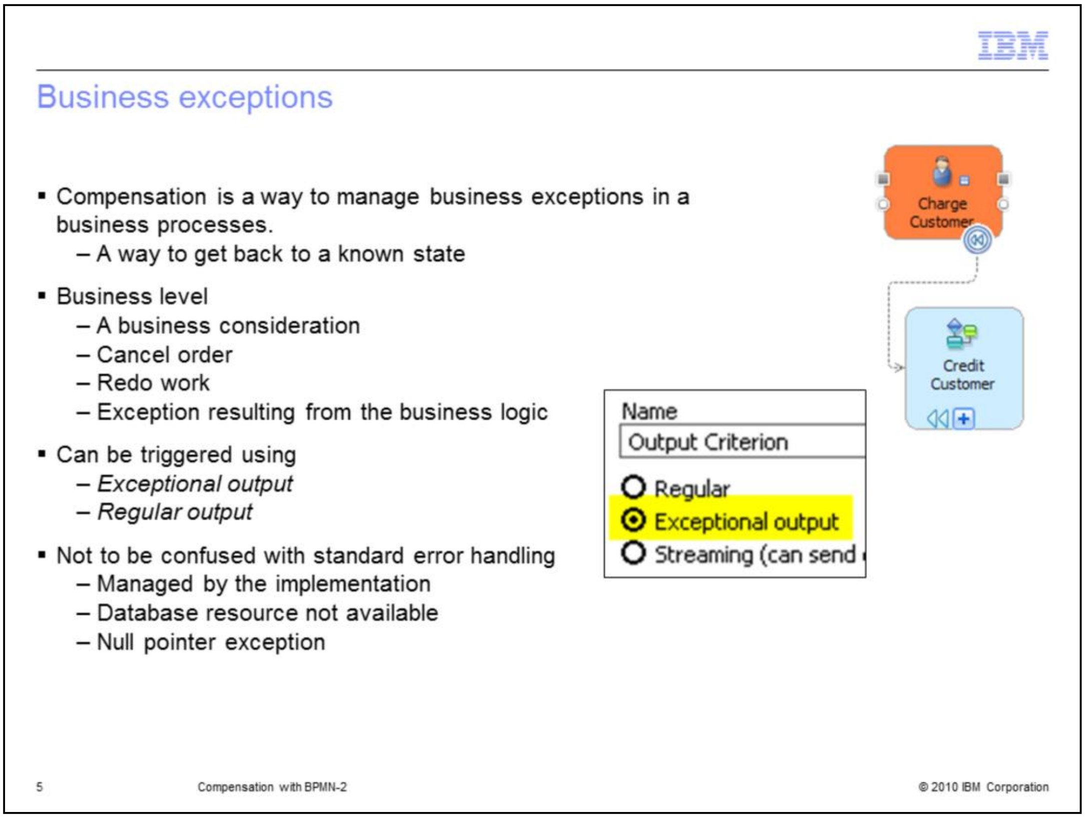
The compensating activity

4 Compensation with BPMN-2 © 2010 IBM Corporation

Compensation is a way to model business exceptions. In this case the customer was billed before the order was completely fulfilled. As part of the order fulfillment sub-process there is a rule that says, “don’t wait forever”. If all the items can’t be assembled within a specified period of time the order is cancelled and the customer is notified. This means that they will also need to have a credit applied to their credit card. It might be the return policy in not straightforward. It might be based on the special deals at the time of purchase or the length of time since the item was purchased. As you’ll see in the example, there is a lot of flexibility with what you can do in the compensating activity. It is just a special kind of sub-process.

The task being compensated is the ‘charge customer’ task. It is compensated by the ‘credit customer’ task and the compensation is triggered by an action in the ‘fulfill order’ sub-process that causes it to take the alternative path out. The alternative path is modeled as a exceptional path. The fork in green on the exceptional path is to connect the output of the business process with an order object that has the basic status information. The fork that goes to the compensation end event will trigger the ‘credit customer’ sub-process. Notice that there are no inputs or outputs available to or from the compensating activity.

Beginning with the outputs on the ‘fulfill order’ task, the feature in WebSphere Business Modeler to display the state is being used. The order has a state of ‘cancelled’, ‘complete’, and ‘shipped’. Custom icons have been associated with the states making it easier to interpret the business process flow diagram.



Business exceptions

- Compensation is a way to manage business exceptions in a business processes.
  - A way to get back to a known state
- Business level
  - A business consideration
  - Cancel order
  - Redo work
  - Exception resulting from the business logic
- Can be triggered using
  - *Exceptional output*
  - *Regular output*
- Not to be confused with standard error handling
  - Managed by the implementation
  - Database resource not available
  - Null pointer exception

Name
Output Criterion
○ Regular
● Exceptional output
○ Streaming (can send

5 Compensation with BPMN-2 © 2010 IBM Corporation

Compensation with business processes is designed to handle those situations in a business process where some activity that has already completed, needs to be undone. It is similar to the concept of a rollback that can be done with a transaction.

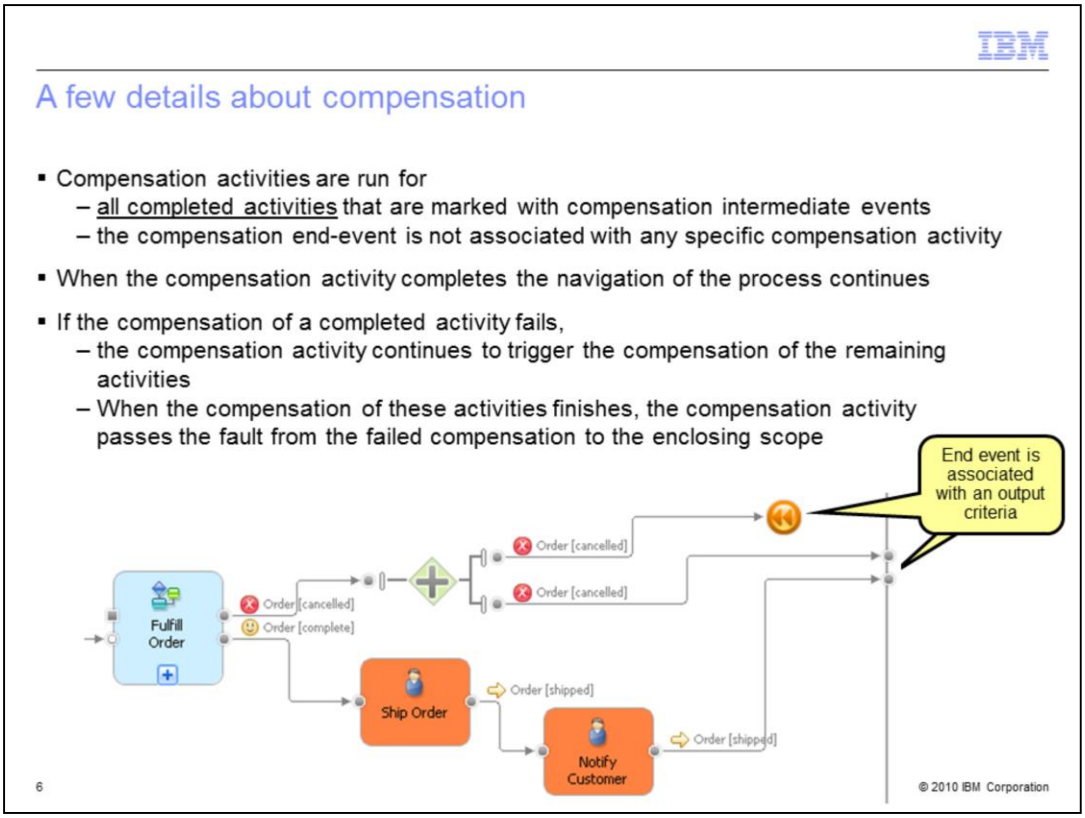
In the case of business compensation though, the unit of work, or transaction, has already been committed and therefore cannot be easily rolled back.

With compensation, a special sub-process is used to encapsulate all the tasks or activities that are required to undo the part of the business process that needs to be returned to the original, or desired state.

When designing business process flows that use compensation, it is important to keep the proper perspective. You should be thinking at the level of the business process and not the lower level error handling which is the programmer’s job.

As an example, think of the situation where a resource is not available. If the missing resource is an item of inventory which is out of stock, then it is part of the business process. If it is the database that is temporarily unavailable, then it is part of the infrastructure.

The depleted inventory can trigger a compensation activity but the case of the database that is unavailable should be left for the programmers to manage using other techniques.



A few details about compensation

- Compensation activities are run for
  - all completed activities that are marked with compensation intermediate events
  - the compensation end-event is not associated with any specific compensation activity
- When the compensation activity completes the navigation of the process continues
- If the compensation of a completed activity fails,
  - the compensation activity continues to trigger the compensation of the remaining activities
  - When the compensation of these activities finishes, the compensation activity passes the fault from the failed compensation to the enclosing scope

End event is associated with an output criteria

6

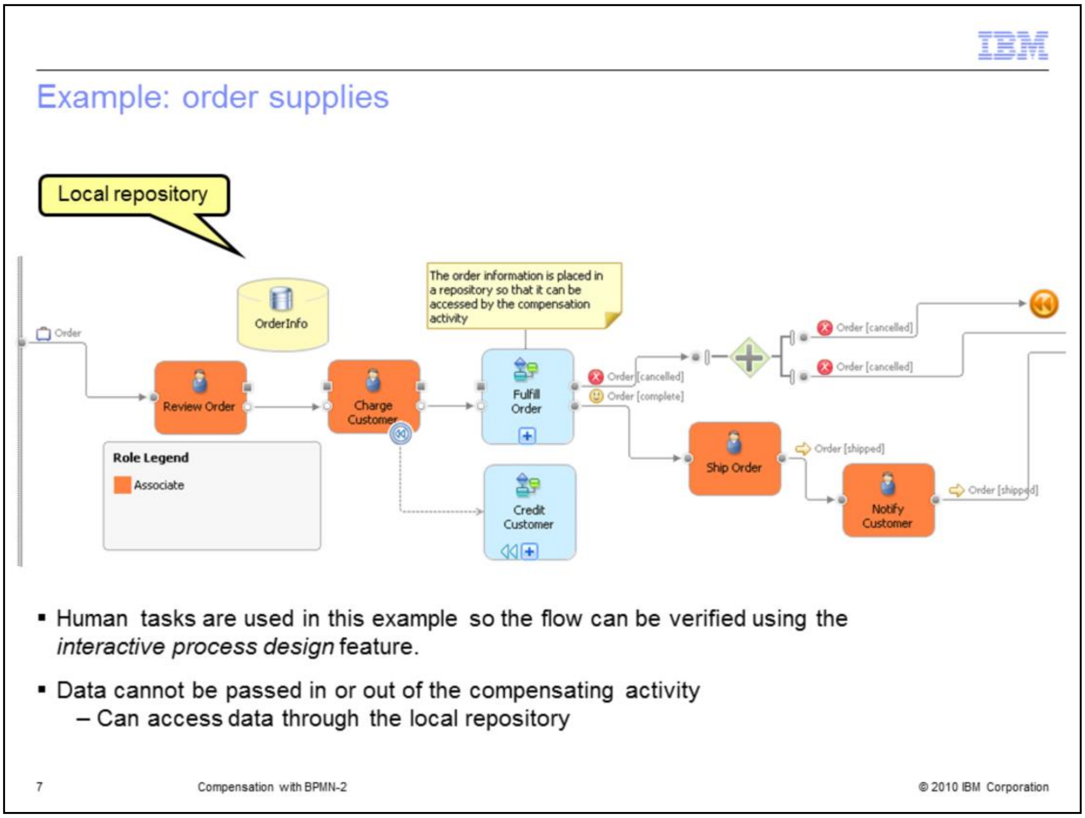
© 2010 IBM Corporation

This might be more than you want to know about compensation at this time.

When modeling compensation, it is the entire business process that is being compensated. When the triggering event happens, all of the compensation activities are run, provided the activity they are associated with have completed. You cannot specifically run a selected compensation activity.

Once the compensation activity completes, the navigation of the business process continues. In this case, the business process will complete along the exceptional output, the bottom branch of the fork. The end event must be associated with an output criteria.

The more complicated situation is when one of the compensation activities fails. All of the compensation activities that are supposed to run are given a chance to run. When they are all finished the fault from the failing compensation is passed up the stack, to the enclosing scope.



Example: order supplies

- Human tasks are used in this example so the flow can be verified using the *interactive process design* feature.
- Data cannot be passed in or out of the compensating activity
  – Can access data through the local repository

The first thing you will notice is that there are a lot of human tasks for this business process. Many of these can be automated, such as the 'notify customer' tasks and they are, in subsequent iterations.

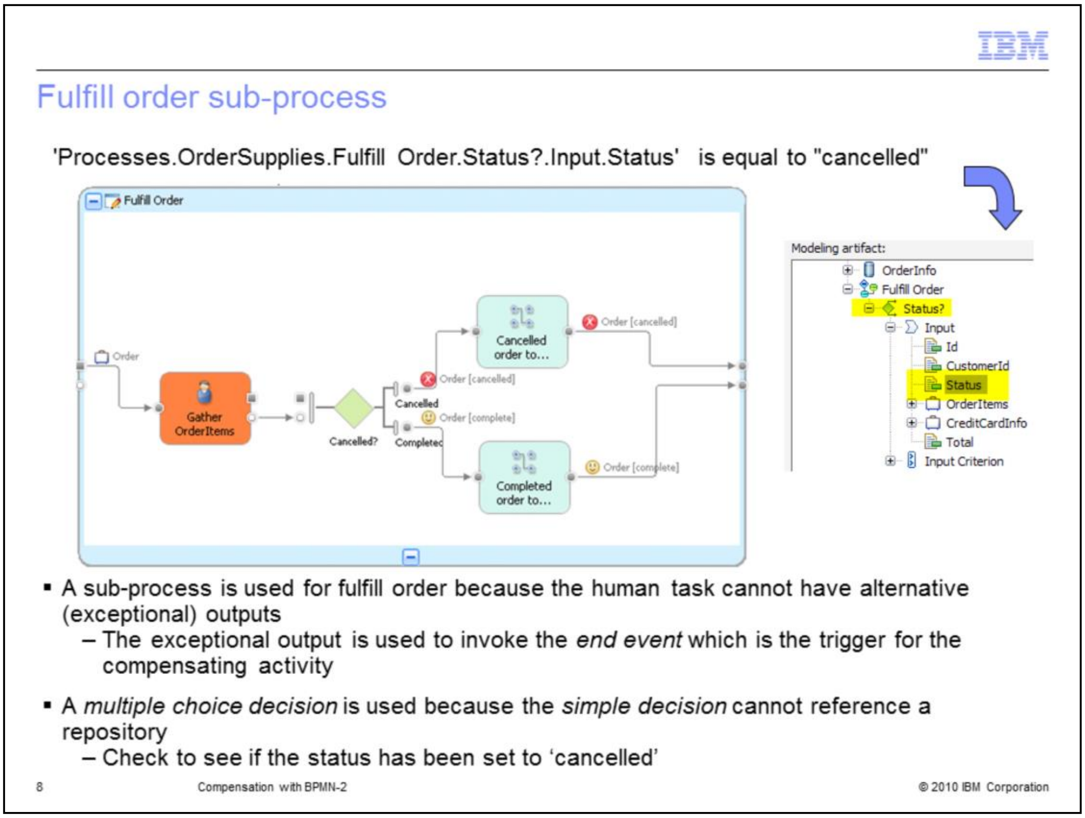
At this stage of development the interactive process design feature of WebSphere Business Modeler is being used to validate the flow. This requires that a task has an implementation that is running, or that it is a human task. Using a human task at this point is a way to simulate the automated task until the implementation becomes available.

A local repository is being used because you are not allowed to pass data into or out-of the compensating activity directly. You can however, reference local repositories.

To do the charge-back it is necessary to know the credit card information and the amounts involved. After the credit has been successfully applied, the customer must be notified of the cancellation.

The repository connections have been hidden to make the diagram easier to read. There is only one local repository used in this business process.

The repository connections are distinguished by the square terminals on the tasks.



Fulfill order sub-process

'Processes.OrderSupplies.Fulfill Order.Status?.Input.Status' is equal to "cancelled"

- A sub-process is used for fulfill order because the human task cannot have alternative (exceptional) outputs
  - The exceptional output is used to invoke the *end event* which is the trigger for the compensating activity
- A *multiple choice decision* is used because the *simple decision* cannot reference a repository
  - Check to see if the status has been set to 'cancelled'

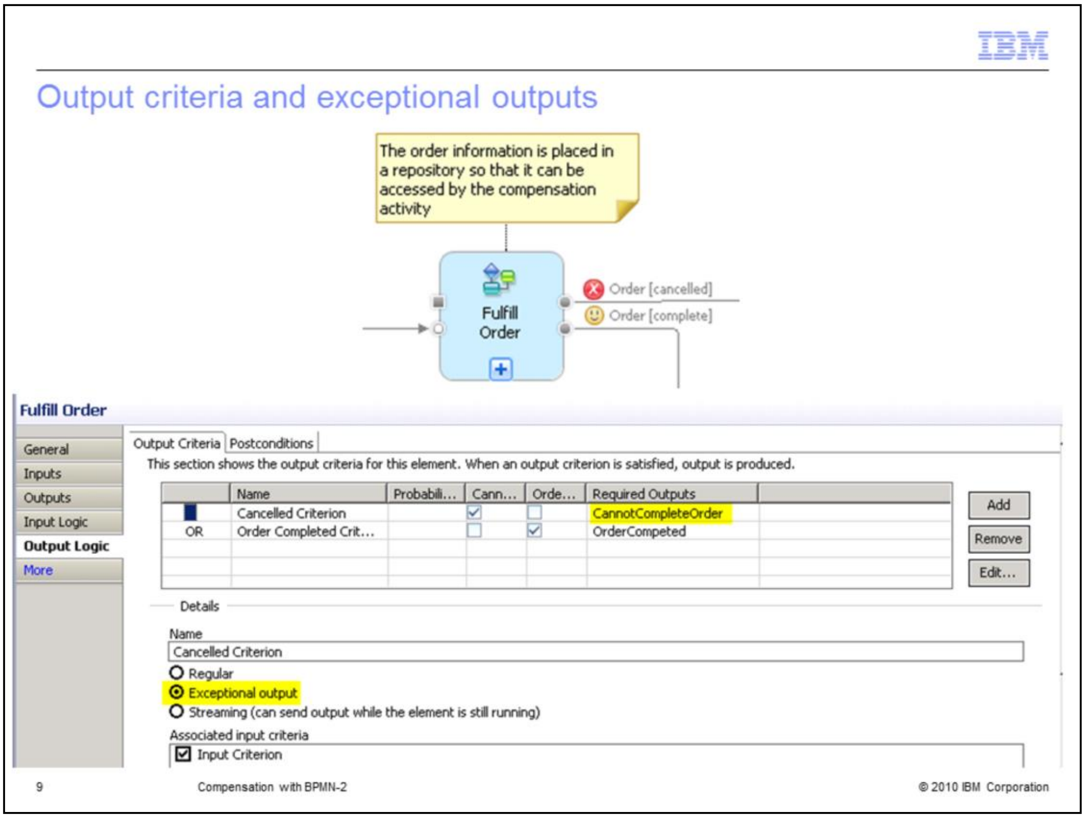
8 Compensation with BPMN-2 © 2010 IBM Corporation

Since the key data are in the OrderInfo local repository, it is important to always reference the repository consistently throughout the business process. Here the person putting the order together can cancel the order by changing the status field in the output form to 'cancelled'.

The order object is updated in the local repository and the decision is called.

The multiple choice decision interrogates the incoming order object and routes to the 'exceptional' output if the status is equal to 'cancelled'. Notice that the expression uses the order object as input to the decision and not the local repository directly. The input is still coming from the repository.

The maps are required to associate the instance data of the OrderInfo repository to the outputs of the sub-process.



Output criteria and exceptional outputs

The order information is placed in a repository so that it can be accessed by the compensation activity

Fulfill Order

This section shows the output criteria for this element. When an output criterion is satisfied, output is produced.

	Name	Probabili...	Cann...	Orde...	Required Outputs
■	Cancelled Criterion		☑	□	CannotCompleteOrder
OR	Order Completed Crit...		□	☑	OrderCompeted

Details

Name: Cancelled Criterion

○ Regular
● Exceptional output
○ Streaming (can send output while the element is still running)

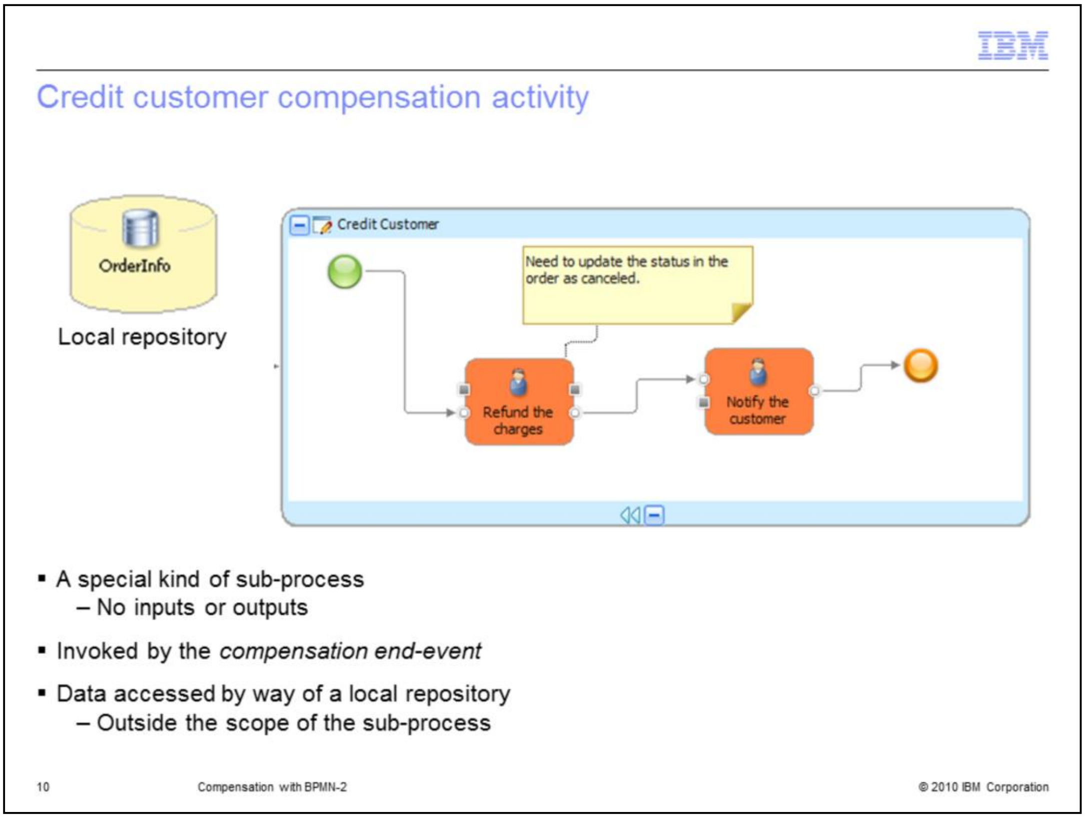
Associated input criteria

☑ Input Criterion

Here you can see how the exceptional output is specified for the output criteria of the ‘fulfill order’’ sub-process.

There are two, mutually exclusive outputs indicated by the ‘or’ condition. The CannotCompleteOrder output has been set to be the exceptional output.

As you can imagine, there are many ways to model the invocation of the end-event. You can also use the mid-process receive to wait for an un-solicited message from the customer to cancel the order.



Credit customer compensation activity

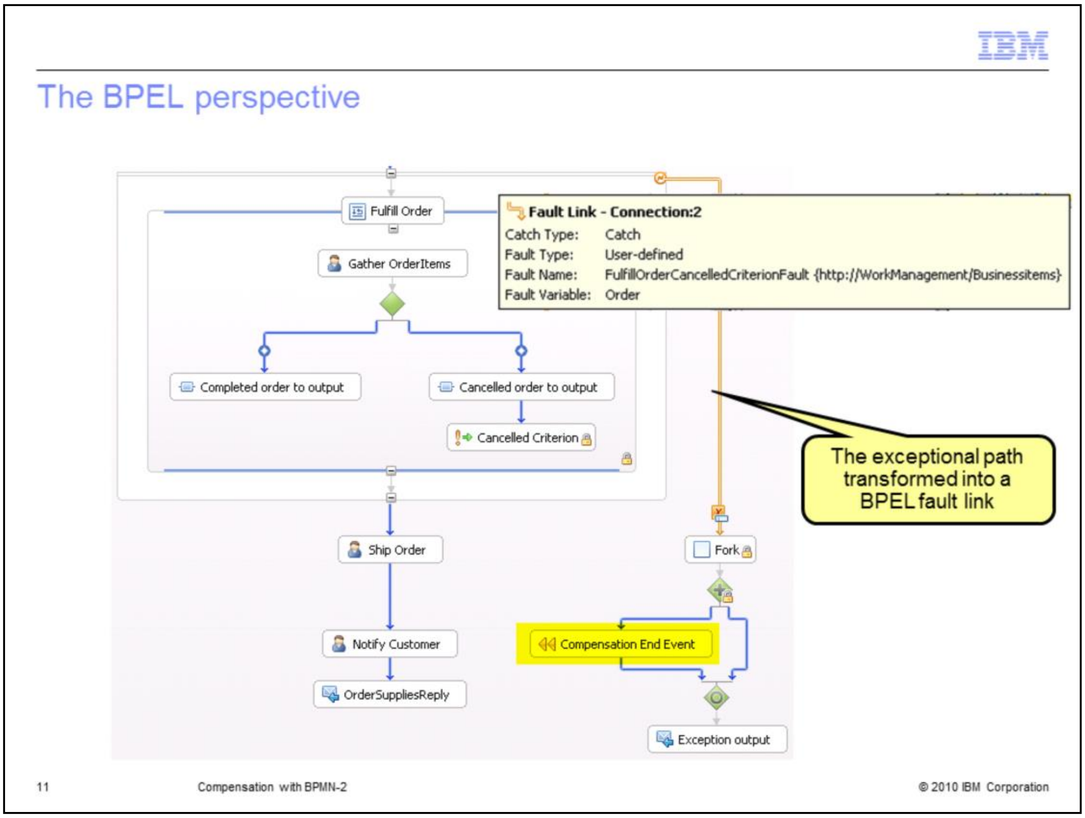
- A special kind of sub-process
  - No inputs or outputs
- Invoked by the *compensation end-event*
- Data accessed by way of a local repository
  - Outside the scope of the sub-process

10 Compensation with BPMN-2 © 2010 IBM Corporation

The compensating activity is a special kind of sub-process, one without inputs or outputs.

It is invoked when the compensating end-event is reached in a flow.

There is nothing fancy happening in this compensation activity. The necessary input information is referenced from the local repository, which is updated after the credit card account has been credited with the refund. The customer is then notified of the refund.



The BPEL perspective

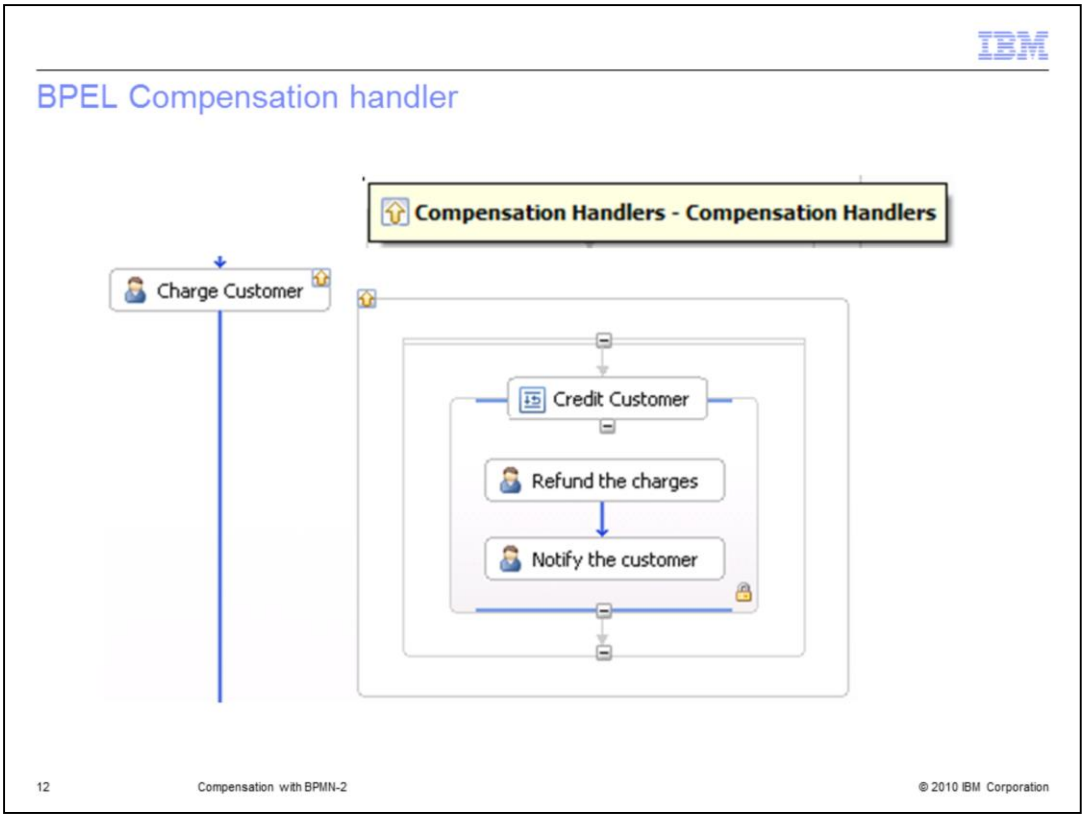
Fault Link - Connection:2

Catch Type: Catch
Fault Type: User-defined
Fault Name: FulfillOrderCancelledCriterionFault {http://WorkManagement/Businessitems}
Fault Variable: Order

The exceptional path transformed into a BPEL fault link

11 Compensation with BPMN-2 © 2010 IBM Corporation

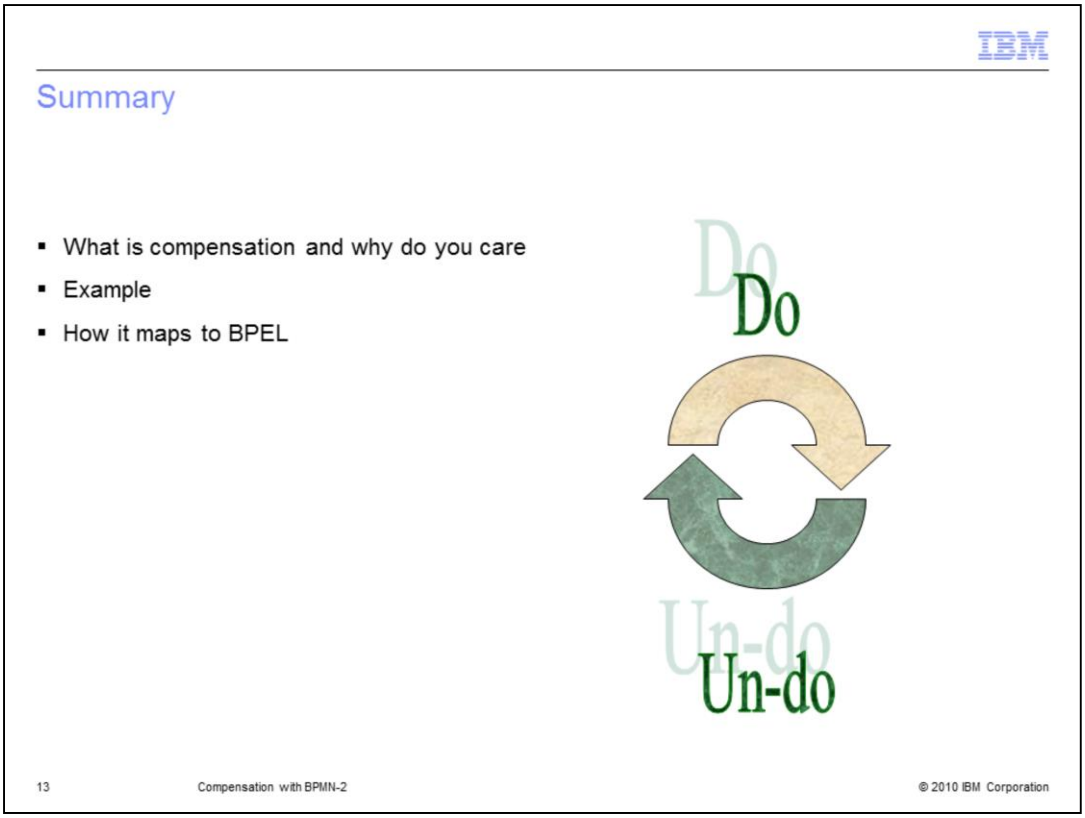
Shown here is what the model looks like when imported into WebSphere Integration Developer. The exceptional output from modeler is transformed into a BPEL fault link. The compensation end-event is mapped directly to it’s BPEL counterpart.



BPEL Compensation handler

12 Compensation with BPMN-2 © 2010 IBM Corporation

Here you can see how the compensation activity is transformed directly into a BPEL compensation handler.



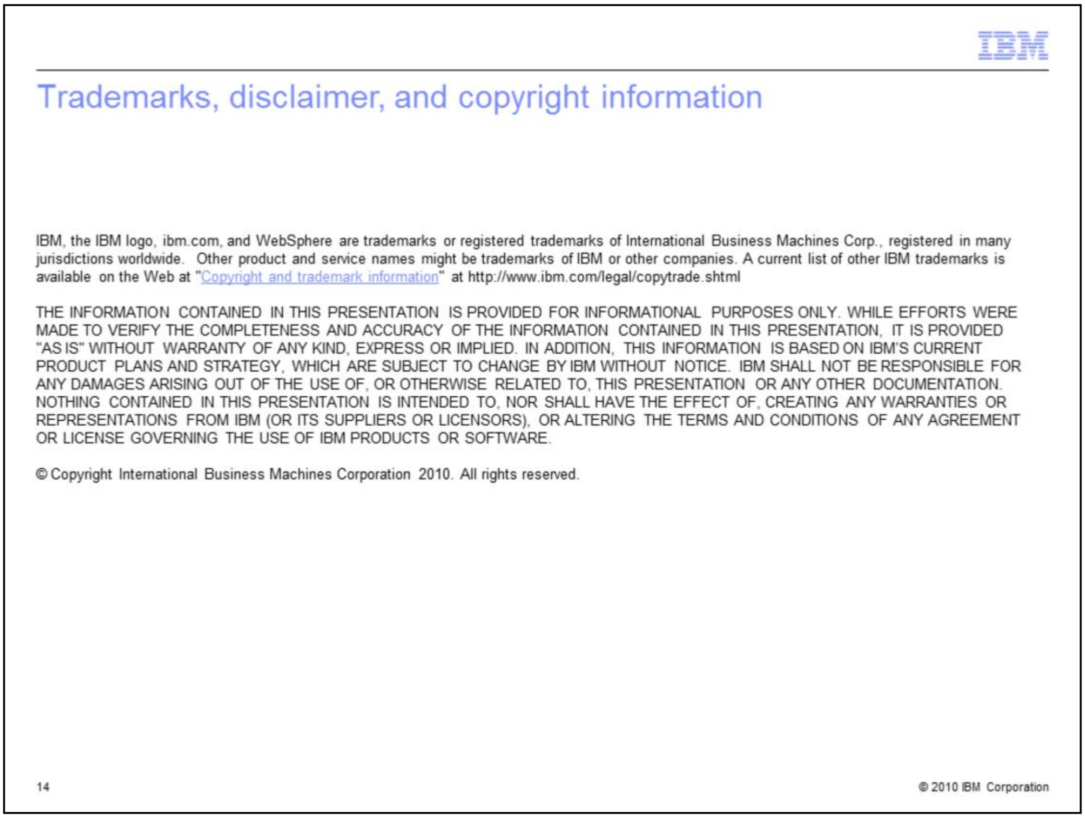
Summary

- What is compensation and why do you care
- Example
- How it maps to BPEL

13 Compensation with BPMN-2 © 2010 IBM Corporation

In this presentation you learned about how to use compensation to manage business exceptions and have distinguished the difference between business exceptions and infrastructure or programming exceptions.

You were given a walk through of the supply ordering example which ended by showing how the BPMN-2 compensation modeled in WebSphere Business Modeler V7 maps to the BPEL compensation in WebSphere Integration Developer V7.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.