



IBM Software Group

WebSphere® Message Broker V6

Property Enhancements



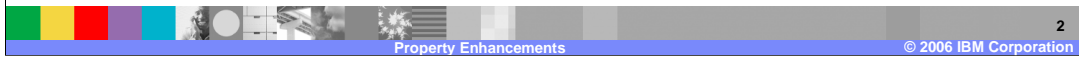
@business on demand.

© 2006 IBM Corporation
Updated November 20, 2006

This presentation discusses Property Enhancements in WebSphere Message Broker Version 6.

Agenda

- Property enhancements
- Summary and references



This topic presents a discussion of the property enhancements in Version 6, followed by a summary and references.

Section

Property enhancements

This section discusses the technical details of the property enhancements in Version 6.

Properties

- The broker maintains a set of properties for environment and flows
- It can be useful, during the runtime of your code, to have real-time access to these properties
- In previous release, you needed to write user-defined extensions to gather equivalent property information
 - ▶ Required coding and more complexity
 - ▶ Required maintenance of these user-defined extensions
 - ▶ Added more system overhead



The WebSphere Message Broker maintains a set of properties for the environment and deployed flows. When you are designing your message flows, it can be very useful to have real-time access to these properties. However, in previous releases, these properties were not available to you, so you needed to write your own user-defined extensions to gather this type of information. This added more work to your development effort and complexity to your message flows. In addition, these additional user-defined extensions had to be maintained. The additional user-defined extensions also created more system overhead which had a negative impact on performance.

Access to properties

- Version 6 provides access to broker properties from the ESQL environment and from Java™ nodes
- If the node allows you to edit ESQL, then you have access to broker properties
 - ▶ Compute nodes
 - ▶ Database nodes
 - ▶ Filter nodes
 - ▶ All derivatives of these nodes



Version 6 now provides the ESQL environment access to broker properties. A subset of these properties is also available to Java nodes as well. In essence, if the node allows you to edit ESQL statements, then you have access to broker properties. This includes compute nodes, database nodes, filter nodes, and all derivatives of these nodes.

Categories of properties

- Categories of broker properties
 - ▶ Node related
 - NodeLabel
 - ▶ Message flow related
 - MessageFlowLabel
 - ▶ General broker properties
 - ExecutionGroupName
 - BrokerName



The three categories of broker properties available are node related, message flow related, and general broker properties.



Variables related to properties

- Null value assigned for unset variables
- If user variable assigned same name as broker variable, user variable takes precedence
 - ▶ You can prefix broker variable name with “SQL.” to access broker variable
- If property is also available from Java nodes, it is available strictly from Java nodes **only** and **not** from Java routines called as ESQL functions or procedures



These properties are referenced by using the assigned variable names documented in the Information center. If a property variable value is unset, it has a null value assigned. If a user variable is assigned the same name as a broker property variable, then the user variable takes precedence. In this situation, you can access the broker variable by prefixing the broker variable name with “SQL.”. (SQL dot) Note that if the property is also available from Java nodes, it is available strictly from Java nodes *only* and *not* from Java routines called as ESQL functions or procedures.

General broker properties

- BrokerDataSourceUserId 
- BrokerDataSource
- BrokerName 
- BrokerUserId
- BrokerVersion
- ExecutionGroupLabel 
- ExecutionGroupName
- Family
- ProcessId
- QueueManagerName 
- WorkPath





Available from Java nodes



Here is a list of the general broker properties that are available from the ESQL environment. The checked items are also available from Java nodes. For more detail about these properties and the properties listed on the next two slides, refer to the topic in the WebSphere Message Broker Version 6 Information center entitled “Broker properties accessible from ESQL and Java”.

Flow properties


- AdditionalInstances
- CommitCount
- CommitInterval
- CoordinatedTransaction 
- MessageFlowLabel 



Available from Java nodes

Here is a list of the flow properties that are available from the ESQL environment. The checked items are also available from Java nodes.

Node properties

- DataSource
- DataSourceUserId
- MessageOptions
- NodeLabel 
- NodeType
- ThrowExceptionOnDatabaseError
- Transaction
- TreatWarningsAsErrors



Available from Java nodes

Here is a list of the node properties or properties that are available from the ESQL environment. The checked item is also available from Java nodes.

Examples

- ESQL

```
DECLARE mybroker CHARACTER BrokerName;
```

```
DECLARE DataSource CHARACTER;           -- Name collision!
```

```
DECLARE brokerDS CHARACTER SQL.DataSource;
                                           --Access broker property
```

- Java

```
String brokerName = getBroker().getName();
```



Here are simple examples showing how to access these properties from ESQL and from Java. Note that the second example for ESQL shows a user-defined variable that is the same name as a general broker property, but by prefixing "SQL." (SQL dot) to the property name, you can still reference the general broker property and not the user-defined variable.

User defined properties

- A user-defined property (UDP) is a property that you define when you construct a message flow
- A UDP allows you to change its initial value at deployment time
- Allows you to design a flow so that a UDP setting can alter the characteristics of a message flow without having to change the code at the message flow node level



A user-defined property, or UDP, is a variable that is defined in your ESQL program by specifying the EXTERNAL keyword on a DECLARE statement.

Using UDPs, configuration data can be set easily and used just like normal constants. Because no external calls to user-written user-defined extensions or parsing of environment trees are involved, the ESQL code is easier to write and maintain, and performs better. Also, the parsing costs of reading data out of trees are removed. The value of the UDP is stamped into the variable at deployment time, which makes it quick to access.

Before you deploy the message flow that uses the UDP, you can change the value of the UDP by using the Broker Archive editor.

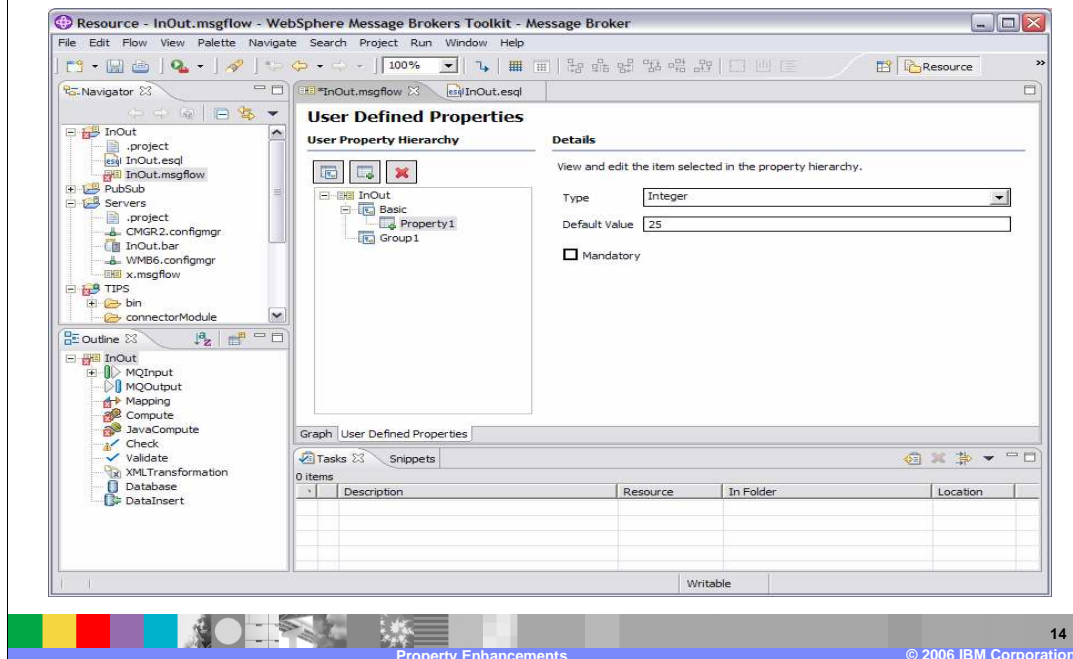
Requirements of UDP

- Define the UDP using the Message flow editor
- Value **MUST** be set by the time you deploy
- Must be defined in ESQL program
 - ▶ DECLARE statement with EXTERNAL keyword
DECLARE UserProperty EXTERNAL CHARACTER;
- Must be retrieved in Java
 - ▶ getUserDefinedAttribute method in any JavaCompute node that uses it
getUserDefinedAttribute(String UserProperty)



A UDP must be defined with the Message flow editor, and its value **MUST** be set by deployment time. It must also be defined with either a DECLARE statement with the EXTERNAL keyword in any ESQL program that uses it, or the getUserDefinedAttribute method in any JavaCompute node that uses it.

Defining a UDP



When the resource that is open is a message flow, at the bottom left of the editor view, there are two tabs, Graph and User Defined Properties.

When the User Defined Properties tab is selected (see screen capture here), a User Defined Properties editor is opened that allows you to change the User Defined Properties of the message flow. The User Defined Properties editor comprises a User Property Hierarchy view and a Details view.

The User Property Hierarchy view provides three icons, Add Property Group, Add Property, and Delete that can be used to update the property hierarchy. When you add a property, a Details view is opened. In the Details view you can define the property Type and Default Value.

Characteristics of UDP

- Any value that you give to a UDP when you define it in a message flow overrides the value of that variable in your ESQL or Java program.
- The value of a UDP can also be modified at deployment time by using the Broker Archive editor to edit the bar file.
 - ▶ BAR editor modification overrides any value that was given when you defined the message flow.
- Every UDP in a message flow **must** have a value at deployment time; otherwise a deployment-time error occurs.
- At run time, after the UDP has been declared, its value can be queried but it cannot be modified.



Any value that you give to a UDP when you define it in a message flow overrides the value of that variable in your ESQL or Java program.

The value of a UDP can also be modified at deployment time by using the Broker Archive editor to edit the bar file. This value, however, overrides any value that was given when you defined the message flow.

Every UDP in a message flow **MUST** have a value, given either when the UDP is declared or by the Message Flow or Broker Archive editor; otherwise, a deployment-time error occurs.

At run time, after the UDP has been declared, its value can be queried by subsequent program statements but its value cannot be modified.

Section

Summary and references

The last portion of the presentation contains a summary and references.

Summary

- **Broker properties available in Version 6**
 - ▶ Provides information efficiently
 - No longer requires you to write your own user-defined extensions to retrieve this information
 - Reduces development complexity and maintenance
 - Reduces system overhead of user-defined extensions to retrieve this information
 - ▶ Accessible from ESQL
 - ▶ Subset of properties accessible from Java
- **User-defined properties (UDP) available**
 - ▶ Value is set by deployment time
 - ▶ Allows message flow characteristics to be changed dynamically at deployment time



In WebSphere Message Broker Version 6, broker properties and user-defined properties (UDP's) are now available. This new functionality allows you to easily and efficiently access important broker properties. You no longer need to write your own user-defined extensions to retrieve this type of information. In addition, development complexity and maintenance overhead are reduced. Run-time system overhead is reduced because the information is provided by broker functionality rather than by running user user-defined extensions to retrieve property information.

User-defined properties - or UDP's - are now available, which allow you to define your own properties with values that are set when the flow is deployed. If you employ UDP's within your flow design, they can provide information that allow the message flow characteristics to be changed dynamically at deployment time without requiring a programming change to the message flow.

References

- WebSphere Message Broker library:

<http://www-306.ibm.com/software/integration/wbimessagebroker/library/>

- WebSphere Message Broker Information Center:

<http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp>

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	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)/business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

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 and/or changes in the product(s) and/or program(s) 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 (e.g., 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 2006. 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.