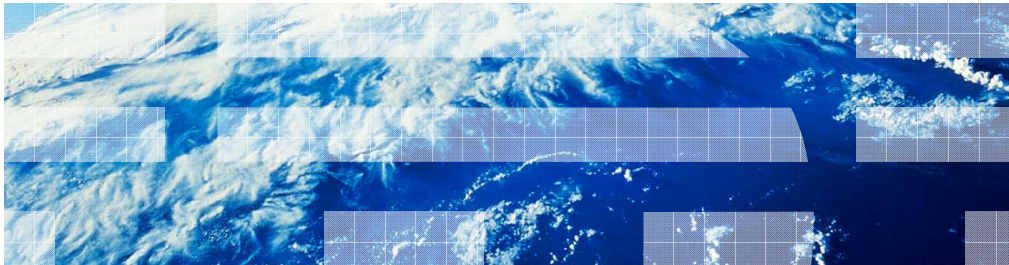


# WebSphere business process management

WebSphere Integration Developer  
WebSphere Enterprise Service Bus  
WebSphere Process Server

Basic concepts and configuration of promoted properties



This is the first of two presentations that address promoted properties. The basic concepts of promoted properties and the configuration details are covered here. The next presentation explains the different usage patterns for promoted properties along with the runtime details.

## Goals and agenda

- Overall goal is to introduce promoted properties
  - Basic concepts
  - Configuration
  - Prepare you for the more detailed presentation on promoted properties
- Agenda:
  - Basic concepts and terminology
    - What are properties and promoted properties
    - Introduce three major variations for how promoted properties can be used in a flow
    - Provide example scenarios
    - Introduce the terminology fundamental to understanding promoted properties
  - Configuration of promoted properties
    - Defining promoted properties
    - Reviewing promoted properties defined at various scopes

The overall goal of this presentation is to introduce you to promoted properties, presenting the basic concepts and reviewing how promoted properties are configured. With the information you learn from this presentation, you are ready to understand the more detailed presentation on usage of promoted properties.

The presentation of concepts starts by defining what a property is and what a promoted property is. There are three essentially different usage patterns for promoted properties. These are introduced here as background along with mention of some example scenarios, but it is the subsequent presentation that provides the details of those usage patterns. Next, there is some very specific terminology introduced that enables clarity when discussing promoted properties.

In the section on configuration, you are shown how promoted properties are configured using WebSphere Integration Developer. Also introduced are the panels that can be used to review the configuration of promoted properties across different scopes, such as an individual flow or an entire component.

## ***Basic concepts and terminology***

This section of the presentation introduces the basic concepts of promoted properties. In addition, the key terms that are used when discussing promoted properties are introduced and explained.

## Introduction to properties and promoted properties

- Properties:
  - A property provides configuration data associated with a primitive (or callout) in a flow
  - Each primitive type provides some pre-defined functional capability (behavior)
  - Primitive types define required and optional properties used to customize behavior
  - Each instance of a primitive in a flow has specific values assigned to the properties
  - The value specified for a property is statically defined in the flow implementation
- Promoted properties
  - A promoted property can have a value assigned that overrides the statically defined value specified in the flow implementation
  - Each primitive type defines which of its properties are capable of being promoted
  - Each instance of a primitive in a flow determines which properties are actually promoted

In order to understand a promoted property, it is first necessary to understand what a property is and how it is used in the definition of a mediation flow. A property is used to provide the specification of some configuration data that is associated with a primitive in a mediation flow. Callout nodes also have properties that can be promoted. A primitive is the basic building block for defining the logic of a flow. Each primitive type provides some predefined functional capability, or behavior, that is further refined through configuration data. Each primitive type defines a set of properties used to specify its configuration data. Some properties might be required whereas others are optional. Every instance of a primitive in a flow has values specified for its properties which are statically defined as part of the flow implementation.

A promoted property allows there to be a value specified that overrides the statically defined value of the property specified in a flow implementation. Just as each primitive type defines what its properties are, the primitive type also defines which of its properties can be promoted. Each instance of a primitive used in a flow contains the static property values and specifies which of the properties are promoted, thus allowing the static value to potentially be overridden.

## Usage patterns for promoted properties

- Runtime administration
  - Enables administrator to modify runtime behavior of a primitive in a flow
  - Administrator supplies the values for promoted properties
  - Applies to all transactions executing in the flow
- Mediation policy
  - Enables use of policy to modify runtime behavior of a primitive in a flow
  - Values for promoted properties obtained using a registry lookup
  - Applies to a single transaction executing the flow implementation
- Subflows
  - Enables the invoking flow to modify configured behavior of a primitive in a subflow
  - Invoking flow supplies values for promoted properties
  - Statically defines the value for the promoted property

There are three essentially different ways that promoted properties are used within mediation flows, specifically runtime administration, use of mediation policies and invocation of subflows. Each of these makes use of the capability to override a statically defined value of a promoted property, but the scope over which the overriding value applies is different in each case.

Runtime administration enables an administrator to modify the runtime behavior of a primitive in a flow contained in an installed mediation application. The administrator provides the overriding value during application installation or as an administrative update to an installed application. The overriding value applies to all transactions executing in that flow.

Mediation policy enables the use of policy specifications to modify the runtime behavior of a primitive in a flow. The overriding value for the promoted property is obtained using a registry lookup. The lookup accesses policy information based on data values in a specific execution instance of the flow. The overriding values only apply to that one execution instance of the flow.

Subflows enable flow logic to be reused from many different flows. Promoted properties enable the invoking flow to configure the behavior of the subflow. The invoking flow provides the overriding value for the promoted properties. In essence, the subflow can be thought of as a primitive type, and the properties of the subflow are the promoted properties of the primitives that make up the subflow logic. Therefore, the statically defined value associated with a property of a primitive in a subflow is statically overridden for its use within a parent flow.

## Scenarios of promoted property usage

- Administrative scenarios
  - Toggle message logging or tracing on and off
  - Toggle validation for mediation primitives
  - Change values used to control flow paths
- Policy scenario
  - Control map selection to enable country specific address formatting
- Subflow scenario
  - Specify reference associated with a service invoke primitive in the subflow

To further understand the three ways promoted properties are used in mediation flows, a few example scenarios are mentioned here. The first set of examples are for the runtime administration of promoted properties. One is the case of a message logger primitive that is only logging a small portion of the message, such as part of the body. When trying to debug a runtime issue, it might be useful to see the entire message written to the message log, allowing headers and context information to be examined. This can be done by administratively modifying the root property of the message logger.

Another example also involves problem resolution. In this case, there is a problem with a malformed message and an attempt is being made to determine where the message gets corrupted. Several of the mediation primitives have a validate input option, which is normally not used in a production system because of performance implications. However, in order to debug the malformed message problem, promoted properties can be used to turn on the validate input options. This allows you to catch the message corruption issue closer to where it originally occurred.

The third example applicable to runtime administration is related more to an application change. Assume you have a mediation that uses a message filter to take different flow paths based on some criteria, such as a rating value. Promoted properties can be used to adjust the rating value threshold used to select a particular path.

An example for mediation policy relates to formatting addresses for different countries. Assume that inbound messages have a generic address format and that the outbound messages need to carry the address in its country specific format. A different XSL map is needed for each different address format. The mediation flow has an XSLT primitive which promotes the property specifying the map name. A policy is then used that provides the correct map name based on the country specified in the inbound message.

The subflow example assumes that there is a subflow that contains a service invoke primitive. The primitive is configured with the interface of the service to be invoked, but the reference used to invoke the service is only known by the parent flow. The promotion of the reference property of the service invoke allows the parent flow to set the name of the reference.

## Not all properties are promoted

- Not all properties can be promoted
  - Only certain properties are allowed to be promoted
    - Restricted to properties whose value can change without:
      - requiring a redeployment
      - affecting some related artifacts created from the development time definition
    - This is determined by the mediation primitive or callout implementation
  - As the integration developer, you
    - Configure the mediation primitives and callouts for a specific mediation module
    - Identify which of the allowable properties are actually promoted

This slide examines which properties are available to be promoted so that their values can be overridden.

The first thing to consider is that only certain mediation properties are eligible to be overridden at runtime. For any given mediation primitive type, only selected properties can be made available to be overridden. Those that cannot be overridden at runtime generally have some characteristic that prevents them from being used in this way. One reason is that changing the value for a particular property can sometimes force an application redeployment, in order for the change to take affect in the running application. Another reason is that some properties define relationships to other development time artifacts, which requires the change to be made using WebSphere Integration Developer. Based on factors such as these, each mediation primitive type defines which of its properties are potentially available for promotion and which are not.

Of those properties which are eligible to be promoted, you, as the integration developer, choose which will actually be made available for a particular mediation flow and mediation primitive instance. It is only those properties which you designate that are made available as promoted properties.

The next slide introduces some terminology which makes these concepts easier to understand.

## Concepts and related terminology

- Promotable
  - Identifies a property which can be available for administrative configuration
  - Applies to all instances of that mediation primitive type
- Promoted
  - Identifies a property which is available for administrative configuration
  - Determined by you when configuring the mediation primitive
  - Applies only to that one configured instance of the mediation primitive
- Alias
  - A name you give to the promoted property
  - Provides a meaningful way for a solution administrator to identify the property
  - An alias is associated with a value for the property
- Shared alias
  - Same alias name can be given to multiple properties in the same group
  - A common name causes the alias to be shared
  - Properties with a shared alias share a common value

The previous slide introduced some concepts about promoted properties and how they work. The terminology introduced here provides specific terms that make those concepts easier to understand and to discuss.

The first term is **promotable**. This term is applied to a specific property associated with a specific mediation primitive type, if the property has the ability to be made available for promotion. Whether a property is promotable is defined by the implementation of the mediation primitive type and applies to all instances of that type of primitive.

The second term is **promoted**. This term is applied to a property which is not only promotable, but has also been explicitly selected to allow its statically defined value to be overridden. This selection is made by you, the integration developer, when developing a mediation flow. Whether a property is promoted applies to only the individual instance of the mediation primitive to which that property belongs.

The next term is **alias**. When a property is promoted, it is associated with an alias, which you can think of as a name value pair. You provide an alias name that has some meaning to the administrator or integration developer who is supplying the overriding value. The alias name is associated with a value that is applied to the property. So the administrator or integration developer, in essence, does not provide a modified value to override the property. Rather, the overriding value provided is associated with the alias and the modified alias value is applied to the property.

The final term is **shared alias**. Within a mediation module, promoted properties are organized into groups. Within a group, promoted properties with the same alias name share the same alias value. This allows multiple related properties within a mediation module to be administered together. An example might help to illustrate this. On a previous slide there was a scenario described where multiple validate input properties were to be toggled between off and on. By having a shared alias for these multiple validate input properties, the administrator can modify them all by only having to change a single alias value.

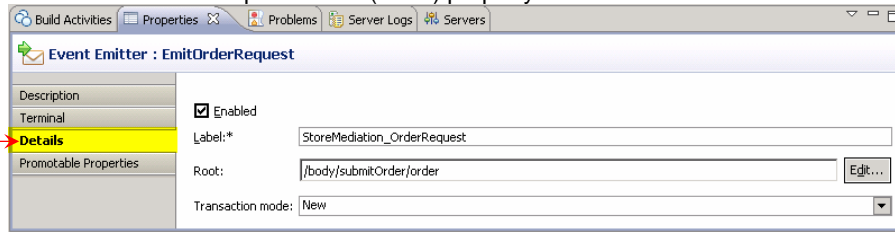


## ***Configuring promotable properties***

Now that the basic concepts of promoted properties have been covered, this section looks at how to use WebSphere Integration Developer to configure promoted properties during development time.

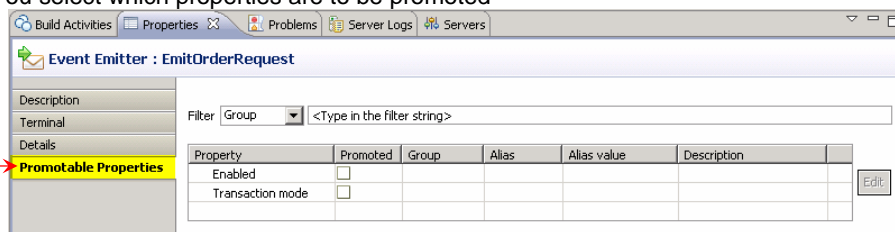
## Promotable properties panel

Details panel - Used to set development time (static) property values



Promotable properties panel

- Table used to list the promotable properties
- You select which properties are to be promoted



10

Basic concepts and configuration of promoted properties

© 2010 IBM Corporation

In the properties view for a mediation primitive, two of the panels provided are the details panel and the promotable properties panel.

In the details panel are the various properties that are used to configure the mediation primitive, as shown in the upper screen capture. In the example shown, the panels are for an event emitter primitive which has four properties, enabled, label, root and transaction mode.

Looking at the promotable properties panel in the lower portion of the slide, you can see that only two of the four properties are promotable, the enabled and transaction mode properties. The label and root properties are not promotable because these affect the contents of the emitted event. The logic for not making these promotable is that there might be external applications which have a dependency on the event format. Therefore, these are not good candidates for administrative control.

The promotable properties panel is composed of a table with a column for the promotable properties and a column to indicate if the property is promoted. When the property is promoted, the group, alias and alias value columns come into play. The description column is only provided for documentation purposes.

In these screen captures, neither of the promotable properties has been promoted.

## Promoting a property

### Promoting a property

- Enabled property
  - Select as promoted on Promotable Properties panel
  - Provide group, alias name and value

Property	Promoted	Group	Alias	Alias value	Description
Enabled	<input checked="" type="checkbox"/>	StoreMediation	EnableOrderEvent	true	
Transaction mode	<input type="checkbox"/>				

- Icon marks property as promoted on the details panel

- Clicking Promoted icon gets you to Promotable Properties panel
- Alias value is displayed but grayed out - cannot be changed here

Enabled  
 Label:\*   
 Root:    
 Transaction mode:

This slide illustrates the promoting of a property. The Promotable Properties panel is shown on the top. As you can see, the Enabled property has been marked as promoted and values for Group and Alias names and Alias value has been filled in.

Looking at the Details panel on the bottom, you can see that the Enabled property looks somewhat different than it did in the previous slide. First of all, an icon has been placed next to the property that indicates the property has been promoted. Also, the value for the property is shown but is no longer editable from this panel. If you click the icon, it switches you to the Promotable Properties panel which allows you to update the default alias value that is applied to the property.

## Table based properties

- Rows of a table can be promoted
  - Rows selected individually to be promoted
  - Only one column of the table is identified as promotable
    - For a promoted row, only the cell in the promotable column is administered
  - When promoted, group and alias names and a value are assigned
  - Table row must be created before it shows up as available to promote

Filters:

Pattern	Terminal name
<input checked="" type="checkbox"/> /body/getCustomerID/input1/age>64	senior
<input type="checkbox"/> /body/getCustomerID/input1/age>19	adult
<input type="checkbox"/> /body/getCustomerID/input1/age>12	teenager
<input checked="" type="checkbox"/> /body/getCustomerID/input1/age<1	baby

Property	Promoted	Group	Alias	Alias value
Enabled	<input type="checkbox"/>			
Distribution mode	<input type="checkbox"/>			
senior [Pattern]	<input checked="" type="checkbox"/>	CustomerUpdate	Senior	/body/getCustomerID/input1/age>64
adult [Pattern]	<input type="checkbox"/>			
teenager [Pattern]	<input type="checkbox"/>			
baby [Pattern]	<input checked="" type="checkbox"/>	CustomerUpdate	Baby	/body/getCustomerID/input1/age<1

12

Basic concepts and configuration of promoted properties

© 2010 IBM Corporation

One of the more interesting uses of promoted properties is the ability to promote a row in a table. It is actually only a single column in the table which is promotable. Individual rows in the table can be identified as promoted. For a row that is promoted, the alias value applies to the single table cell identified by the promotable column and the promoted row. When configuring a mediation primitive containing a table, the table rows must first be created in the Details panel. The individual rows then show up in the Promotable Properties panel so that they can be selectively promoted.

This slide illustrates this capability using the Filters property of a message filter primitive. The Filters property from the Details panel is shown on the left side. Basically, it is controlling the flow through the mediation based on the age of a customer identified in the message. Looking at the Pattern column, you can see that customers greater than 64 are considered senior. Customers greater than 19 through 64 are considered adult. Customers greater than 12 through 19 are considered teenager and customers less than one are considered baby. This leaves the default flow to be for customers who are one through 12, which are considered children. Looking at the filters table you can see by the icons that the greater than 64 and less than one patterns are shown as being promoted. Now look at the Promotable Properties panel on the lower right side. You can see the Terminal name is used to identify the row but, as indicated, it is actually the Pattern column that is promotable. For those rows that are promoted, the pattern becomes the default alias value. In this particular example, the solution administrator is given the capability to modify the age dividing adult from senior and the age dividing baby from child.

## All promotable properties in a flow

Primitive	Property	Promoted	Group	Alias	Alias value
getQuote : DelayedSvcPartner	Use dynamic endpoint if ...	<input type="checkbox"/>			
getQuote : DelayedSvcPartner	Async timeout (seconds)	<input type="checkbox"/>			
getQuote : DelayedSvcPartner	Invocation Style	<input type="checkbox"/>			
getQuote : DelayedSvcPartner	Retry on	<input type="checkbox"/>			
getQuote : DelayedSvcPartner	Retry count	<input checked="" type="checkbox"/>	SQMed	RetryCount	2
getQuote : DelayedSvcPartner	Retry delay (seconds)	<input type="checkbox"/>			
getQuote : DelayedSvcPartner	Try alternate endpoints	<input type="checkbox"/>			
LookupAcct	Table	<input type="checkbox"/>			
LookupAcct	Search column	<input type="checkbox"/>			
LookupAcct	Search location	<input type="checkbox"/>			
LookupAcct	Validate input	<input checked="" type="checkbox"/>	SQMed	Validate	False
XForm2Delayed	Root	<input type="checkbox"/>			
XForm2Delayed	Mapping file	<input type="checkbox"/>			
XForm2Delayed	Validate input	<input checked="" type="checkbox"/>	SQMed	Validate	False

- Properties view for a mediation flow
  - Shows all properties for the flow
  - Properties can be edited from this panel
- Review overall usage of promoted properties in the flow
  - Identify shared aliases in the flow

This slide shows the table from the Promotable Properties panel that is associated with a mediation flow. It shows the promotable properties associated with an individual request flow or an individual response flow. It is slightly different than the table used for a mediation primitive's promotable properties. It includes a Primitive column to identify the specific primitive or callout node the property is associated with. This table can be edited to specify whether a property is promoted and for providing group and alias names and values. It provides a good way to review the overall use of promoted properties in a flow and to easily identify shared aliases in the flow. In the screen capture, the highlighted group and alias names for two of the properties show that they have the same values, indicating that they are using a shared alias.

Looking at the screen capture, notice in the Primitive column the entries that are labeled getQuote colon StockPartner. This identifies a promotable property associated with a callout node, rather than a property for a mediation primitive.

## All promotable properties in a component

Mediation Flow: SQMed

Description

Filter: Flow

Display full flow name

Flow	Primitive	Property	Promo...	Group	Alias	Alias value
getQuote	getQuote : DelayedSvcPartner	Include the original re...	<input type="checkbox"/>			
getQuote	getQuote : DelayedSvcPartner	Automatically convert...	<input type="checkbox"/>			
getQuote	getQuote : DelayedSvcPartner	Use dynamic endpoint...	<input type="checkbox"/>			
getQuote	getQuote : DelayedSvcPartner	Async timeout (seconds)	<input type="checkbox"/>			
getQuote	getQuote : DelayedSvcPartner	Invocation Style	<input type="checkbox"/>			
getQuote	getQuote : DelayedSvcPartner	Retry on	<input type="checkbox"/>			
getQuote	getQuote : DelayedSvcPartner	Retry count	<input checked="" type="checkbox"/>	SQMed	RetryCount	2
getQuote	getQuote : DelayedSvcPartner	Retry delay (seconds)	<input type="checkbox"/>			
getQuote	getQuote : DelayedSvcPartner	Try alternate endpoints	<input type="checkbox"/>			
getQuote	getQuote : StockQuoteService	Use dynamic endpoint...	<input type="checkbox"/>			
getQuote	getQuote : StockQuoteService	Automatically convert...	<input type="checkbox"/>			
getQuote	LookupAcct	Table	<input type="checkbox"/>			
getQuote	LookupAcct	Search column	<input type="checkbox"/>			
getQuote	LookupAcct	Search location	<input type="checkbox"/>			
getQuote	LookupAcct	Validate input	<input checked="" type="checkbox"/>	SQMed	Validate	false
getQuote	XForm2Delayed	Root	<input type="checkbox"/>			
getQuote	XForm2Delayed	Mapping file	<input type="checkbox"/>			
getQuote	XForm2Delayed	Validate input	<input checked="" type="checkbox"/>	SQMed	Validate	false
getQuote	XFormResponse	Root	<input type="checkbox"/>			
getQuote	XFormResponse	Mapping file	<input type="checkbox"/>			
getQuote	XFormResponse	Validate input	<input checked="" type="checkbox"/>	SQMed	Validate	false

- Properties for an entire mediation flow component
  - Properties view of the Overview panel
  - Adds Flow column identifying the flow containing the property
- Use this to check and verify appropriate shared alias usage

This slide shows the Promotable Properties panel that is associated with an entire mediation flow component, encompassing all the request and response flows defined within the component. It is part of the Properties view of the Overview panel of the Mediation Flow Editor. It is similar to the table for an individual request or response flow, with the addition of the Flow column and a property to indicate if full or abbreviated flow names should be used. The addition of the flow name helps to distinguish between primitives with the same name that are in different flows.

The same editing capabilities described for the table on the previous slides are available for this table as well. The key use for this particular table is to ensure that you have configured your shared aliases correctly. A more detailed discussion of shared alias usage is provided in the subsequent presentation.

## All promotable properties in a module

- Promoted properties at the module level
  - Modules and mediation modules can have more than one mediation flow component
  - Shared aliases are scoped at the module level
    - Promoted properties from two different mediation flow components with the same group and alias name are shared at runtime
  - The default group name assigned is based on the module name and mediation flow component name
    - By default there is no sharing across components
    - Sharing across components requires explicit assignment of a common group name

The table on the previous slide addressed all the promotable properties in a mediation flow component. In most cases, there is only one mediation flow component in a module or mediation module. When that is the case, the previous table is sufficient to review promotable and promoted properties for an entire module. However, it is possible to have more than one mediation flow component in a module. The considerations for that are addressed on this slide and the next.

Shared aliases are scoped at the module level. Therefore, when promoted properties from two different mediation flow components have the same group and alias names, they are shared at runtime. By default, the group name assigned to each promoted property is constructed from the module name and mediation flow component name. This means that by default, there are no shared aliases across components. If you want to have shared aliases across components, you must explicitly set the group name for each property to a common value.

## Pitfalls when sharing across multiple modules

- There are pitfalls you need to be aware of when sharing across modules
  - There is no mechanism provided to review all the promotable/promoted properties at the module level
    - Makes it difficult to review for shared alias usage
  - There is no development time management of shared alias values across mediation flow components, as illustrated by this example:
    - Four properties share the same group and alias name
    - Two of the properties are in one module, the second two are in another module
    - Updating the value of one of the properties:
      - » Automatically updates the value of the other in the same module
      - » Does not update the two in the other module
    - However, at runtime all four properties will share the same value

There are some pitfalls you should be aware of when sharing promoted properties across multiple modules. First of all, there is no mechanism provided to review all of the promotable and promoted properties at the module level. This makes it difficult to review the promoted properties to verify that the shared aliases defined are in fact those that you intended. Secondly, there is no management of shared alias values across mediation flow components. This is best illustrated by an example. Assume there are four properties that share the same group name and alias name. Two exist in one mediation flow component and the other two are in a different mediation flow component. When you update the value for one of the properties the value for the shared property in the same module is also update, but the values for the shared properties in the other module are not updated. So from a development time perspective, the properties in the different modules appear to have different values. However, at runtime all four will share the same value.

Considering this, you need to be very careful if you are sharing promoted properties in a module with multiple mediation flow components.



## Summary

- Basic concepts and terminology
  - What are properties and promoted properties
  - Introduce three major variations for how promoted properties can be used in a flow
  - Provide example scenarios
  - Introduce the terminology fundamental to understanding promoted properties
- Configuration of promoted properties
  - Defining promoted properties
  - Reviewing promoted properties defined at various scopes

In summary, the presentation started by defining what a property is and what a promoted property is. The three essentially different usage patterns for promoted properties were introduced along with mention of some example scenarios. Next, the specific terminology required for discussing promoted properties was introduced.

In the section on configuration, you were shown how promoted properties are configured using WebSphere Integration Developer. Also introduced were the panels that can be used to review the configuration of promoted properties across different scopes.



## 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 e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_WBPMv7\\_PromotedPropertiesBasics.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WBPMv7_PromotedPropertiesBasics.ppt)

This module is also available in PDF format at: [..\\WBPMv7\\_PromotedPropertiesBasics.pdf](..\\WBPMv7_PromotedPropertiesBasics.pdf)

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



## 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](http://www.ibm.com/legal/copytrade.shtml)" 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.

© Copyright International Business Machines Corporation 2010. All rights reserved.