WebSphere® Enterprise Service Bus V6.0.2
WebSphere Process Server V6.0.2
WebSphere Integration Developer V6.0.2

Promoted properties – Common details

New V602

Promoted properties for mediation primitives and flows was newly introduced in V6.0.2. This presentation looks at the common aspects of promoted properties from both a development and a runtime administration perspective.

Goals

- Introduce the concepts of promoted properties
  - Basics of function and terminology
  - Configuration in WebSphere Integration Developer
  - Runtime administration
  - Best practices for usage

- Provide the basic understanding needed before examining individual primitives

2

The goal of this presentation is to introduce you to the concepts of promoted properties. The presentation starts out by providing some background information and introducing the terminology that is used with promoted properties. You are then shown how promoted properties are configured using WebSphere Integration Developer. The major purpose of promoted properties is to enable administrators to have some dynamic control over certain aspects of a mediation flow. Therefore, the next topic examined are the various runtime administrative tasks and capabilities. Once you have the full end to end picture, additional details are presented with an emphasis on best practices for the use of promoted properties. With the information you learn from this presentation, you will be ready to understand the specifics of promoted properties for each of the individual mediation primitive types.

Section

Basics of function and terminology

The first section of the presentation introduces the basic functions and characteristics associated with promoted properties. In addition, the key terms that are used when discussing promoted properties are introduced and explained.

Why promoted properties?

- Characteristics without promoted properties:
  - ▶ Property values are statically defined during development
  - ▶ Property values cannot be modified at runtime
    - Changes can only be made using WebSphere Integration Developer
    - The mediation module must be redeployed to the server.

- Promoted properties enable dynamic runtime changes, for example:
  - ▶ How much of the message to log
  - ▶ Toggle input validation for mediation primitives
  - ▶ Change values used to control flow paths

This slide addresses the rational for having promoted properties.

In the absence of the promoted properties capability, mediation property values are statically defined at development time using WebSphere Integration Developer. The values cannot be modified for a mediation application that is already installed in a WebSphere Process Server or WebSphere Enterprise Service Bus server. If there is a need to modify a property value the mediation must be modified in WebSphere Integration Developer and reinstalled into the server.

Promoted properties enable property values to be updated for a running application. This gives the solution administrator the flexibility to dynamically make changes to the mediation flows. Here are some examples where this could be used.

The first example would be a case where there is 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 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 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 could be used to adjust the rating value threshold used to select a particular path.

Overview of function

- Solution administrators can modify property values
  - ▶ Can be properties of mediation primitives or of callouts
  - ▶ Changes can be applied during application installation
  - ▶ Changes can be applied while the application is running
    - The changes take effect dynamically
    - Server does not need to be restarted
    - Mediation module does not need to be reinstalled or restarted

5

The next couple of slides provide an overview of the functionality of promoted properties. As previously mentioned, promoted properties allow mediation property values to be modified for a running mediation application in a WebSphere Process Server or WebSphere Enterprise Service Bus server. This is normally done by someone assuming the role of solution administrator.

The properties that can be modified include mediation properties associated with mediation primitives in the flow and properties associated with the callout nodes in the flow.

At the time the application is installed, the property values assigned during development are used unless they are explicitly changed during the application installation process. Property values can also be changed for an application that is already installed and running. When this is done, the changes take effect dynamically. There is no need to restart the server, reinstall the application or restart the application.

Overview of function

- Not all properties can be administered
  - Only certain properties are allowed to be administered
    - 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 that will actually be administered

6

© 2007 IBM Corporation

This slide examines which properties are available to have their values changed by a solution administrator.

The first thing to consider is that only certain mediation properties are eligible to be administered. For any given mediation primitive type, only selected properties can be made available to be administered. Those that cannot be administered generally have some characteristic that prevents them from being used in this way. One reason might be that changing the value for a particular property might 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 would require the change to be made using WebSphere Integration Developer. Each mediation primitive type defines which of its properties are potentially available for administration and which are not.

Of those properties which are eligible to be administered, 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 for the solution administrator to modify.

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 module
  - 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 discuss.

The first term in **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 modification by a solution administrator. Whether or not a property is promotable applies to all instances of the mediation primitive type.

The second term is **Promoted**. This term is applied to a property which is not only promotable, but has also been explicitly selected to be made available for modification by a solution administrator. This selection is made by you, the integration developer, when developing a mediation flow. Whether or not a property is promoted applies to only to an 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 solution administrator. The alias name is associated with a value that is applied to the property. So the solution administrator, in essence, does not provide a modified value for the property. Rather, the value 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 which have the same alias name share the same alias value. This allows multiple related properties within a mediation module to be administered together. An example may help to illustrate this. On a previous slide there was a scenario described where multiple validate input properties were to be toggled from off to on. By having a shared alias for these multiple validate input properties, the solution administrator could modify them all by only having to change a single alias value.
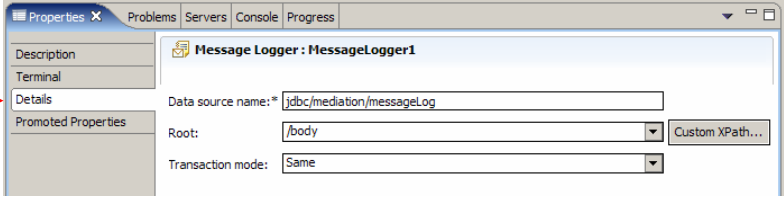
Section

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.
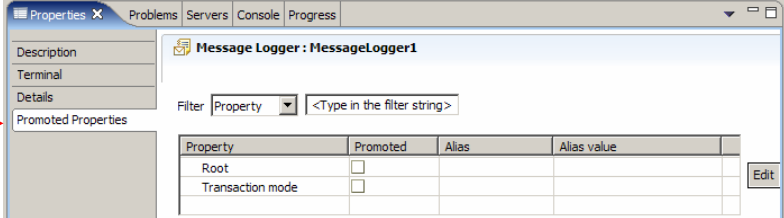
Promoted Properties panel

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

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

In the Properties view for a mediation primitive, two of the panels provided are the Details panel and the Promoted 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 a Message Logger primitive which as three properties, the Data source name, the Root and the Transaction mode.

Looking at the Promoted Properties panel in the lower portion of the slide, you can see that only two of the three properties are promotable. The Root and the Transaction mode are promotable whereas the Data source name is not. The Promoted 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 Alias and Alias value columns come into play.

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

Promoting a property

Promoting a property
- Root property
  - Select as promoted on Promoted Properties panel
  - Provide Alias name and value

Property	Promoted	Alias	Alias value	
Root	☑	Msg1-Root	/body	
Transaction mode	☐			

- Icon marks property as promoted on the Details panel
  - Clicking Promoted icon gets you to Promoted Properties panel
  - Alias value is displayed but grayed out – cannot be changed here

Data source name:* jdbc/mediation/messageLog

Root: /body   Custom XPath...

Transaction mode: Same

This slide illustrates the promoting of a property. The Promoted Properties panel is shown on the top and you can see that the Root property has been marked as promoted along with values for Alias name and Alias value being filled in.

Looking at the Details panel on the bottom, you can see that the Root property looks somewhat different than it did in the previous slide. First of all, an icon has been placed next to the property label 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 on the icon, it switches you to the Promoted 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 prompted
  - 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, an alias and value are assigned
  - Table row must be created before it shows up as available to promote

Filters:

Pattern	Terminal name
/body/getCustomerID/input1[age>64]	senior
/body/getCustomerID/input1[age>19]	adult
/body/getCustomerID/input1[age>12]	teenager
/body/getCustomerID/input1[age<1]	baby

Property	Promoted	Alias	Alias value
Pattern{/body/getCustomerID/input1[age>12]}	☐		
Distribution mode	☐		
Pattern{/body/getCustomerID/input1[age>19]}	☐		
Pattern{/body/getCustomerID/input1[age>64]}	☑	Senior	/body/getCustomerID/input1[age>64]
Pattern{/body/getCustomerID/input1[age<1]}	☑	Baby	/body/getCustomerID/input1[age<1]

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 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 Promoted Properties panel so that then 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 1 are considered baby. This leaves the default flow to be for customers 1 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 1 patterns are shown as being promoted. Now look at the Promoted Properties panel on the lower right side. You can see the Pattern from the table row is used as the Property. For those rows that are promoted, the pattern also 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.

This slide shows the Promoted Properties panel that is associated with a mediation flow. It shows the promotable properties associated with a request flow or a response flow. It is slightly different than the table used for a mediation primitive's promoted properties. It includes a Primitive column to identify the specific primitive the property is associated with. This table can be edited to specify whether or not a property is promoted and for providing 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.

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

All promoted properties in a mediation module

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

Promoted properties – Common details   © 2007 IBM Corporation

This slide shows the Promoted Properties panel that is associated with an entire mediation module. It is part of the Properties view of the Operation Connections panel of the Mediation Flow Editor. It adds a Flow column which identifies the flow with which a property is associated. It has the same editing capabilities as described on the previous slides.

The key use for this particular table is to ensure that you have configured your shared aliases correctly, which is discussed in more detail later in this presentation.

Section

Runtime administration

This section shows you about how promoted properties are administered at runtime, looking at both installation and dynamic runtime updates.

Application installation

Administrative console application installation

- Alias value specified in WebSphere Integration Developer is the default

- Alias value can be changed during installation

Edit Module Properties panel

Install New Application

Specify options for installing enterprise applications and modules.

Step 1 Select installation options

Edit module properties

Edit any of the current module properties.

Step 2 Map modules to servers

InboundMsgLog   XPATH   /body

Step 3 Provide options to perform the EJB Deploy

LookupKeyColumn   STRING   CUSTOMER_NO

LookupTable   STRING   MY_TABLE

Step 4 Provide listener bindings for message-driven beans

Step 5 Provide JNDI Names for Beans

Alias Name

Step 6 Ensure all unprotected 2.x methods have the correct level of protection

Native Property Type

→ Step 7: Edit module properties

Editable Alias Value

Step 8 Summary

Previous   Next   Cancel

IBM Software Group

Promoted properties – Common details

© 2007 IBM Corporation

15

This slide shows a screen capture of the Administrative Console when doing a new application installation for an application containing a mediation module. The left side contains a list of panels which are relevant to the installation of an application. Shown in the screen capture is the Edit module properties panel which provides the ability to modify the alias values for the promoted properties. You can see in the screen capture that each alias is listed by name along with it native property type. The alias values are presented in an editable form, initialized to the values that were specified for them in WebSphere Integration Developer. The solution administrator has the opportunity to specify new alias values at this time.

Application installation

- Command line application installation
  - Normal use of: wsadmin $AdminApp install
  - Use SIBSCAClientInstall to specify alias override values
    - { -SIBSCAClientInstall {{,<aliasName>=<aliasValue>}}}
    - Aliases not specified default to value specified in WebSphere Integration Developer
  - Example:

```
wsadmin>$AdminApp install InstallExample.ear
  {-SIBSCAClientInstall
   {
      { InboundMsgLog=/body,
        LookupKeyColumn=CUSTOMER_NO,
        LookupTable=MY_TABLE }
   }
  }
```

This slide also describes application installation, but for command line usage rather than through the Administrative Console. It makes use of the **wsadmin $AdminApp install** command with the **SIBSCAClientInstall option**. The syntax for the command and an example are shown in the slide. Any aliases not specified using this option default to the value given to the alias in the WebSphere Integration Developer.

Dynamic runtime updates

This slide shows how to display and update aliases for a mediation module in an installed application.

To navigate to the aliases, you open Applications and then select SCA Modules. This puts you in a panel listing all of the SCA modules within the scope of the Administrative Console. Selecting a particular module puts you into the configuration panel for that module, on which you then select Module Properties.

You are now shown the configuration panel listing all of the aliases, their native property types and editable values. If you edit a value, you need to apply the change and then save the configuration as you would with any other change made using the Administrative Console.

Once the configuration is saved, the new alias value applies to the mediation module. Any in flight instances of a mediation flow continue to use the old value, but new instances that are started make use of the new value. As with any server configuration change, if this is a Network Deployment environment, there may be delays between when the configuration is saved and when the changes take affect on individual servers within the cell.

Section

Additional details and best practices

To fully understand promoted properties, there are some additional details that you need to be aware of. This section looks at some of those details and addresses some best practices for the use of promoted properties.

Alias names and shared aliases

- Alias names
  - ‣ <mediation_primitive_name>.<property_name> is the default
  - ‣ You can change the default to another more meaningful name

- Shared alias
  - ‣ Multiple properties can share a common alias
    - Scope of sharing is the Mediation Module (not just an individual flow)
    - Using the same alias name defines the sharing association
  - ‣ Alias value updating
    - Can update on Promoted Properties panel for any of the properties sharing the alias
    - Shared value is automatically updated for all properties sharing the alias
  - ‣ Normally used to control a common property
    - Example – toggle Input Validation for multiple primitives in the flow

This slide goes into slightly more detail on alias names and shared aliases then what was previously covered.

First, when your mark a property as being promoted, WebSphere Integration Developer assigns an alias name composed from the mediation primitive name dot property name, as shown in the slide. In general, you should change this generated name to something that is more meaningful to both you and the solution administrator.

As has been discussed, multiple properties can have a shared alias by using the same alias name, which results in them having the same value assigned. The scope of this sharing is the entire mediation module, not just across an individual flow. Because of this, it important to make use of the Promoted Properties panel associated with the Operation Connections panel of the Mediation Flow Editor, which was shown on a previous slide. This allows you to check all aliases assigned in the flow and determine if all the sharing relationships are correct.

The next point to mention is related to the updating of the value for a shared alias. The value for the shared alias can be updated on any Promoted Properties panel on which it is found. When the value is updated, it is be reflected in all other Promoted Properties panels which contain that alias.

An example of the usage of a shared alias would be to toggle the Validate Input property contained on some mediation primitives. By having a shared alias for all of these properties, validation can be easily be turned on for debugging and turned off when in a normal running environment where performance is important.

Enumerated values

- Representation of enumerated values
  - WebSphere Integration Developer displays meaningful values for properties with enumerated values
  - However, at runtime alias values are String representations of integers
    - Promoted properties panel shows both the meaningful value and the integer value
  - WebSphere Integration Developer ensures value set is a valid choice
    - In Details panel
    - In Promoted Properties panel

Alias in Promoted Properties panel

Property	Promoted	Alias	Alias value
Transaction mode	☑	TxMode	Same (0)

Property in Details panel

Transaction mode: Same ▼
Same
New

Alias in Administrative Console

Configuration

General Properties

TxMode   STRING 0

- High potential for error by runtime administrator using administrative console
  - No way to check for correct value
  - Use of integer rather than meaningful value

20

One of the things to be aware of with promoted properties is the representation of enumerated values for properties. In WebSphere Integration Developer, if some property has a defined set of values, they are displayed to you with meaningful names. However, in the artifacts that are generated for the runtime to use, the meaningful name is normally replaced with some integer value expected by the runtime. When one of these integer values is used as an alias value, it is in fact treated as a String representation of the integer value. Because of this, the solution administrator is not presented with alias values that are meaningful to a human. Looking at the example, you can see on the right that the Transaction mode property of a Message Logger primitive can be set to either Same or New. For the runtime, this choice is represented by the integers 0 and 1. When this property is promoted, WebSphere Integration Developer displays both the meaningful and integer value in the Promoted Properties panel so that you can know what integer is assigned to that choice. Finally, in the Administrative Console, the solution administrator is only presented with the String representation of the integer value.

This overall situation presents a few problems. First, is that the solution administrator must be made aware of what the alias values of 0 and 1 actually mean. A second problem has to do with specification of an invalid value. In WebSphere Integration Developer, you cannot set any other value for Transaction Mode than Same or New. However, at runtime, the only checking done when the alias value is updated is to insure it is a valid String. If the solution administrator provides any other value than 0 or 1, a runtime error occurs for all subsequent mediation flows using that alias.

Best practice considerations

- Alias names
  - ▶ Do not use the default value
  - ▶ Choose a name that is meaningful to an administrator
- Determining which properties to promote
  - ▶ Only promote properties which can be meaningfully changed for your specific scenario
  - ▶ Decision made through agreement between you and the administrator
- Document usage to administrator
  - ▶ Administrator is not likely to understand details of the flow
  - ▶ Documentation should describe when and why alias value would be changed
  - ▶ Need to document valid values:
    - Numeric representations of enumerations
    - XPath values – valid values and potentially useful values

Here are some best practices that can help you to make the best use of promoted properties.

First, it is always best to use alias names that are meaningful to the solution administrator. They do not have access to the mediation module in WebSphere Integration Developer and are probably not aware of the names of the mediation primitives and properties which have been promoted. They need to have something at makes sense to them in the context of the module they are administering.

Secondly, only promote properties which result in some meaningful usage for your mediation module. This decision should most likely be made through a dialog between yourself as the integration developer and the solution administrator. If the use of the promoted property does not make sense to the solution administrator, it is probably best not to promote it.

Finally, all promoted properties should be well documented for the solution administrator as they are not likely to understand the details of the flow. The documentation should cover every alias name, describing when and why the alias value might be changed. It is very important to document what the valid values are. This is particularly true for enumerated types which do not have meaningful representations in the Administrative Console. It is also true of properties that contain XPath expressions. Which XPath expressions would be valid and why each would be used should be documented.

Aliases and problem determination

- Runtime problems can be caused by:
  - ▸ Specification of an incorrect alias value
  - ▸ An unintended shared alias

- Examine alias values in administrative console
  - ▸ Check for an alias with an invalid value
    - Enumerated value set incorrectly
    - XPath expression incorrect

- Check alias names in WebSphere Integration Developer
  - ▸ Check for inadvertent use of a common alias name
    - Results in a shared alias when that was not your intention
      - Remember, scope of an alias name is the entire mediation module, not just a flow
  - ▸ Check for unshared property which should have been shared
    - Property was supposed to be promoted but was not
    - Property was given an incorrect alias name

Promoted properties – Common details
© 2007 IBM Corporation

As with any functionality that provides flexibility and dynamic capabilities, there is the potential for problems to occur. There can be runtime problems caused by aliases which have been given invalid values. In addition, an unintended or incorrectly configured usage of a shared alias can also lead to unexpected results. This slide examines some of the things you can do when faced with a problem you suspect to be caused by use of an alias.

The first place to start with a runtime error is in the Administrative Console, checking the aliases listed in the Module Properties configuration panel. Make sure that all of the aliases have the right values. Those for enumerated types or XPath expressions are probably the ones most prone to error. If everything appears to be OK, the next place to look would be at the mediation module itself in the WebSphere Integration Developer.

One problem might be the inadvertent use of a common alias name, thus establishing a sharing relationship that you did not intend. Remember that the scope of sharing is the entire module, so use the Promoted Properties panel associated with the Operation Connections of the Mediation Flow Editor to check for this possibility.

Another problem to check for would be when a property was supposed to be shared but does not appear to be using the shared alias value. It could be caused if the property just was not promoted when it should have been and can also occur if the alias name was not specified correctly.

Usage example

- Administratively controlled message logging
  - Transient context contains field to indicate logging should occur
  - Message element setter used to set true/false value in field
    - Property is promoted to allow administrative control of value
  - Message filter controls flow based on value
  - When field is 'true' flow through a message logger

Message Element Setter with promoted property

Administrative Console panel for setting property value

Message Filter routing on property value

23

Promoted properties – Common details   © 2007 IBM Corporation

This example of the use of promoted properties is a scenario that enables administrative control over the whether or not message logging is done.

To enable this, the flow has a transient context with a boolean field used to indicate if logging should occur. Looking at the left side of the slide, you see that there is a Message Element Setter primitive which as been configured to set the field in the transient context to false. The table row in which this is defined has been promoted so that a solution administrator has control over this setting. The next primitive in the flow is a Message Filter which checks to see if the field is true, in which case the flow is send through a Message Logger before going to the callout. If it is false, the flow goes directly to the callout without logging. The bottom panel shows the Administrative Console where the solution administrator can toggle this setting.

Section

Summary

The next slide presents a summary of this presentation.

Summary

- Concepts of promoted properties were presented
    - Basics of function and terminology
    - Configuration in WebSphere Integration Developer
    - Runtime administration
    - Best practices for usage
    - An example

In this presentation you were introduced to the concepts of promoted properties. The presentation started out by providing some background information and introducing the terminology that is used with promoted properties. You were then shown how promoted properties are configured using WebSphere Integration Developer. Since the major purpose of promoted properties is to enable administrators to have some dynamic control over certain aspects of a mediation flow, the next topic examined was the various runtime administrative tasks and capabilities. Having completed the full end to end picture, additional details are presented with an emphasis on best practices for the use of promoted properties. Finally, an example was presented illustrating a potential usage scenario for promoted properties.

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

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

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM   WebSphere

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.