



Informatica® PowerExchange for JMS
10.2

User Guide for PowerCenter

© Copyright Informatica LLC 2009, 2019

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

Informatica, the Informatica logo, PowerCenter, and PowerExchange are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved. Copyright © Thinkmap, Inc. All rights reserved. Copyright © Clearpace Software Limited. All rights reserved. Copyright © Information Builders, Inc. All rights reserved. Copyright © OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright © International Organization for Standardization 1986. All rights reserved. Copyright © ej-technologies GmbH. All rights reserved. Copyright © Jaspersoft Corporation. All rights reserved. Copyright © International Business Machines Corporation. All rights reserved. Copyright © yWorks GmbH. All rights reserved. Copyright © Lucent Technologies. All rights reserved. Copyright © University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright © EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerik Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, Inc. All rights reserved. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jQWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright © MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. All rights reserved. Copyright © MapR Technologies Inc. All rights reserved. Copyright © Amazon Corporate LLC. All rights reserved. Copyright © Highsoft. All rights reserved. Copyright © Python Software Foundation. All rights reserved. Copyright © BeOpen.com. All rights reserved. Copyright © CNRI. All rights reserved.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>), and/or other software which is licensed under various versions of the Apache License (the "License"). You may obtain a copy of these Licenses at <http://www.apache.org/licenses/>. Unless required by applicable law or agreed to in writing, software distributed under these Licenses is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Licenses for the specific language governing permissions and limitations under the Licenses.

This product includes software which was developed by Mozilla (<http://www.mozilla.org/>), software copyright The JBoss Group, LLC, all rights reserved; software copyright © 1999-2006 by Bruno Lowagie and Paulo Soares and other software which is licensed under various versions of the GNU Lesser General Public License Agreement, which may be found at <http://www.gnu.org/licenses/lgpl.html>. The materials are provided free of charge by Informatica, "as-is", without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

The product includes ACE(TM) and TAO(TM) software copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright (©) 1993-2006, all rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (copyright The OpenSSL Project. All Rights Reserved) and redistribution of this software is subject to terms available at <http://www.openssl.org> and <http://www.openssl.org/source/license.html>.

This product includes Curl software which is Copyright 1996-2013, Daniel Stenberg, <daniel@haxx.se>. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://curl.haxx.se/docs/copyright.html>. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

The product includes software copyright 2001-2005 (©) MetaStuff, Ltd. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://www.dom4j.org/license.html>.

The product includes software copyright © 2004-2007, The Dojo Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://dojotoolkit.org/license>.

This product includes ICU software which is copyright International Business Machines Corporation and others. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://source.icu-project.org/repos/icu/icu/trunk/license.html>.

This product includes software copyright © 1996-2006 Per Bothner. All rights reserved. Your right to use such materials is set forth in the license which may be found at <http://www.gnu.org/software/kawa/Software-License.html>.

This product includes OSSP UUID software which is Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at <http://www.opensource.org/licenses/mit-license.php>.

This product includes software developed by Boost (<http://www.boost.org/>) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at http://www.boost.org/LICENSE_1_0.txt.

This product includes software copyright © 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at <http://www.pcre.org/license.txt>.

This product includes software copyright © 2007 The Eclipse Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://www.eclipse.org/org/documents/epl-v10.php> and at <http://www.eclipse.org/org/documents/edl-v10.php>.

This product includes software licensed under the terms at <http://www.tcl.tk/software/tcltk/license.html>, <http://www.bosrup.com/web/overlib/?License>, <http://www.stlport.org/doc/license.html>, <http://asm.ow2.org/license.html>, <http://www.cryptix.org/LICENSE.TXT>, <http://hsqldb.org/web/hsqldbLicense.html>, <http://httpunit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, http://www.gzip.org/zlib/zlib_license.html, <http://www.openldap.org/software/release/license.html>, <http://www.libssh2.org>, <http://slf4j.org/license.html>, <http://www.sente.ch/software/OpenSourceLicense.html>, <http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>, <http://antlr.org/license.html>, <http://aopalliance.sourceforge.net/>, <http://www.bouncycastle.org/licence.html>, <http://www.jgraph.com/jgraphdownload.html>, <http://www.jcraft.com/jsch/LICENSE.txt>, http://jotm.objectweb.org/bsd_license.html, <http://www.w3.org/>

Consortium/Legal/2002/copyright-software-20021231; <http://www.slf4j.org/license.html>; <http://nanoxml.sourceforge.net/orig/copyright.html>; <http://www.json.org/license.html>; <http://forge.ow2.org/projects/javaservice/>; <http://www.postgresql.org/about/license.html>; <http://www.sqlite.org/copyright.html>; <http://www.tcl.tk/software/tcltk/license.html>; <http://www.jaxen.org/faq.html>; <http://www.jdom.org/docs/faq.html>; <http://www.slf4j.org/license.html>; <http://www.iodbc.org/dataspace/iodbc/wiki/IODBC/License>; <http://www.keplerproject.org/md5/license.html>; <http://www.toedter.com/en/jcalendar/license.html>; <http://www.edankert.com/bounce/index.html>; <http://www.net-snmp.org/about/license.html>; <http://www.openmdx.org/#FAQ>; http://www.php.net/license/3_01.txt; <http://srp.stanford.edu/license.txt>; <http://www.schneider.com/blowfish.html>; <http://www.jmock.org/license.html>; <http://xsom.java.net>; <http://benalman.com/about/license/>; <https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>; <http://www.h2database.com/html/license.html#summary>; <http://jsoncpp.sourceforge.net/LICENSE>; <http://jdbc.postgresql.org/license.html>; <http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>; <https://github.com/rantav/hector/blob/master/LICENSE>; <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>; <http://jibx.sourceforge.net/jibx-license.html>; <https://github.com/lyokato/libgeohash/blob/master/LICENSE>; <https://github.com/hjiang/jsonxx/blob/master/LICENSE>; <https://code.google.com/p/lz4/>; <https://github.com/jedisct1/libsodium/blob/master/LICENSE>; <http://one-jar.sourceforge.net/index.php?page=documents&file=license>; <https://github.com/EsotericSoftware/kryo/blob/master/license.txt>; <http://www.scala-lang.org/license.html>; <https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>; <http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>; <https://aws.amazon.com/asl/>; <https://github.com/twbs/bootstrap/blob/master/LICENSE>; <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt>; <https://github.com/documentcloud/underscore-contrib/blob/master/LICENSE>, and <https://github.com/apache/hbase/blob/master/LICENSE.txt>.

This product includes software licensed under the Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>), the Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>), the Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (<http://www.opensource.org/licenses/bsd-license.php>), the new BSD License (<http://opensource.org/licenses/BSD-3-Clause>), the MIT License (<http://www.opensource.org/licenses/mit-license.php>), the Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>) and the Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>).

This product includes software copyright © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://xstream.codehaus.org/license.html>. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit <http://www.extreme.indiana.edu/>.

This product includes software Copyright (c) 2013 Frank Balluffi and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to terms of the MIT license.

See patents at <https://www.informatica.com/legal/patents.html>.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, please report them to us in writing at Informatica LLC 2100 Seaport Blvd. Redwood City, CA 94063.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2019-04-10

Table of Contents

Preface	6
Informatica Resources.	6
Informatica Network.	6
Informatica Knowledge Base.	6
Informatica Documentation.	6
Informatica Product Availability Matrixes.	7
Informatica Velocity.	7
Informatica Marketplace.	7
Informatica Global Customer Support.	7
 Chapter 1: Understanding PowerExchange for JMS.....	8
Understanding PowerExchange for JMS Overview.	8
Integrating PowerCenter with JMS.	8
Designer Integration with JMS.	8
JMS Message Structure.	9
PowerCenter Integration Service and JMS Integration.	10
Understanding Secure Sockets Layer.	11
Authentication Components	11
Communication Between a JMS Provider and the PowerCenter Integration Service.	11
 Chapter 2: Configuring PowerExchange for JMS.....	13
Configuring PowerExchange for JMS Overview.	13
Minimum System Requirements.	13
Step 1. Configure JVM Options for the PowerCenter Integration Service.	14
Step 2. Register JMS Provider Client Libraries.	14
Copying JMS Provider Libraries to the PowerCenter Integration Service Directory.	14
Setting the Classpath to JMS Provider Client Libraries on Windows.	14
Registering JMS Provider Client Libraries on UNIX.	14
 Chapter 3: Working with JMS Sources and Targets.....	15
Working with JMS Sources and Targets Overview.	15
Editing JMS Definitions.	16
Working with Header Fields in a JMS Target.	16
Maintaining Transactional Consistency for JMS Targets.	17
Creating JMS Source and Target Definitions.	17
Editing JMS Source and Target Definitions.	17
Working with the Application Source Qualifier.	19
 Chapter 4: Creating and Configuring JMS Workflows.....	20
Creating and Configuring JMS Workflows Overview.	20

Loading JVM.	20
Reading and Writing JMS Messages in Sequence.	20
Working with JMS Sessions.	21
Filtering Source Messages.	21
Configuring Terminating Conditions.	21
Configuring Real-time Processing.	22
Configuring Durable Subscriptions.	22
Configuring Message Recovery.	23
Configuring JMS Header Fields.	23
JMS Transacted Mode.	23
Configuring Transactional Consistency for JMS Targets.	24
Pipeline Partitioning.	24
Streaming JMS Messages to an XML Parser Transformation.	24
Configuring a JMS Session.	25
Properties Tab - General Options.	25
Config Object Tab - Error Handling.	25
Mapping Tab - Sources Node.	26
Mapping Tab - Targets Node.	27
Scheduling Workflows.	28
JMS Resiliency.	28
JMS Message Queue Resiliency	28
JMS Topic Resiliency.	28
Resiliency During Recovery.	29
JMS Connection Error Codes File.	29
Connection Retry Period.	29
Troubleshooting JMS Workflows.	29
Appendix A: JMS Datatype Reference.....	31
JMS and Transformation Datatypes.	31
Boolean Datatype.	32
Reading From JMS Sources.	32
Appendix B: Glossary of Terms.....	34
Index.	36

Preface

The *Informatica PowerExchange® for JMS for PowerCenter® User Guide* provides information to build mappings to extract data from and load data into JMS messages. It is written for the database administrators and developers who are responsible for extracting data from JMS messages and loading data into JMS messages.

This book assumes you have knowledge of relational database concepts and database engines, PowerCenter, JMS, and JNDI. You should also be familiar with the interface requirements for other supporting applications.

Informatica Resources

Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit <https://network.informatica.com>.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- Review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit <https://kb.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com.

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access PAMs at

<https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions.

If you are an Informatica Network member, you can access Informatica Velocity resources at <http://velocity.informatica.com>.

If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that augment, extend, or enhance your Informatica implementations. By leveraging any of the hundreds of solutions from Informatica developers and partners, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at <https://marketplace.informatica.com>.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through Online Support on Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<http://www.informatica.com/us/services-and-training/support-services/global-support-centers>.

If you are an Informatica Network member, you can use Online Support at <http://network.informatica.com>.

CHAPTER 1

Understanding PowerExchange for JMS

This chapter includes the following topics:

- [Understanding PowerExchange for JMS Overview, 8](#)
- [Integrating PowerCenter with JMS, 8](#)
- [Understanding Secure Sockets Layer, 11](#)

Understanding PowerExchange for JMS Overview

PowerExchange for JMS extracts data from JMS sources and writes data to JMS targets. With PowerExchange for JMS, you can create JMS source and target definitions in the Designer.

The PowerCenter Integration Service connects to a JMS provider when it reads data from JMS or writes data to JMS. JMS providers are message-oriented middleware systems that send and receive JMS messages. WebSphere MQ JMS is an example of a JMS provider.

The PowerCenter Integration Service can read JMS messages from a JMS provider message queue. Or, it can read messages from a JMS provider based on the message topic. Similarly, the PowerCenter Integration Service can write JMS messages to a message queue. Or, it can write messages to a JMS provider based on the message topic.

This chapter provides an overview of the integration between PowerCenter and JMS.

Integrating PowerCenter with JMS

PowerExchange for JMS enables the PowerCenter Integration Service to read messages from JMS sources and write messages to JMS targets. To read and write messages in JMS message format, create mappings with JMS source and target definitions in the Designer. After you create a mapping, use the Workflow Manager to create a session and workflow. During the session, the PowerCenter Integration Service connects to JMS providers to read and write JMS messages.

Designer Integration with JMS

You manually create JMS source and target definitions to reflect the message structure of JMS messages.

JMS Message Types

The PowerCenter Integration Service can read and write the following JMS message types:

- **Message.** Contains only header and properties fields.
- **TextMessage.** Contains a string object. TextMessages can contain XML message data.
- **BytesMessage.** A stream of uninterpreted bytes. Use a BytesMessage for encoding a message body to match an existing message format. BytesMessages generally do not include property fields.
- **MapMessage.** Contains a set of name/value pairs. The names are in string format. The values are of Java primitive datatypes.

JMS Message Structure

JMS messages contain the following components:

- Header
- Properties
- Body

Header Fields

JMS messages contain a fixed number of header fields. Each JMS message uses these fields regardless of message type. Every JMS source and target definition includes a pre-defined set of header fields.

The following table describes the JMS message header fields:

Header Field	Description
JMSDestination	Destination to which the message is sent. JMS destinations can be a message queue or a recipient who listens for messages based on the message topic.
JMSDeliveryMode	Delivery mode of the message. The delivery mode can be persistent or non-persistent.
JMSMessageID	Unique identification value for the message.
JMSTimestamp	Time at which the message was handed off to the provider to be sent to the destination.
JMSCorrelationID	Links one message with another. For example, JMSCorrelationID can link a response message with the corresponding request message.
JMSReplyTo	Destination to which a reply message can be sent.
JMSType	Type of message based on a description of the message. For example, if a message contains a stock trade, the message type might be stock trade.
JMSExpiration	Amount of time in milliseconds the message remains valid. The messages remain in memory during this period.
JMSPriority	Priority of the message from 0-9. 0 is the lowest priority. 9 is the highest.
JMSRedelivered	Indicates that a message might have been delivered previously, but not acknowledged.

Property Fields

JMS source and target definitions can optionally include message property fields. Property fields contain additional message header information. JMS providers use properties in a JMS message to give provider-specific information. Applications that use a JMS provider can add property fields with application-specific information to a message.

Body Fields

JMS source and target definitions can optionally include a message body. The body contains one or more fields. Only certain types of JMS messages contain a body.

PowerCenter Integration Service and JMS Integration

The PowerCenter Integration Service connects to the Java Naming and Directory Interface (JNDI) server to determine connection information for the JMS provider that is the source or target for the JMS message data. After the PowerCenter Integration Service determines the connection information, it connects to the JMS provider to read or write JMS messages.

The PowerCenter Integration Service can read or write messages based on the message domain. It can read and write JMS messages according to the message topic. Or, it can read messages from a JMS provider queue and write messages to a JMS provider queue.

Connecting to JMS Providers

When you configure a JMS workflow, you use JNDI and JMS application connections to connect to the provider. JMS providers use JNDI to store their connection information. When you configure a JMS provider to send and receive messages, you configure JNDI information for the provider.

JNDI stores the following connection information:

- **Connection factory.** Contains information to create a connection to the JMS provider. For example, a connection factory for a WebSphere MQ JMS provider might contain information about a host name and port number for the provider.
- **Destination.** Contains information about message sources and destinations. For example, a destination for a WebSphere MQ JMS provider might contain information about a queue name and queue manager.

Configure a JNDI application connection to set connection information that enables the PowerCenter Integration Service to connect to the JNDI server. Configure a JMS application connection to provide a connection factory name and a destination name. You also provide a domain for the messages.

The PowerCenter Integration Service connects to the JNDI server based on the JNDI connection information you provide in the JNDI application connection. When it connects to JNDI, it uses the connection factory name you provide in the JMS application connection to retrieve the JMS provider connection information. It also uses the destination name you provide to determine the source and target for JMS messages. It then reads JMS messages or writes JMS messages according to the message domain you specified in the JMS application connection.

Message Domains

The PowerCenter Integration Service can read and write messages with the point-to-point (PTP) or publish-subscribe domain.

The PTP domain means that JMS providers store messages in a message queue. The PowerCenter Integration Service reads messages from the JMS provider message queue. It writes messages to a JMS provider message queue.

The publish-subscribe domain means that JMS providers read and write messages with the message topic as the address. When the PowerCenter Integration Service reads messages, it receives messages with a particular message topic from the JMS provider. When the PowerCenter Integration Service writes messages, it writes messages with a particular message topic to the JMS provider.

Understanding Secure Sockets Layer

The PowerCenter Integration Service uses the Secure Sockets Layer (SSL) protocol to manage the security of messages transmitted on the Web. SSL helps you achieve authentication, integrity, and privacy on the Web. It allows for secure communication between client and server by allowing mutual authentication, use of digital signatures on messages for integrity, and encryption for privacy.

Authentication Components

The PowerCenter Integration Service uses the Public Key Infrastructure (PKI) standard to ensure authentication for JMS messages. PKI includes the following components:

- **Authentication certificate.** A certificate authority provides digital certificates to verify and authenticate parties in internet communications.
- **Trust store.** The PowerCenter Integration Service uses the trust store to authenticate requests from JMS providers. You can store multiple authentication certificates in the trust store.
- **Client store.** The PowerCenter Integration Service sends the client store that contains authentication certificates to JMS providers for authentication. You can store multiple authentication certificates in the client store.

If a JMS provider requires authentication, the PowerCenter Integration Service and the JMS provider authenticate each other with authentication certificates before either transmits data. When a JMS provider sends an authentication certificate to the PowerCenter Integration Service, the PowerCenter Integration Service verifies that the certificate exists in the trust store before it authenticates the JMS provider. When the PowerCenter Integration Service authenticates to a JMS provider, it sends an authentication certificate from its client store.

Communication Between a JMS Provider and the PowerCenter Integration Service

To ensure communication between the JMS provider and PowerCenter Integration Service, you need to configure the ConnectionFactory in PowerCenter. You can use the provider-specific administration tool to configure the ConnectionFactory. The PowerCenter Integration Service uses the ConnectionFactory to establish communication with the JMS provider over a secure communication channel. However, if the lookup for ConnectionFactory uses non-secure JNDI, the communication is not completely secure.

PowerCenter Integration Service performs secure JNDI lookups using the SSL protocol. You can configure the ConnectionFactory with the provider-specific administration tool. When the InitialContext for JNDI is created, configure the SSL properties for the client. The SSL properties are specific to JMS providers. Add the SSL properties to the jndi.properties file and copy it to the PowerCenter Integration Service source files directory.

Specifying SSL Properties

You can use the context factory in the JNDI connection as `FSContextFactory`. To enable SSL, enter the values of the data integration properties `truststore` and `keystore` in the Informatica Administrator as the client trust store and keystore respectively.

CHAPTER 2

Configuring PowerExchange for JMS

This chapter includes the following topics:

- [Configuring PowerExchange for JMS Overview, 13](#)
- [Step 1. Configure JVM Options for the PowerCenter Integration Service, 14](#)
- [Step 2. Register JMS Provider Client Libraries, 14](#)

Configuring PowerExchange for JMS Overview

PowerExchange for JMS installs with the PowerCenter Services. To use PowerExchange for JMS, you must install and configure a JMS provider, such as WebSphere MQ JMS or BEA WebLogic Server. The system administrators should perform the installation and configuration tasks.

To install and configure PowerExchange for JMS, complete the following steps:

1. Configure JVM options for the PowerCenter Integration Service.
2. Register JMS provider client libraries on the machine running the PowerCenter Integration Service process.

After you install and configure PowerExchange for JMS, you can create connections to access JMS providers. Create connection objects in the Workflow Manager so the PowerCenter Integration Service can connect to JMS providers.

Minimum System Requirements

Before you configure PowerExchange for JMS, install and configure a JMS provider. You can install the JMS provider on any machine. However, you must also install the JMS provider client and register the JAR files on the machine running the PowerCenter Integration Service process.

Step 1. Configure JVM Options for the PowerCenter Integration Service

When you configure PowerExchange for JMS, you must configure Java Virtual Machine (JVM) options. Configure JVM options to run Java-based programs. When you configure JVM options, set the following properties in the Administrator tool:

- Java SDK classpath
- Java SDK minimum memory
- Java SDK maximum memory

Step 2. Register JMS Provider Client Libraries

Register JMS provider client libraries on the machine running the PowerCenter Integration Service. The JMS provider client libraries are JAR files that the JMS provider uses to run JMS.

To register the JMS provider libraries, copy the libraries to the `\server\bin\javali` directory. Or, set the `CLASSPATH` to the libraries.

Copying JMS Provider Libraries to the PowerCenter Integration Service Directory

On Windows or UNIX, copy the JMS provider libraries to the following directory:

```
<PowerCenter Installation Directory>\server\bin\javali
```

Setting the Classpath to JMS Provider Client Libraries on Windows

If the JMS provider client libraries are not in the `JRE \lib\ext` directory, set the classpath to each JMS provider client library on Windows.

For example, to set the classpath to the WebSphere MQ JMS client JAR file `com.ibm.mq.jar`, enter the following text:

```
c:\Program Files\WebSphere MQ\JMS\com.ibm.mq.jar
```

Registering JMS Provider Client Libraries on UNIX

If the JMS provider client libraries are not in the `JRE /lib/ext` directory, set the classpath to each JMS provider client library.

For example, to set the classpath to the WebSphere MQ JMS client JAR file `com.ibm.mq.jar` using the C shell, enter the following text:

```
setenv CLASSPATH /home/WebSphere MQ/JMS/com.ibm.mq.jar:$CLASSPATH
```

CHAPTER 3

Working with JMS Sources and Targets

This chapter includes the following topics:

- [Working with JMS Sources and Targets Overview, 15](#)
- [Creating JMS Source and Target Definitions, 17](#)
- [Editing JMS Source and Target Definitions, 17](#)
- [Working with the Application Source Qualifier, 19](#)

Working with JMS Sources and Targets Overview

Every JMS source and target definition contains JMS message header fields. Source definitions contain the header fields used to read messages from JMS sources. Target definitions contain the header fields used to write messages to JMS targets.

The following table shows the message header fields that JMS source and target definitions contain:

JMS Message Header Field	JMS Source Definition	JMS Target Definition
JMSCorrelationID	Yes	Yes
JMSMessageID	Yes	No
JMSType	Yes	Yes
JMSDestination	Yes	Yes
JMSReplyTo	Yes	Yes
JMSDeliveryMode	Yes	Yes
JMSExpiration	Yes	No
JMSTimeToLive	No	Yes
JMSPriority	Yes	Yes

JMS Message Header Field	JMS Source Definition	JMS Target Definition
JMSTimestamp	Yes	No
JMSRedelivered	Yes	No

JMS source and target definitions can also contain message property and body fields depending on which JMS message type the source or target definition represents. JMS source and target definitions can represent metadata for several types of JMS messages.

The following table shows the types of JMS messages that source or target definitions can represent:

Message Type	Description
Message	JMS source and target definitions should represent the message type Message when the source or target messages only need to contain header and property fields. Messages of the type Message cannot contain body fields.
TextMessage	JMS source and target definitions should represent the message type TextMessage when the source or target messages contain XML data. Source and target definitions that represent messages of the type TextMessage contain a single body field and can have multiple property fields.
BytesMessage	JMS source and target definitions should represent the message type BytesMessage when the source or target messages contain a stream of uninterpreted bytes. Source and target definitions that represent messages of the type BytesMessage contain a single body field and can have multiple property fields.
MapMessage	JMS source and target definitions should represent the message type MapMessage when the source or target messages contain name/value pairs. Source and target definitions that represent messages of the type MapMessage can contain any number of body and property fields.

JMS sources and targets that you define represent messages of the type Message by default. Since JMS messages of the type Message do not contain body fields, default source and target definitions do not contain body fields. The default source definition also does not contain property fields. However, you can add property fields to the source definition since JMS messages of the type Message can contain property fields.

Editing JMS Definitions

You can edit a JMS definition to reflect changes in the source or target data. For example, you can edit JMS message header, property, or body fields. You can add and remove fields. You can also change the JMS message type.

When you change the message type, the structure of the source or target definition changes. For example, if you edit the source or target definition to represent the message type TextMessage, the Designer adds a body field to the source or target definition. You can also add any number of property fields.

Working with Header Fields in a JMS Target

When the PowerCenter Integration Service writes a value to the JMSDeliveryMode field in the target, it writes 1 for non-persistent and 2 for persistent.

When the target is a WebSphere MQ JMS provider, and the PowerCenter Integration Service writes a value of -1 to the JMSDeliveryMode field or JMSPriority field in the target, WebSphere MQ JMS writes a value of 4 to the field, and the session completes successfully.

Maintaining Transactional Consistency for JMS Targets

The PowerCenter Integration Service can maintain transactional consistency when writing data to JMS targets. When the PowerCenter Integration Service maintains transactional consistency, it commits messages to JMS targets after all documents in a transaction group are loaded to the targets. A transaction group consists of all documents that the PowerCenter Integration Service commits when it reaches a commit point.

To ensure transactional consistency, all JMS targets in the same pipeline must belong to the same target connection group. JMS targets that are in the same target connection group receive data from the same transactional source. Additionally, two or more JMS targets are in the same target connection group if they have the same value for each of the following connection properties in JMS and JNDI application connections:

- JNDI Context Factory
- JNDI Provider URL
- JNDI UserName
- JNDI Password
- JMS Destination Type
- JMS Connection Factory Name

RELATED TOPICS:

- [“Configuring Transactional Consistency for JMS Targets” on page 24](#)

Creating JMS Source and Target Definitions

You create a JMS source or target definition manually in the Designer. When you manually create a JMS definition, select JMS as the database type. You can also create a JMS target definition based on a JMS source definition.

Editing JMS Source and Target Definitions

You can edit JMS source and target definitions to reflect changes in source or target messages. You can also modify the name and description of the source and target definitions. Edit source and target definitions from the Edit Tables dialog box.

If the source or target is WebSphere MQ JMS, use the following guidelines when editing JMS source and target definitions:

- Property field names cannot begin with JMS_.
- Property and body field names cannot begin with a number.

To edit JMS source and target definitions:

1. For source definitions, double-click the title bar of the source definition in the Source Analyzer. For target definitions, double-click the title bar of the target definition in the Target Designer.

The Edit Tables window appears.

2. Click the JMS Message Header Columns tab.
3. Optionally, edit the datatype and precision of the header columns.
Only some header columns have editable datatype and precision.
4. For JMS target definitions, optionally select the Not Null option for any header field.
When you select Not Null, the field must contain a value when the PowerCenter Integration Service writes messages to the target during the session. Otherwise, the PowerCenter Integration Service generates an error for the field.
For some header fields, you can configure values for the fields in the session properties.
5. Click the JMS Message Property Columns tab.
6. Edit the property fields:
 - To add a new property field, click the Add button. When you add a new field, you can change the name, datatype, and precision of the field.
 - To remove a property field, click the Delete button.
 - To change the order of property fields, use the up and down arrows.
7. For JMS target definitions, optionally select the Not Null option for any property field.
When you select Not Null, the field must contain a value when the PowerCenter Integration Service writes messages to the target during the session. Otherwise, the PowerCenter Integration Service generates an error for the field.
When you leave Not Null blank and the mapping does not contain a link to the field in the target definition, the PowerCenter Integration Service does not include the field in the JMS target messages.
8. Click the JMS Message Body Columns tab.
9. Optionally, select one of the following message types from the Message Body Type list:

JMS Message Type	Description
Message	Designer removes any body fields from the source or target definition.
TextMessage	Designer removes any body fields and adds a new body field called BodyText to the source or target definition. Source and target definitions in TextMessage format can only contain one body field.
BytesMessage	Designer removes any body fields and adds a new body field called BodyBytes to the source or target definition. Source and target definitions in BytesMessage format can only contain one body field.
MapMessage	Designer removes any body fields from the source or target definition. However, you can add new body fields.

10. For messages of the type MapMessage, edit the body fields:
 - To add a new body field, click the Add button. When you add a new field, you can change the name, datatype, and precision of the field.
 - To remove a body field, click the Delete button.
 - To change the order of the body fields, use the up and down arrows.

11. For JMS target definitions, optionally select the Not Null option for any body field.

When you select Not Null, the field must contain a value when the PowerCenter Integration Service writes messages to the target during the session. Otherwise, the PowerCenter Integration Service generates an error for the field.

When you leave Not Null blank and the mapping does not contain a link to the field in the target definition, the PowerCenter Integration Service does not include the field in the JMS target messages.

Working with the Application Source Qualifier

An Application Source Qualifier in a mapping determines how the PowerCenter Integration Service reads data from the JMS source. The PowerCenter Integration Service reads messages from the JMS source based on the connected ports and transformation properties. To extract data from multiple JMS sources, you must have an Application Source Qualifier for each source definition in the mapping.

After you create an Application Source Qualifier, you can set several configuration options. You might want to change numeric or date datatypes to string to maintain accuracy during conversion.

CHAPTER 4

Creating and Configuring JMS Workflows

This chapter includes the following topics:

- [Creating and Configuring JMS Workflows Overview, 20](#)
- [Working with JMS Sessions, 21](#)
- [Configuring a JMS Session, 25](#)
- [Scheduling Workflows, 28](#)
- [JMS Resiliency, 28](#)
- [Troubleshooting JMS Workflows, 29](#)

Creating and Configuring JMS Workflows Overview

Before configuring a JMS session, configure an PowerCenter Integration Service to run JMS workflows. You also need to configure an application connection for JMS sources and targets in the Workflow Manager.

Loading JVM

When the PowerCenter Integration Service connects to JMS during session initialization, it loads JVM. The PowerCenter Integration Service loads JVM once for each session even if the session contains multiple JMS sources and targets. When you run multiple JMS sessions simultaneously, the sessions might require additional system resources.

Reading and Writing JMS Messages in Sequence

JMS does not guarantee the delivery of messages in sequence. Therefore, when the PowerCenter Integration Service reads JMS messages, it might not read the messages in the sequence that the source JMS provider sends the messages. Similarly, when the PowerCenter Integration Service writes JMS messages, the target JMS provider might not receive messages in the order the PowerCenter Integration Service writes the messages to the target.

Working with JMS Sessions

When you configure a JMS session, you define properties that determine how the PowerCenter Integration Service reads messages from a JMS source or writes messages to a JMS target. You can configure the following JMS session properties:

- **Message filtering.** Enter a filter condition to filter the messages from the source.
- **Terminating conditions.** Set terminating conditions to determine when the PowerCenter Integration Service stops reading from the source and ends the session.
- **Real-time data extraction.** Configure flush latency to extract data in real time.
- **Durable subscription.** Use durable subscriptions to preserve messages published on a topic while the subscriber is not active.
- **Message recovery.** Enable message recovery for a JMS session.
- **JMS target header fields.** Configure values for header fields for JMS target messages.
- **JMS Transacted mode.** Configure a session within JMS to run in Transacted mode.
- **Transactional consistency.** Configure transactional consistency for JMS targets.
- **Pipeline partitioning.** Configure pipeline partitions for a JMS session.

Filtering Source Messages

When you configure a session, you can enter a filter condition to filter messages from the source. You can filter messages by header or property fields. For example, you want to read messages from a JMS source that contain the value *buy* in the property field TradeDescription. You can enter the following filter condition:

```
TradeDescription = 'buy'
```

To filter messages from a JMS source, enter a filter condition for the Message Selector attribute in the session properties. Enter a filter condition according to the JMS syntax in the JMS documentation. If you do not enter a valid filter condition, the session fails during initialization.

Configuring Terminating Conditions

Terminating conditions determine when the PowerCenter Integration Service stops reading from the source and ends the session. You can define the following terminating conditions:

- Idle Time
- Message Count
- Reader Time Limit

When you specify values for multiple terminating conditions, the PowerCenter Integration Service stops reading from JMS and ends the session when it meets the first terminating condition. For example, if you set idle time to 10 seconds and message count to 100 messages, the PowerCenter Integration Service stops reading from JMS after 10 seconds or after reading 100 messages, whichever comes first.

Idle Time

Configure idle time to indicate how long the PowerCenter Integration Service waits when no messages arrive before it stops reading from JMS and ends the session. Specify an idle time value in seconds. For example, if you specify 30 for idle time, the PowerCenter Integration Service waits 30 seconds after reading from JMS. If no new messages arrive in JMS within 30 seconds, the PowerCenter Integration Service stops reading from JMS.

Message Count

Configure message count to control the number of messages the PowerCenter Integration Service reads before it stops reading from JMS and ends the session. For example, when you specify 100 for message count, the PowerCenter Integration Service reads 100 messages from JMS.

Reader Time Limit

Configure reader time limit to read messages from JMS for a set period of time. When you use the reader time limit, the PowerCenter Integration Service reads messages from JMS for the time period you specify. Specify a reader time limit value in seconds. For example, if you specify 10 for reader time limit, the PowerCenter Integration Service stops reading from JMS after 10 seconds.

Configuring Real-time Processing

You can configure flush latency to process data in real time. A real-time session reads, processes, and writes data to targets continuously. Flush latency determines how often the PowerCenter Integration Service flushes data from the source.

Configuring Durable Subscriptions

A JMS publisher uses durable subscriptions to save messages published on a topic to a temporary store. If there is no active subscriber, the JMS provider retains messages until the subscriber receives them and acknowledges receipt, until they expire, or until the durable subscription is deleted.

The PowerCenter Integration Service can create a durable subscription to a JMS provider. When the PowerCenter Integration Service connects to a JMS provider, it receives all messages published on a particular topic before session initialization. It also receives the messages that the JMS publisher sends during the session.

If the PowerCenter Integration Service receives durable messages and the session ends before it receives the messages, the messages are available when the JMS subscriber restarts. If the PowerCenter Integration Service does not receive durable messages and the session ends before it receives messages, the JMS provider deletes the messages.

When a session completes successfully, the PowerCenter Integration Service can delete the durable subscription, or it can remain subscribed. If the PowerCenter Integration Service deletes the durable subscription, it can no longer receive messages that the JMS subscriber publishes on the topic when the session is not running. If the PowerCenter Integration Service remains subscribed, it can receive messages that the JMS subscriber publishes on the topic when the session is not running.

To configure a session for a durable subscription, complete the following steps in the session properties:

1. Select a JMS application connection with the topic as the destination type.
2. Set the JMS subscription mode to durable.
3. Enter a durable subscription name.
4. Select to unsubscribe or remain subscribed to the durable subscription.

Enter unique subscription names within a client ID. The JMS provider stores messages published to a topic according to the durable subscription name in the session properties and the client ID in the connection factory. The JMS provider uses the subscription name to deliver messages to the subscriber.

Recovery Performance

When you recover a session that reads durable subscription messages, the following factors can impact recovery performance:

- **Message priority.** During recovery, the PowerCenter Integration Service processes new messages from the JMS publisher if the messages have a higher priority than the recovery messages.
- **Message volume.** During recovery, the PowerCenter Integration Service processes all messages that the JMS publisher sent after the session failed. The JMS publisher can send a large number of messages between the time that the session fails and the time you recover the session.

Configuring Message Recovery

When you configure message recovery for a real-time session, the PowerCenter Integration Service can recover unprocessed messages from a failed session. When you enable message recovery for a real-time session, the PowerCenter Integration Service stores source messages or message IDs in a recovery file, recovery table, recovery queue, or recovery topic. If the session fails, run the session in recovery mode to recover the messages that the PowerCenter Integration Service did not process. For sessions with JMS topics, the PowerCenter Integration Service connects to the recovery topic as a durable subscriber so messages persist even if the session ends.

To configure message recovery for a session that writes to a JMS target, create a recovery queue or recovery topic, and enter the name in the JMS application connection properties. Configure the recovery queue or recovery topic with the same JNDI and JMS application connection properties as the message queue. The session fails if the connection properties are different.

Configuring JMS Header Fields

You can configure a session to write values to message header fields when the mapping does not pass data to the header fields in the JMS target definition. If the mapping passes values to the header fields in the target and the session configuration contains values for these fields, the PowerCenter Integration Service overwrites the values in the session properties with the values in the target.

You can configure values for the following header fields in the session properties:

- JMSReplyTo
- JMS Time Limit
- JMSDeliveryMode
- JMSPriority
- JMSType

JMS Transacted Mode

You can configure JMS to run in Transacted or Non-Transacted mode.

When you configure JMS to run in Transacted mode, the PowerCenter Integration Service receives and processes a group of JMS messages as a transaction. The PowerCenter Integration Service acknowledges all the messages by committing the transaction. If a transaction rolls back, the JMS provider delivers all the messages to the PowerCenter Integration Service again.

When you configure MS to run in Non-Transacted mode, the PowerCenter Integration Service uses the CLIENT_ACKNOWLEDGE method to acknowledge messages. When the PowerCenter Integration Service acknowledges one message in the JMS session, then all the messages are acknowledged by default. PowerCenter JMS sessions run in Non-Transacted mode by default.

Configure Transacted mode for JMS sessions on the Mapping tab for a JMS session. Click SQ_JMSReader in the left panel. Updated JMS sources have Transacted mode disabled by default.

Note: Data loss may occur if you enable Transacted mode for a SONIC JMS source in a multi-partitioned session.

Configuring Transactional Consistency for JMS Targets

The PowerCenter Integration Service can maintain transactional consistency for sessions with JMS sources and targets. With transactional consistency, the PowerCenter Integration Service commits documents to JMS targets in transaction groups. If the session aborts or fails during a transaction, the PowerCenter Integration Service rolls back all documents in the group from the targets.

Use the following guidelines when you configure transactional consistency:

- Select an application connection with the same connection properties for each JMS target in the pipeline. This ensures that all JMS targets in a single pipeline belong to the same target connection group.
- Configure source-based commits for the session.
- Set a commit interval to define the commit point for a transaction group.

Pipeline Partitioning

If the JMS provider supports partitioning, you can increase the number of partitions in a pipeline to improve session performance. Increasing the number of partitions allows the PowerCenter Integration Service to create multiple connections to sources and targets and process partitions of sources and targets concurrently.

The following table describes the partition types for partition points in JMS mappings:

Partition Point	Partition Type
Application Source Qualifier	Pass-through
JMS target	Pass-through

Note: The session fails when you configure multiple partitions and the provider is WebLogic Server.

Use the following rules and guidelines to process JMS messages in multiple partitions:

- Configure the Queue Reader Mode session property to control how the PowerCenter Integration Service extracts data from a JMS message queue and passes it to the source qualifier. Choose one of the following options to improve performance:
 - **Message Consumer.** Default. The JMS reader runs in multiple partitions.
 - **Connection Consumer.** The JMS reader runs in one partition.
- When you configure the Recovery Cache Folder attribute in the session properties, enter a cache folder on a different device for each reader partition in the pipeline.

Note: To extract data from an ActiveMQ JMS queue, choose **Message Consumer** as the default option.

Streaming JMS Messages to an XML Parser Transformation

If you configure an XML Parser transformation to parse XML from a JMS TextMessage, you can configure the PowerCenter Integration Service to stream the XML between the JMS Source Qualifier and the XML Parser

transformation. When the PowerCenter Integration Service streams XML data, it splits XML data from the JMS Source Qualifier into multiple segments. You can configure a smaller input port in the XML Parser transformation and reduce the amount of memory that the session requires to process large XML documents. The XML Parser transformation can parse messages that are larger than 100 MB.

When you enable XML streaming, the XML Parser transformation receives data in segments that are less than or equal to the port size. When the XML document is larger than the port size, the PowerCenter Integration Service passes more than one row to the XML Parser transformation. Each XML row has a row type of streaming. The last row has a row type of insert.

The XML Parser transformation input port precision must be equal to or greater than the JMS Source Qualifier output port precision. When some of the messages are large, but most of the messages are small, set the XML Parser transformation input port size equal to the size of the smaller messages for best performance.

If you enable XML streaming for the JMS source, you must also enable it in the XML Parser transformation.

You can stream a JMS messages if the message structure is TextMessage. You cannot stream the message if the structure is MapMessage or BytesMessage.

To enable XML streaming:

1. Edit the JMS session in the Workflow Manager.
2. Click the Mapping tab.
3. Click the JMS Source Qualifier in the Navigation panel.
4. Enable XML Streaming in the Source Qualifier session properties.
5. Click the XML Parser transformation in the left panel.
6. Enable XML Input Streaming in the XML Parser transformation session properties.

Configuring a JMS Session

When you configure a session with a JMS source or target, configure the following session properties.

Properties Tab - General Options

In the General Options section of the Properties tab, configure the following session properties:

1. Select Source as the commit type to extract data in real time and maintain transactional consistency for JMS targets.
2. Optionally, edit the commit interval.
3. Select a recovery strategy.

To enable message recovery, select Resume from Last Checkpoint.

If you enable message recovery, you can configure a value for the recovery cache folder from the Properties settings of the Sources node section of the Mapping tab. Or, use the default cache folder \$PMCCacheDir\.

Config Object Tab - Error Handling

In the Error Handling section of the Config Object tab, optionally select a value for Stop on Errors. When you select a value for Stop on Errors, the PowerCenter Integration Service fails the session when it reaches the number of specified errors for the partition.

Mapping Tab - Sources Node

In the Sources Node section of the Mapping tab, configure the following session properties:

1. In the Connection settings, select a JNDI application connection for Application Source Qualifiers connected to JMS source definitions, and then select a JMS application connection value for Application Source Qualifiers connected to JMS source definitions.
2. In the Properties settings, optionally enter a message filter to filter messages from the JMS source.
3. Optionally, edit the values for the Idle Time, Message Count, and Reader Time Limit terminating conditions.

The Workflow Manager assigns the following default values to the terminating conditions:

Terminating Condition	Description
Idle Time	JMS can remain idle for an infinite period of time before the session ends. Default is -1, indicating an infinite period of time.
Message Count	PowerCenter Integration Service can read an infinite number of messages before the session ends. Default is -1, indicating an infinite number of messages.
Reader Time Limit	PowerCenter Integration Service can read source messages from JMS for an infinite period of time. Default is 0, indicating an infinite period of time.

4. Optionally, configure real-time flush latency.

By default, the PowerCenter Integration Service does not run the session in real time. Default value is 0.

5. Enter the following information to determine how the PowerCenter Integration Service reads durable subscription messages:

Attribute Name	Description
JMS Subscription Mode	Specifies the subscription mode. Select Durable for the PowerCenter Integration Service to read durable subscription messages. Otherwise, select Non-Durable. Default is Non-Durable.
Durable Subscription Name	Subscription name required for durable subscriptions. Default is None. Optionally, you can use the <code>\$ParamName</code> session parameter.
Unsubscribe at End	Specifies whether the PowerCenter Integration Service retains a durable subscription after a session completes successfully. Select Yes to delete the durable subscription. Select No to retain the durable subscription. Default is Yes.

6. If you enabled message recovery, you can configure a value for the recovery cache folder. Or, use the default cache folder `$PMCacheDir\`.
7. If you configured an XML Parser transformation to parse a JMS XML message, you can enable XML streaming. If you enable XML streaming, you must also enable the XML input streaming property for the XML Parser.

Mapping Tab - Targets Node

In the Targets Node section of the Mapping tab, configure the following session properties:

1. In the Connection settings, select a JNDI application connection for each JMS target definition.

Tip: If you configure the session for transactional consistency, verify that all JMS target definitions in a single pipeline belong to the same target connection group.

2. Select a JMS application connection value for each JMS target definition.
3. Enter the following properties:

Properties Settings	Description
IsDestinationName Dynamic	Determines if the JMS target has a static or dynamic destination name. <ul style="list-style-type: none">- Enabled. The JMS target receives the destination name in the JMSDestination port at run time. The JMS target receives the destination name with the JMS header data.- Disabled. The PowerCenter Integration Service creates the JMS target with the destination name from the JMS application connection. The PowerCenter Integration Service sends all messages to that destination. When you disable this property, do not project the JMSDestination name port in the JMS target. Default is enabled.
Send Interval	Minimum amount of time in milliseconds the PowerCenter Integration Service waits between each message it writes to the target. For example, if you set the send interval to 2, the PowerCenter Integration Service waits at least two milliseconds before writing messages to the target. The targets receive the messages after the PowerCenter Integration Service issues a commit. If you stop the session, the PowerCenter Integration Service waits until it writes the message it is processing to the target before stopping the session. Therefore, if you set a high value for Send Interval, the session might take longer to stop. Default is 0.
JMS Reply To	Name of the topic or queue. If you enter an invalid value, the session fails.
JMS Time Limit	Time period, in seconds, that messages are valid after the PowerCenter Integration Service writes the messages to the target. During this time period, JMS retains messages in memory. When the time limit expires, it deletes the target messages. When you enter a time limit value, the PowerCenter Integration Service writes the value to the JMS Time Limit field in the JMS target messages. Default is 0. If you enter 0, the PowerCenter Integration Service retains target messages in memory for an infinite period of time.
JMS Delivery Mode	Select Persistent or Non-Persistent. If you select Persistent, the PowerCenter Integration Service guarantees the delivery of JMS messages to the targets. If you select Non-Persistent, the PowerCenter Integration Service attempts to deliver each JMS message to the targets, but does not guarantee delivery.
JMS Priority	Enter a value from 0 to 9 for the message priority. The higher the value, the greater priority the PowerCenter Integration Service gives target messages. Default is 4.
JMS Type	Enter a description of the message.

The mapping overwrites the value in the session properties if it passes a value for any of the following fields in the target:

- JMS Reply To
- JMS Time Limit

- JMS Delivery Mode
- JMS Priority
- JMS Type

Scheduling Workflows

You can schedule a workflow to run continuously, run at a given time or interval, or you can manually start a workflow. The PowerCenter Integration Service runs scheduled workflows through the duration of the schedule, unless the workflow fails.

To run a continuous workflow, select Run Continuously when you edit the scheduler for the workflow. A continuous workflow starts as soon as the PowerCenter Integration Service initializes. When the workflow stops, it restarts immediately.

JMS Resiliency

The PowerCenter Integration Service is resilient to JMS connection failures when recovery is enabled. The PowerCenter Integration Service is resilient to connection failures when the PowerCenter Integration Service is reading a JMS message queue or JMS topic. It is also resilient to connection failures when the session runs over multiple partitions.

JMS Message Queue Resiliency

When a connection error occurs while the PowerCenter Integration Service is acknowledging a checkpoint in the source, the PowerCenter Integration Service fetches and acknowledges the messages in the current checkpoint. The session fails if the PowerCenter Integration Service does not find any message to acknowledge.

If the connection error occurs while the PowerCenter Integration Service is fetching messages from the source, the PowerCenter Integration Service issues a real-time commit to the target before it fetches the messages in the current checkpoint.

JMS Topic Resiliency

When a connection error occurs while the PowerCenter Integration Service is acknowledging a checkpoint in the source, the PowerCenter Integration Service fetches all messages from the source in the current checkpoint. It fetches the messages in one partition. If the PowerCenter Integration Service detects new messages that it did not process before the connection failed, it processes messages in two transactions. The first transaction includes all the messages which were processed just before the connection failed. The second transaction includes new messages and messages that were not processed.

When the connection error occurs while the PowerCenter Integration Service is fetching messages from the source, the PowerCenter Integration Service issues a real-time commit to the target before it fetches all the messages from the source in the current checkpoint.

Resiliency During Recovery

When a connection error occurs while the PowerCenter Integration Service is recovering a failed session, the PowerCenter Integration Service can continue the session depending on when the connection error occurs. The PowerCenter Integration Service can recover the session if the connection error occurs after the session has resumed from recovery. The session fails if the connection error occurs while the PowerCenter Integration Service is recovering the session and the session has not resumed.

JMS Connection Error Codes File

You can configure error messages for JMS resilience based on messages received from JMS. WebSphere MQ has error message IDs that correspond to connection error messages. Sonic MQ and TIBCO do not provide error message IDs that identify connection failures to PowerCenter.

The PowerCenter installation includes a connection error code file named `pmjmsconnerr.properties`. The file contains name-value pairs of JMS error message IDs and the messages that identify connection failures for PowerCenter. The name is a name of the error message or a message ID that you can define to identify a connection error. The value is the error message ID or error message that is returned from JMS. The `pmjmsconnerr.properties` file contains error IDs for WebSphere MQ, TIBCO, and Sonic MQ JMS providers.

The `pmjmsconnerr.properties` file is in the following location:

```
<PowerCenter Installation Directory>\server\bin\pmjmsconnerr.properties
```

You can change the name of the connection error code file in the JMS application connection. Configure the Retry Connection Error Code File Name. Default is `pmjmsconnerr.properties`.

When the PowerCenter Integration Service receives a JMS exception error, it searches for the error in the properties file. It searches on the error value. If the error is in the file, the PowerCenter Integration Service attempts to reconnect to JMS.

You can update the properties file to add new messages.

Connection Retry Period

You can configure the connection retry period in the JMS connection attributes. The connection retry period is the number of seconds the PowerCenter Integration Service attempts to reconnect to JMS if the connection fails. Default value is 0.

Troubleshooting JMS Workflows

I ran a session to write JMS messages to a WebSphere MQ JMS provider. The session completed successfully, but the target messages do not contain property fields.

When you configured the connection factory in JNDI for the WebSphere MQ JMS provider, you set the `TARGETCLIENT` value to `MQ`. When WebSphere MQ JMS receives target messages, it translates them to WebSphere MQ message format. Messages in WebSphere MQ format contain header fields and a body field. They do not contain property fields.

To receive target messages with property fields, set the value for `TARGETCLIENT` to `JMS` in your JNDI configuration, and run the session again. For information about the format of property fields in WebSphere MQ JMS messages, see the WebSphere MQ JMS documentation.

I ran a session to write data from an WebSphere MQ JMS source to a JMS target. The session failed because the PowerCenter Integration Service rejected data from the following header fields: JMSDestination, JMSReplyTo, and JMSMessageID.

When you have a mapping that passes JMS message header data from a JMS source to a JMS target and the source is WebSphere MQ JMS, WebSphere MQ JMS prepends data to the data in the JMSDestination, JMSReplyTo, and JMSMessageID header fields.

For example, the PowerCenter Integration Service writes data from the JMSDestination field in the source to the JMSDestination field in the target. The value for the JMSDestination field in the source is JMS_1. When the PowerCenter Integration Service writes the data to the target, the value is queue://QM_pcserv/JMS_1. The PowerCenter Integration Service rejects the message.

Use an Expression transformation in the mapping to extract the correct value and pass it to the target.

I ran a session to read data from a JMS source and write the data to a non-JMS target. The message ID value for the JMSMessageID field begins with "ID:" instead of only containing the message ID.

When you have a mapping that passes data from the JMSMessageID field in a JMS source to a non-JMS target, JMS includes the string "ID:" in the JMSMessageID field. To remove the "ID:" characters before the message ID, you must use an Expression transformation in the mapping.

I ran a session to write messages to an WebSphere MQ JMS target. The commit interval for the session was high. The session failed due to a JMS exception. It failed to send messages to the WebSphere MQ queue.

When you run a session with a high commit interval to write message to an WebSphere MQ JMS target, the PowerCenter Integration Service writes messages to WebSphere MQ JMS. WebSphere MQ JMS writes the messages to a cache file. Because the commit interval is high, the cache file might run out of memory. As a result, the session fails. To prevent the session from failing, lower the commit interval. Or, increase the size of the WebSphere MQ cache file.

I tried to run several sessions simultaneously to read messages from multiple JMS sources and write the messages to a single target. The sessions used the Idle Time terminating condition and completed when the Idle Time condition was met. When the sessions completed, the session log reported that the PowerCenter Integration Service only read a small number of messages from the source during some sessions.

When you run multiple sessions simultaneously to read messages from multiple JMS sources and write the messages to a single target, the PowerCenter Integration Service might not read messages from the source for a few moments while it writes data to the target. As a result, the PowerCenter Integration Service meets the Idle Time terminating condition and ends the session. To make sure that each session reads the appropriate number of messages, specify a higher value for the Idle Time terminating condition.

I receive the following error message when I run a JMS session with an WebSphere MQ provider:

```
MAPPING> [JAVA PLUGIN] [ERROR] Unable to load message catalog - mqji
```

This error message appears even if the session completes successfully. To prevent the error message from displaying, include the <WebSphere MQ JMS>\lib directory in the CLASSPATH.

APPENDIX A

JMS Datatype Reference

This appendix includes the following topic:

- [JMS and Transformation Datatypes, 31](#)

JMS and Transformation Datatypes

PowerCenter uses the following datatypes in JMS mappings:

- **JMS native datatypes.** JMS datatypes appear in JMS definitions in a mapping.
- **Transformation datatypes.** Set of datatypes that appear in the transformations. They are internal datatypes based on ANSI SQL-92 generic datatypes, which the PowerCenter Integration Service uses to move data across platforms. They appear in all transformations in a mapping.

When the PowerCenter Integration Service reads source data, it converts the native datatypes to the comparable transformation datatypes before transforming the data. When the PowerCenter Integration Service writes to a target, it converts the transformation datatypes to the comparable native datatypes.

The following table lists the JMS datatypes that PowerCenter supports and the corresponding transformation datatypes:

JMS Datatype	Range and Description	Transformation Datatype	Range and Description
Binary	1 to 104,857,600 bytes. You can pass binary data from a source to a target, but you cannot perform transformations on binary data.	Binary	1 to 104,857,600 bytes
Boolean	Unicode mode: Precision of 2 bytes. ASCII mode: Precision of 1 byte.	String	Precision 1
Byte	Precision of 5 and scale of 0. Integer value.	Small Integer	Precision 5, scale 0
Char	Unicode mode: Precision of 2 bytes. ASCII mode: Precision of 1 byte.	String	Precision 1
Datetime	Jan 1, 1970 to Dec 31, 9999 A.D. Combined date/time value with precision to the millisecond.	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)

JMS Datatype	Range and Description	Transformation Datatype	Range and Description
Double	Precision of 18 digits.	Double	Precision 15
Float	Precision of 7 digits.	Real	Precision 7, scale 0
Integer	Precision of 10 and scale of 0. Integer value.	Integer	Precision 10, scale 0
Long	Precision 1 to 19 digits and scale of 0.	Bigint	Precision 19, scale 0 Integer value.
Short	Precision of 5 and scale of 0. Integer value.	Small Integer	Precision 5, scale 0
String	Unicode mode: (precision + 1) * 2. ASCII mode: precision + 1. 1 to 104,857,600 characters. Fixed-length or varying-length string.	String	1 to 104,857,600 characters

Boolean Datatype

When the PowerCenter Integration Service reads data from a JMS source field of the Boolean datatype, the PowerCenter Integration Service converts the value "True" to "T" and the value "False" to "F."

When the PowerCenter Integration Service writes data to a JMS target field of the Boolean datatype, it converts the value "T" to "True" and the value "F" to "False." If the field contains any other value, the PowerCenter Integration Service rejects the field.

Reading From JMS Sources

Datatypes for fields in JMS source messages must be compatible with datatypes for the corresponding field in the source definition. For example, if a property field called TradeAmount in a JMS message is of the datatype Long, the corresponding field in the source definition must be of the datatype Long or String.

When the source message contains a field that has a datatype that is not compatible with the datatypes for the corresponding field in the source definition, the PowerCenter Integration Service generates an error for that field.

The following table describes datatypes in source definitions and the datatypes they are compatible with in JMS messages:

Datatype in the Source Definition	Datatype in a JMS Message
Binary	Binary
Boolean	Boolean Byte

Datatype in the Source Definition	Datatype in a JMS Message
Byte	Byte Integer Long Short String
Char	Char String
Double	Double String
Float	Double Float String
Integer	Integer Long String
Long	Long String
Short	Integer Long Short String
String	Boolean Byte Double Float Integer Long Short String

APPENDIX B

Glossary of Terms

BytesMessage

A message that is a stream of uninterpreted bytes. It does not include property fields.

connection factory

A component of the JNDI server that contains information to connect to the JMS provider.

destination

A component of the JNDI server that contains information about message sources and destinations.

durable subscription

A subscription that requires the JMS provider to save messages published on a topic to a temporary store until a client receives or deletes them.

Java Naming and Directory Interface (JNDI) server

A server that contains connection information to connect to a JMS provider.

JMS provider

A message-oriented middleware system that sends and receives JMS messages.

MapMessage

A message that contains a set of name/value pairs.

point-to-point

A message domain that requires a JMS provider to store messages in a message queue. Clients read messages from the queue or write messages to the queue.

property

A field in a JMS message that contains additional message header information.

publish-subscribe

A message domain that requires a JMS provider to read and write messages with the message topic as the address.

TextMessage

A message that contains a string object. TextMessages can contain XML message data.

topic

The subject of a JMS message.

INDEX

A

authentication certificates
JMS SSL authentication component [11](#)

B

BEA WebLogic Server
partitioning for JMS sources and targets [24](#)
body fields
JMS messages [10](#)
BytesMessage
JMS message type [9](#), [15](#)

C

client store
JMS SSL authentication component [11](#)
commit interval
transactional consistency for JMS [24](#)
Connection Consumer mode
reading JMS queues [24](#)
continuous workflows
description for JMS [28](#)

D

datatypes
PowerExchange for JMS [31](#)
durable subscription
description for JMS [22](#)
Durable Subscription Name (property)
configuring for a JMS session [26](#)

F

filter
for JMS source messages [21](#)

H

header fields
in a JMS target [16](#)
JMS messages [9](#)
troubleshooting for JMS [29](#)

I

idle time
description for JMS [21](#)

idle time (*continued*)
troubleshooting [29](#)
IsDestinationNameDynamic
JMS session property [27](#)

J

Java Naming and Directory Interface server
description [10](#)
JMS datatypes
Boolean [32](#)
reading from source messages [32](#)
JMS Destination
configuring static or dynamic [27](#)
JMS provider
communication with PowerCenter Integration Service [11](#)
reading queues in partitions [24](#)
registering client libraries [14](#)
Transacted mode [23](#)
using JNDI and JMS application connections [10](#)
JMS source definitions
creating [17](#)
editing [17](#)
overview [15](#)
JMS Subscription Mode (property)
configuring for a JMS session [26](#)
JMS target definitions
creating [17](#)
editing [17](#)
overview [15](#)
JMS targets
maintaining transactional consistency [17](#)
transactional consistency, configuring [24](#)
JMSDeliveryMode
in a JMS target [16](#)
JMSPriority
in a JMS target [16](#)
JVM
loading [20](#)
JVM options
configuring PowerCenter Integration Service for JMS [14](#)

K

keystore
JMS SSL authentication [12](#)

M

MapMessage
JMS message type [9](#), [15](#)
message
JMS message type [9](#), [15](#)

- Message Consumer mode
 - reading JMS queues [24](#)
- message count
 - description for JMS [22](#)
- message filter
 - JMS source messages [21](#)
- message header fields
 - JMS, configuring in the session properties [23](#)
- message recovery
 - description for JMS [23](#)
 - JMS session, configuring [23](#)
 - specifying partitions and a recovery cache folder for JMS [24](#)
- message structure
 - description for JMS [9](#)
- message types
 - description [15](#)
 - JMS [9](#)
- messages
 - JMS components [9](#)

P

- partitions
 - BEA WebLogic Server [24](#)
 - JMS [24](#)
 - specifying partitions and a recovery cache folder for JMS [24](#)
- PowerExchange
 - configuring for JMS [13](#)
- property fields
 - JMS messages [10](#)
 - troubleshooting JMS [29](#)

Q

- Queue Reader Mode
 - reading JMS queues [24](#)

R

- reader time limit
 - description for JMS [22](#)
- real-time flush latency
 - continuous JMS workflows [28](#)
 - source-based commit for a JMS session [25](#)
- real-time sessions
 - description for JMS [22](#)

S

- scheduling workflows
 - JMS [28](#)
- Secure Sockets Layer
 - PKI standard for JMS [11](#)
- sessions
 - configuring using JMS [25](#)

- sessions (*continued*)
 - JMS, overview [20](#)
 - loading JVM [20](#)
- source messages
 - filtering JMS messages [21](#)
- source-based commits
 - transactional consistency for JMS [24](#)
- SSL authentication components
 - authentication certificate for JMS [11](#)
 - client store for JMS [11](#)
 - trust store for JMS [11](#)
- streaming XML
 - JMS messages to XML Parser transformation [24](#)

T

- target connection groups
 - transactional consistency for JMS [24](#)
- terminating conditions
 - description for JMS [21](#)
 - JMS idle time [21](#)
 - JMS message count [22](#)
 - JMS reader time limit [22](#)
 - JMS real-time flush latency [22](#)
- TextMessage
 - JMS message type [9](#), [15](#)
- Transacted mode
 - JMS messaging [23](#)
- transactional consistency
 - configuring for JMS targets [24](#)
- troubleshooting
 - JMS workflows [29](#)
- trust store
 - JMS SSL authentication component [11](#)

U

- Unsubscribe at End (property)
 - configuring for a JMS session [26](#)

W

- WebSphere MQ JMS
 - fields for a WebSphere MQ JMS source or target [17](#)
 - troubleshooting [29](#)
 - writing to the JMSDeliveryMode field [16](#)
 - writing to the JMSPriority field [16](#)
- workflows
 - configuring a JMS session [25](#)
 - scheduling JMS [28](#)
 - troubleshooting JMS [29](#)