

IBM WebSphere Message Broker and Microsoft .NET in Midmarket Solutions

IBM Redbooks Solution Guide

IBM® WebSphere® Message Broker is a lightweight, advanced enterprise service bus (ESB) that enables the integration of data sources from a wide range of platforms across service-oriented architecture environments (SOA) and non-SOA environments. You can use IBM WebSphere Message Broker to connect applications together, regardless of the message formats or protocols that they support. This connectivity means that your diverse applications can interact and exchange data with other applications in a flexible, dynamic, and extensible infrastructure. WebSphere Message Broker routes, transforms, and enriches messages from one location to any other location:

- It supports a wide range of protocols: WebSphere MQ, Java Message Service (JMS) 1.1, HTTP and HTTPS, web services (SOAP and REST), file, Enterprise Information Systems (including SAP and Siebel), and TCP/IP.
- It supports a broad range of data formats: binary formats (C and COBOL), XML, and industry standards (including SWIFT, EDI, and HIPAA). You can also define your own data formats.
- It supports many operations, including routing, transforming, filtering, enriching, monitoring, distribution, collection, correlation, and detection.

Two integration capabilities that WebSphere Message Broker provides are of particular interest to many midmarket companies. First is the ability to use WebSphere Message Broker to integrate Microsoft .NET applications into a broader connectivity solution, which is the capability highlighted in this IBM Redbooks® Solution Guide. Second is the ability to use WebSphere Message Broker with WebSphere MQ File Transfer Edition to transfer and use business data that is held in files. Figure 1 shows a high-level overview of the WebSphere Message Broker capabilities and of features that are of special interest to midmarket companies.

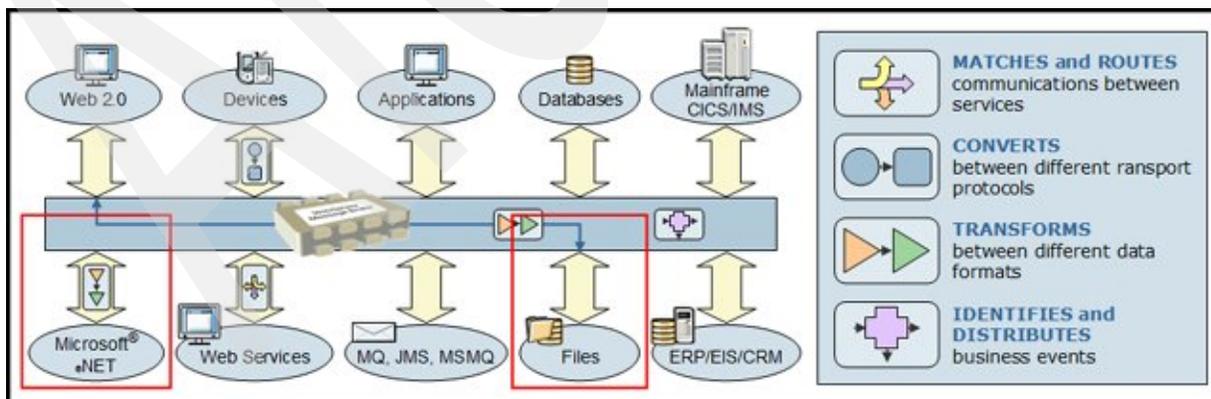


Figure 1. WebSphere Message Broker features and midmarket highlights

Did you know?

WebSphere Message Broker was used to power the mobile application for the Wimbledon tennis tournament in 2010. For more information, see "All England Lawn Tennis and Croquet Club (AELTC) - Wimbledon" at:

<http://ibm.co/OM5s69>

Business value

WebSphere Message Broker provides a broad range of integration capabilities so that companies can rapidly integrate internal applications and connect to partner applications. The types and complexity of the integration required vary by company, application types, and several other factors. With WebSphere Message Broker, organizations of any size can eliminate point-to-point connections and batch processing. Moving forward in this way increases business flexibility and smart interoperability of systems regardless of platform, protocol, or data format. WebSphere Message Broker is frequently associated with large enterprise solutions because of the capabilities to connect to enterprise information systems, such as SAP, Siebel, IBM CICS®, and others. However, IBM also recognized a growing need for ESB capabilities in midmarket companies. For this reason, new features and editions of WebSphere Message Broker provide targeted capability for this growing segment.

WebSphere Message Broker offers comprehensive support for Microsoft .NET environments. Through this support, midmarket clients who have an investment in .NET solutions and skills can enhance their business solutions by using the integration capabilities of WebSphere Message Broker. WebSphere Message Broker can interact with new or existing .NET and Component Object Model (COM) applications. This way WebSphere Message Broker applications can access data that is held in Microsoft business applications and customer relationship management (CRM) suites. The proven scalability and availability of WebSphere Message Broker ensures that clients can continue to grow and protect their software investments.

Solution overview

WebSphere Message Broker V8 introduces the ability to integrate with existing Microsoft .NET Framework applications. A .NET assembly can be called from within a message flow, and the WebSphere Message Broker run time can host and run .NET code. Even the application development aspect is tightly integrated. A new *.NETCompute node* in WebSphere Message Broker can be dragged to a message flow in the canvas in the Message Broker Toolkit. Double-clicking the .NETCompute node opens Microsoft Visual Studio 2010, so that you can develop the code for the node. Programmers can use Microsoft Visual Studio and Common Language Runtime (CLR)-compliant languages, such as C#, Visual Basic (VB), F#, and C++/CLI (Common Language Infrastructure) to build their applications. At the same time, they can integrate the capabilities of these applications into message flows.

Solution architecture

Processing logic in WebSphere Message Broker is implemented by using message flows. Through message flows, messages from business applications can be transformed, augmented, and routed to other business applications. Message flows are created by connecting nodes. WebSphere Message Broker provides a wide selection of built-in nodes. These nodes perform tasks that are associated with message routing, transformation, and enrichment. Message flows are created and tested by using the WebSphere Message Broker Toolkit, a sophisticated, easy-to-use programming tool that provides a full range of programming aids.

A new node, called the *.NETCompute node*, was added to the set of built-in nodes so that message data can be processed by using a CLR-compliant .NET programming language. You can use this new node to build messages, interact with .NET or COM applications, transform messages from one format to another, copy messages, route messages, and modify messages. You must configure the node with a .NET assembly file that contains the code that performs the logic of the node.

The Message Broker Toolkit provides integration with Microsoft Visual Studio 2010, so that you can develop .NET solutions in Visual Studio and integrate them into the message flow. You can develop applications that use the .NET CLR from ESQL procedures or from a .NETCompute node. By using the Message Broker Toolkit, you can rapidly open Visual Studio solutions that are associated with .NETCompute nodes by double-clicking the node and selecting **Open Microsoft Visual Studio**, as shown in Figure 2.

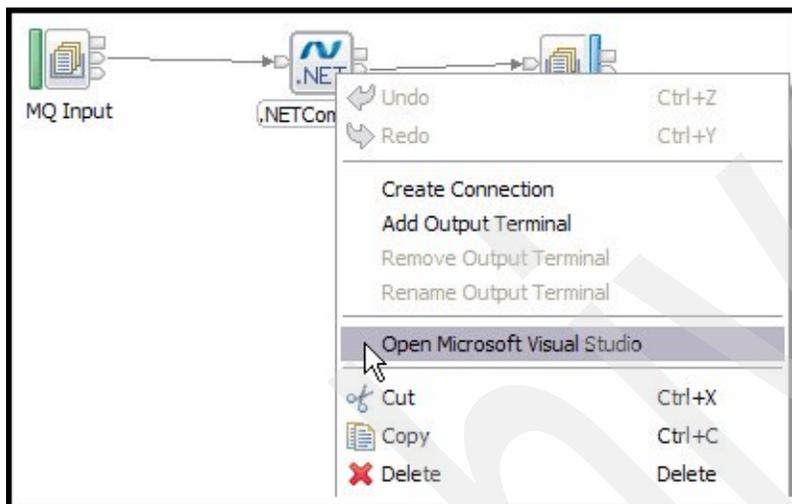


Figure 2: Opening Microsoft Visual Studio from the Message Broker Toolkit

By using Message Broker Toolkit, you can also drag dynamic link libraries (DLLs) that are associated with Visual Studio solutions to the message flow to create a .NETCompute node that is configured for the solution. As a further development aid, the Visual Studio debugger can be attached to the broker execution group process. You can use the Visual Studio debugger with the broker debugger to perform step-through debugging of flows and code.

The WebSphere Message Broker run time hosts and runs .NET code that is associated with a WebSphere Message Broker solution. Message flows that contain .NET code can only be deployed and run successfully on brokers that are on Windows platforms. WebSphere Message Broker can also integrate with Microsoft .NET applications that expose a Windows Communication Framework web service interface. This method is less efficient than a direct invocation by using the .NETCompute node, but enables brokers that are not based on Windows to remotely invoke .NET business logic.

Finally, by using the .NETCompute node, WebSphere Message Broker acts as a bridge between IBM messaging technology (WebSphere MQ) and Microsoft messaging technology (MSMQ). This bridge enables the creation of solutions that span the two messaging environments.

Usage scenarios

This solution guide highlights two usage scenarios:

- Creating a single interface that can connect and integrate heterogeneous data stores for transforming data
- Accessing a Windows Communication Foundation service from a message flow

Scenario 1: Creating a single interface that can connect and integrate heterogeneous data stores for transforming data.

This scenario uses the WebSphere Message Broker capabilities to connect and integrate applications that use Microsoft Dynamics CRM, SAP, and Microsoft SQL Server 2008 databases with an online store scenario. The .NETCompute node is used to evaluate messages and transform the request, providing .NET programmers with a familiar way to add logic to the message flow. The node is also used to interact with the Microsoft SQL Server 2008 database and the Microsoft Dynamics CRM.

A large online store, whose data warehouse is based on an SAP system, recently acquired another online store whose data warehouse is based on a home-grown database system. After the acquisition, the store wants a clean merger with SAP as the master data warehouse and with Microsoft Dynamics CRM for their call center service representatives and any front-end user interface applications.

For example, a customer wants to subscribe to a service with the online store. If the customer is an existing customer, their previous customer information must be associated with the current request. These existing records can be on the SAP system, the database (from the acquisition), or both. Therefore, these systems require one interface and a consolidated view for better customer service.

The IT team deployed WebSphere Message Broker connectivity to connect their acquired SQL-based home-grown data warehouse and SAP systems with Microsoft Dynamics CRM Online. This solution creates a 360-degree view of customers in real time. All account, credit, and sales information can now be sent to Microsoft Dynamics CRM Online from the SAP and database. Figure 3 illustrates the message flow for this scenario.

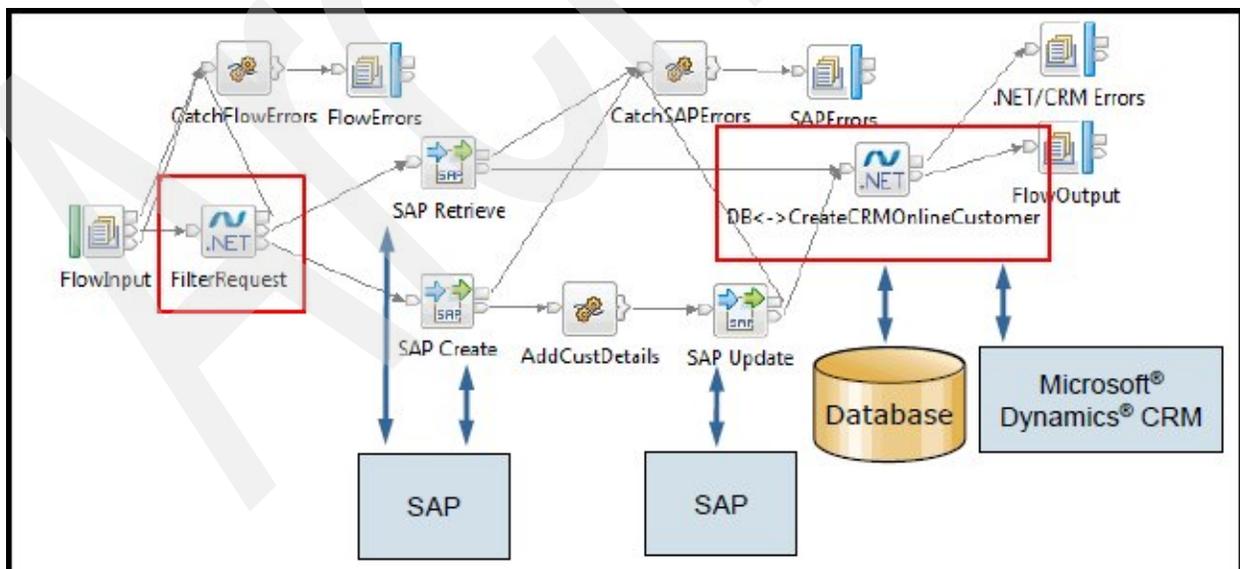


Figure 3. Message flow for scenario 1

This solution includes the following highlights:

- WebSphere MQ messages enter the flow through the FlowInput node and exit through the FlowOutput nodes.
- The FilterRequest .NETCompute node determines whether the request is from an existing or new customer. If the request is from a new customer, the message is sent to the SAP Create node. If the request is from an existing customer, the message is sent to the SAP Retrieve node.
- Three SAPRequest type nodes are used to interact with the SAP system:
 - SAP Retrieve retrieves the Customer records from the SAP system based on a customer number search.
 - SAP Create creates a customer record on the SAP system based on a reference customer.
 - SAP Update updates the newly created customer record with field values.
- A Compute node, called *AddCustDetails*, uses ESQL to construct the SAP request to update the customer details.
- The .NETCompute node DB CreateCRMOnlineCustomer is perhaps the most complex node in the sequence. This node interacts with the Microsoft SQL Server database and creates the customer record in Microsoft Dynamics CRM Online.

Scenario 2: Accessing a Windows Communication Foundation service from a message flow

Windows Communication Foundation (WCF) is a distributed application interface (API) that is part of the Microsoft .NET framework. With WCF, independent applications can interoperate across a network through a message passing interface. In this scenario, a WCF service is consumed by a WebSphere Message Broker message flow to expose the WCF service to a wide range of enterprise applications as part of an ESB.

This scenario (Figure 4) represents an application that is used by a fictional insurance company to process insurance claims. The insurance company provides insurance policy products from financial partners that are transparently under a single brand. The insurance company developed a WCF service, called *ClaimsProcessingWcfService*, to make the insurance policies and claims available to client applications.

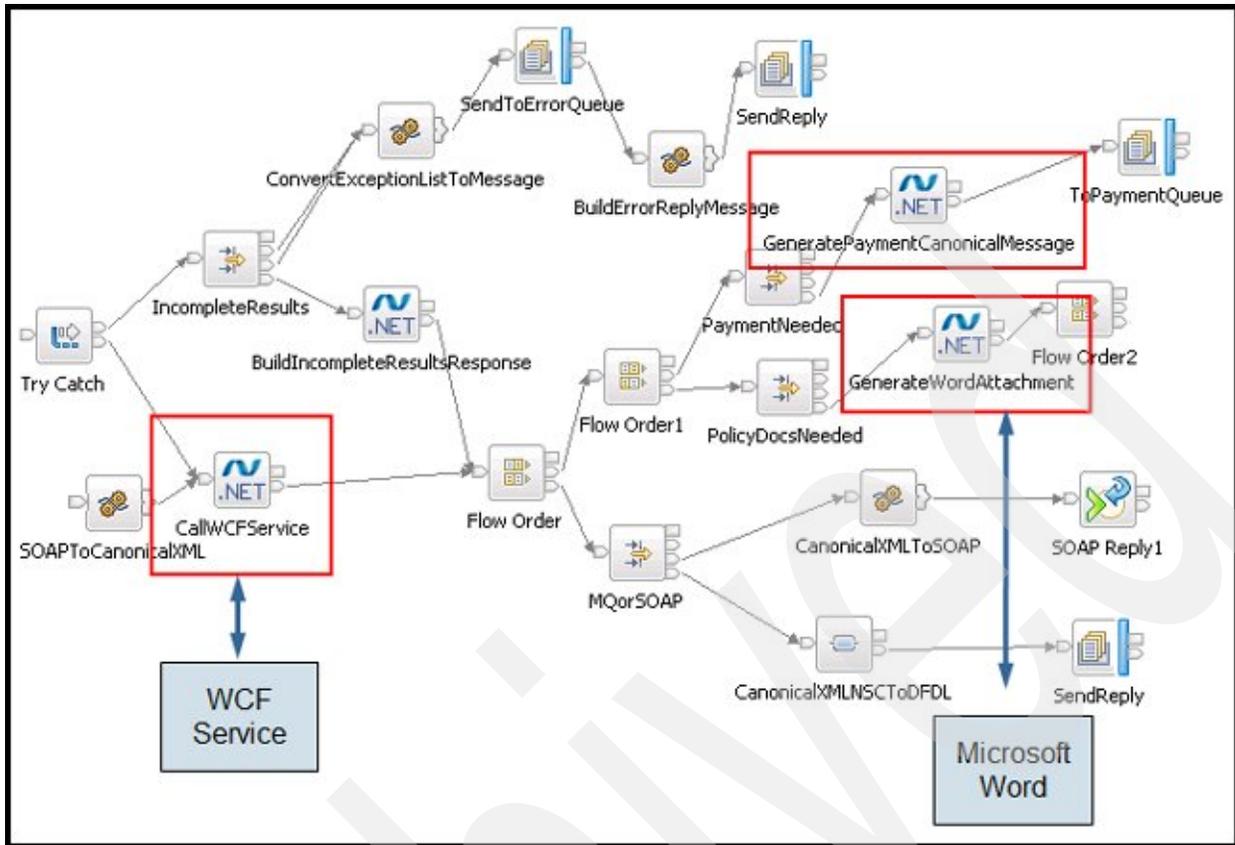


Figure 4. Message flow for scenario 2

The WCF service consists of several business objects that operations can be performed on, including customers, policies, claims, and payments. These business objects are kept in a persistent store so that details can be accessed from multiple applications and physical locations. When an operation is performed on one of these business objects, the WCF service stores the updated state of the objects and generates appropriate responses to return to the requesting application. The operations that are supported are ViewClaim, ViewPolicy, ViewOutstandingClaims, CreateClaim, CreatePolicy, AcceptClaim, and RejectClaim.

A message flow accesses the WCF service to integrate policy and claims information into the WebSphere Message Broker application. The message flow uses the .NETCompute node to embed a WCF client for the service by using a TCP/IP-based transport.

The message flow follows this sequence:

- An input message is received on one of the input nodes for the flow. The message is transformed into a Canonical Message Format that is suitable for transmission across other ESB services (not shown).
- The CallWCFService node executes the .NET application that contains the WCF client code for the WCF service. When the node is reached in the message flow, a WCF request is made to the ClaimsProcessingWcfService by using TCP/IP. The WCF service executes the operation.

- Additional processing that is required for this operation is performed by the .NETCompute nodes:
 - When a new policy is created, a Microsoft Word document that contains a policy confirmation letter is generated. This letter is emailed to the customer's email address and is written to the local file system so that it can be printed and mailed to the customer's physical address.
 - When an insurance claim is approved, a payment is issued by sending the appropriate data to the financial partner that underwrites the particular policy that the customer bought. This data is placed on an MQ queue and is assumed to exchange data in the Canonical Message Format.
- A reply message is sent to the originator of the request with the results from the operation.

Integration

WebSphere Message Broker interfaces with other IBM *products*, and with products from other software vendors, to provide services that enhance message processing in the broker:

- IBM WebSphere Service Registry and Repository
- IBM WebSphere Process Server and IBM Process Server
- IBM WebSphere Integration Developer and IBM Integration Designer
- IBM WebSphere Business Monitor
- IBM WebSphere Transformation Extender
- IBM WebSphere MQ File Transfer Edition
- Enterprise Information Systems
 - SAP
 - Siebel
 - Peoplesoft
- Security
 - IBM Tivoli® Federated Identity Manager (TFIM)
 - Lightweight Directory Access Protocol (LDAP)
 - IBM RACF® and other External Security Managers, on IBM z/OS® only
- Management
 - Tivoli License Manager
- Citrix Presentation Server

WebSphere Message Broker interfaces with other IBM *resources*, and with resources from other software vendors, to provide extensions to message processing:

- Databases and ODBC support
 - IBM DB2®
 - Oracle
 - Sybase
 - Microsoft SQL Server
 - IBM Informix®
- Databases and JDBC support
 - IBM DB2 Driver for JDBC and SQLJ
 - Microsoft SQL Server 2005 JDBC driver
 - Oracle JDBC Driver
 - Sybase jConnect for JDBC
 - IBM Informix JDBC

- File systems
 - FTP
 - SFTP
- Email systems
- Development repositories
 - IBM Rational® ClearCase®
 - CVS
 - Other repositories supported by Eclipse
- Compilers (C and Java)
- Browsers
 - Microsoft Internet Explorer
 - Mozilla
- Adobe Flash Player for Quick Tour
- Adobe Acrobat for reading PDF files

Products and applications that adhere to certain specifications can also interact with WebSphere Message Broker:

- Java and JMS providers
- XSD
- WSDL
- SOAP
- XSLT
- WS-Addressing
- WS-Security

Supported platforms

WebSphere Message Broker supports IBM AIX®, HP, Linux, Solaris, Windows, and z/OS. However, when using the .NETCompute node, be aware that the WebSphere Message Broker run time hosts the .NET code that is associated with a WebSphere Message Broker solution. That is, message flows that contain .NET code can only be deployed and run successfully on brokers that are on Windows platforms.

For specific information about software and hardware requirements, see the WebSphere Message Broker product page at:

<http://www.ibm.com/software/integration/wbimessagebroker/requirements>

Ordering information

Table 1 shows the ordering information.

Table 1. Ordering part numbers and feature codes

Program name	PID number	Charge unit description
IBM WebSphere Message Broker V8.0	5724-J05	Processor Value Unit
IBM WebSphere Message Broker Hypervisor Edition V8.0 for IBM AIX	5725-B71	Processor Value Unit
IBM WebSphere Message Broker Hypervisor Edition V8.0 for Red Hat Enterprise Linux Server for x86	5725-B72	Processor Value Unit
WebSphere Adapter for JD Edwards EnterpriseOne	5724-N41	Application Instance
WebSphere Adapter for JD Edwards EnterpriseOne	5724-N41	Establishment
IBM WebSphere Adapter for PeopleSoft Enterprise	5714-L81	Per Application Instance
IBM WebSphere Adapter for PeopleSoft Enterprise	5714-L81	Establishment
IBM WebSphere Adapter for SAP Software	5724-L79	Per Application Instance
IBM WebSphere Adapter for SAP Software	5724-L79	Establishment
IBM WebSphere Adapter for SAP Software System z®	5724-L79	Establishment
IBM WebSphere Adapter for Siebel Business Applications	5724-L80	Application Instance
IBM WebSphere Adapter for Siebel Business Applications	5724-L80	Establishment

IBM WebSphere Message Broker is also available from IBM Passport Advantage.

For complete ordering information, see "IBM WebSphere Message Broker V8.0 delivers enhancements for connectivity and an improved development experience, and adds new support for virtualization environments on IBM AIX" at:

<http://ibm.co/T1XbHj>

Related information

For more information, see the following documents:

- IBM WebSphere Message Broker product page
<http://www.ibm.com/software/integration/wbimessagebroker>
- IBM WebSphere Message Broker Sales Manual
<http://ibm.co/RkGkSO>
- IBM WebSphere Message Broker announcement
<http://ibm.co/T1XbHj>
- IBM WebSphere Message Broker Information Center
<http://publib.boulder.ibm.com/infocenter/wmbhelp/v8r0m0/index.jsp>
- IBM WebSphere Message Broker Requirements
<http://www.ibm.com/software/integration/wbimessagebroker/requirements>

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. 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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

© Copyright International Business Machines Corporation 2012. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This document was created or updated on November 8, 2012.

Send us your comments in one of the following ways:

- Use the online **Contact us** review form found at:
ibm.com/redbooks
- Send your comments in an e-mail to:
redbook@us.ibm.com
- Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400 U.S.A.

This document is available online at <http://www.ibm.com/redbooks/abstracts/tips0927.html> .

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml

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

AIX®	Rational®
CICS®	Redbooks®
ClearCase®	Redbooks (logo)®
DB2®	System z®
IBM®	Tivoli®
Informix®	WebSphere®
Passport Advantage®	z/OS®
RACF®	

The following terms are trademarks of other companies:

Adobe, the Adobe logo, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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