



Informatica® PowerExchange for MongoDB
JDBC
10.4.1

User Guide for PowerCenter

© Copyright Informatica LLC 2020, 2021

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.sl4j.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/licence.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.sl4j.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.schneier.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, report them to us at infa_documentation@informatica.com.

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: 2021-07-30

Table of Contents

Preface	6
Informatica Resources.	6
Informatica Network.	6
Informatica Knowledge Base.	6
Informatica Documentation.	6
Informatica Product Availability Matrices.	7
Informatica Velocity.	7
Informatica Marketplace.	7
Informatica Global Customer Support.	7
Chapter 1: Introduction to PowerExchange for MongoDB JDBC	8
PowerExchange for MongoDB JDBC Overview.	8
Introduction to MongoDB JDBC.	9
PowerExchange for MongoDB Implementation.	9
Chapter 2: PowerExchange for MongoDB JDBC Configuration	11
PowerExchange for MongoDB JDBC Configuration Overview.	11
Installation and Configuration Overview.	11
Prerequisites.	11
Registering the Plug-in.	12
Registering the Plug-in from the Administrator Tool.	12
Registering the Plug-in from the Command Line Interface.	13
Chapter 3: Schema Definition	14
Schema Definition Overview.	14
Virtual Tables.	14
Virtual Tables Example.	15
Chapter 4: MongoDB JDBC Sources and Targets	17
MongoDB JDBC Sources and Targets Overview.	17
Import MongoDB JDBC Source and Target Definitions.	17
Chapter 5: MongoDB JDBC Mappings	21
MongoDB JDBC Mappings Overview.	21
Source Filter.	21
Source Join.	23
Writing to an empty collection.	25
MongoDB JDBC Mapping Example.	25

Chapter 6: MongoDB JDBC Sessions.....	27
MongoDB JDBC Sessions Overview.	27
PowerExchange for MongoDB JDBC Connections.	28
Configuring a MongoDB JDBC Connection.	29
Configure MongoDB JDBC Source Session Properties.	29
Configure a Filter Override.	30
Configure MongoDB JDBC Target Session Properties.	30
Parameterization.	31
Lookup.	31
Update Strategy Transformation.	31
Partitioning.	31
Configure Pass-Through Partitioning for a MongoDB JDBC Session.	32
 Chapter 7: Data Type Reference.....	 34
Data Type Reference Overview.	34
MongoDB and Transformation Data Types.	34
 Index.....	 36

Preface

Use the *Informatica® PowerExchange® for MongoDB User Guide for PowerCenter®* to learn how to read from or write to MongoDB by using PowerCenter client. Learn to create a connection, develop mappings, and run sessions in an Informatica domain.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit <https://network.informatica.com>.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and 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 find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit <https://search.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

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit <https://docs.informatica.com>.

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

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica 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 and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find 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 extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at <https://marketplace.informatica.com>.

Informatica Global Customer Support

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

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

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

To find online support resources on the Informatica Network, visit <https://network.informatica.com> and select the eSupport option.

CHAPTER 1

Introduction to PowerExchange for MongoDB JDBC

This chapter includes the following topics:

- [PowerExchange for MongoDB JDBC Overview, 8](#)
- [Introduction to MongoDB JDBC, 9](#)
- [PowerExchange for MongoDB Implementation, 9](#)

PowerExchange for MongoDB JDBC Overview

PowerExchange for MongoDB JDBC provides connectivity between Informatica and MongoDB. Use PowerExchange for MongoDB JDBC to extract and load MongoDB documents through the PowerCenter Integration Service.

You can use PowerExchange for MongoDB JDBC to integrate and migrate data from diverse data sources that are incompatible with MongoDB architecture.

You can use PowerExchange for MongoDB JDBC for the following PowerCenter Integration scenarios:

- Perform insert, update or delete operations on the MongoDB collections. You can aggregate data from MongoDB and other source systems, transform the data, and write the data to MongoDB.
- Migrate data from a relational database or other data sources to MongoDB. For example, you want to migrate data from a relational database to MongoDB. You can write data from multiple relational database tables with different schemas to the same MongoDB collection. A MongoDB collection contains the data in a MongoDB database.
- Move data between operational data stores to synchronize data. For example, an online marketplace uses a relational database as the operational data store. You want to use MongoDB instead of the relational database. However, you want to maintain the relational database along with MongoDB for a period of time. You can use PowerExchange for MongoDB to synchronize data between the relational data store and the MongoDB data store.
- Migrate data from MongoDB to a data warehouse for reporting. For example, your organization uses a business intelligence tool that does not support MongoDB. You must migrate the data from MongoDB to a data warehouse so that the business intelligence tool can use the data to generate reports.

Introduction to MongoDB JDBC

MongoDB is an open source, document based, NoSQL database that maintains dynamic schema. You can maintain more than one database on a MongoDB server.

A MongoDB database contains a set of collections. A collection is a set of documents and is similar to a table in a relational database. MongoDB stores records as documents that are similar to rows in a relational database. A document contains fields that are similar to columns in a relational database. A document can have a dynamic schema. A document in a collection does not need to have the same set of fields or structure as another document in the same collection. A document can also contain nested documents.

The following schema provides a sample MongoDB document from the collection called Product:

```
{
  sku: "111445GB3",
  title: "CM Phone",
  description: "The best in the world.",

  manufacture_details: {
    model_number: "CMP",
    release_date: new ISODate("2011-07-17T22:14:15.656Z")
  },

  shipping_details: {
    weight: 350,
    width: 10,
    height: 10,
    depth: 1
  },

  quantity: 99,

  pricing: [
    {region: "North America",
      cost_price: 1000,
      sale_price: 1200},
    {region: "Europe",
      cost_price: 1200,
      sale_price: 1500}
  ]
}
```

In the example, sku, title, description, quantity, manufacture_details, shipping_details, and pricing are fields. The fields manufacture_details and shipping_details are nested document type fields and pricing is an array type field.

PowerExchange for MongoDB Implementation

To extract and load MongoDB data, create MongoDB source and target definitions in the Designer. You can add a source or target definition to a session and run the session to process the data.

PowerExchange for MongoDB includes the Informatica MongoDB JDBC driver that connects to the MongoDB server. You can create an JDBC connection to extract data from or load data to a MongoDB database. You can also configure the replica sets for the MongoDB server so that the PowerCenter Integration Service can access the secondary servers if the primary server is not available.

The Designer uses the schema of a collection, or you can define the schema for the collection before you import a source or target definition. The Designer flattens the schema if there is any hierarchical element in the collection and retains the original schema of the collection when you import it.

The Designer imports a document based on the schema that you set for the collection. If a document contains hierarchical elements like arrays or nested documents, the Designer imports them as columns at the same level as other columns.

For example, you need to import the collection `product_details` with the following schema:

```
{
  sku: "sku_name",
  title: "product_name",
  description: "description",

  manufacture_details: {
    model_number: "model_number",
    release_date: new ISODate("date")
  },

  shipping_details: {
    weight: <value>,
    width: <value>,
    height: <value>,
    depth: <value>
  },

  quantity: <value>,

  pricing: [
    {region: "North America",
      cost_price: 1000,
      sale_price: 1200},
    {region: "Europe",
      cost_price: 1200,
      sale_price: 1500}
  ]
}
```

The Designer imports the collection schema into a tabular format. You can identify arrays and nested documents with the naming convention of the column. The naming convention of a nested document is `<top level element name>.<nested document name>.<nested document element name>`. The naming convention of an array is `<array name>.<element number>`.

The following figure shows the source definition when you import the collection into the Designer:

K	Name	Datatype	Length/Precision
	_id	DOUBLE	15
	MongoDB_PC_...	INTEGER	10
	quantity	DOUBLE	15
	extended_price	DOUBLE	15
	discount	DOUBLE	15
	tax	DOUBLE	15
	returnflag	VARCHAR	255
	lnstatus	VARCHAR	255
	shipdate	TYPE_TIMESTAMP	26
	commitdate	TYPE_TIMESTAMP	26
	receiptdate	TYPE_TIMESTAMP	26
	shipinstruct	VARCHAR	255
	shipmode	VARCHAR	255
	comment	VARCHAR	255
	partsupp	DOUBLE	15

When you run a session, the PowerCenter Integration Service uses the MongoDB JDBC data source name in the machine that runs the PowerCenter Integration Service to extract data from or load data to a MongoDB database.

CHAPTER 2

PowerExchange for MongoDB JDBC Configuration

This chapter includes the following topics:

- [PowerExchange for MongoDB JDBC Configuration Overview, 11](#)
- [Installation and Configuration Overview, 11](#)
- [Prerequisites, 11](#)
- [Registering the Plug-in, 12](#)

PowerExchange for MongoDB JDBC Configuration Overview

You can use PowerExchange for MongoDB JDBC on Windows, Linux, or AIX. You must configure PowerExchange for MongoDB JDBC before you can extract data from or load data to MongoDB database.

Installation and Configuration Overview

PowerExchange for MongoDB JDBC installs with PowerCenter. After you install or upgrade Informatica Services, you must register the PowerExchange for MongoDB JDBC plug-in with the PowerCenter repository.

Prerequisites

Before you use PowerExchange for MongoDB JDBC, perform the following tasks:

- Install or upgrade PowerCenter.
- Verify that you have read and write permissions on the following directories on each machine that runs the PowerCenter Integration Service and PowerCenter Repository Service:

- <Informatica installation directory>\server\connectors\cci\plugins

```
- <Informatica installation directory>\server\bin\Plugin  
- <Informatica installation directory>\connectors\thirdparty
```

- Verify that you have read and write permissions on the following directories on each machine where you installed the PowerCenter Client:

```
- <Informatica installation directory>\clients\PowerCenterClient  
- <Informatica installation directory>\clients\PowerCenterClient\bin\Help\<language>
```

For more information about product requirements and supported platforms, see the Product Availability Matrix on Informatica Network:

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

- To invoke the Schema Editor from the JDBC driver in PowerCenter, follow the instructions from the following Knowledge Base article:
https://knowledge.informatica.com/s/articlepreview?c__number=HOW-TO-Invoke-schema-editor-from-MongoDB-JDBC-driver&language=en_US&type=external

Registering the Plug-in

After you complete the installation, register the PowerExchange for MongoDB JDBC plug-in with the repository.

To register the plug-in, the repository must be running in exclusive mode. Use the Administrator tool or the pmrep RegisterPlugin command line program to register the plug-in. If you do not have the correct privileges to register the plug-in, contact the user who manages the PowerCenter Repository Service.

The plug-in file is an .xml file that defines the functionality of the adapter. When you install the server component, the installer copies the plug-in file to the following directory:

```
<Informatica installation directory>/server/bin/Plugin
```

The name of the plug-in file for PowerExchange for MongoDB JDBC is MongoDB_Plugin.xml.

Registering the Plug-in from the Administrator Tool

Register a repository plug-in to add its functionality to the repository.

1. Run the PowerCenter Repository Service in exclusive mode.
2. In the **Navigator**, select the PowerCenter Repository Service to which you want to add the plug-in.
3. In the **Contents** panel, click the **Plug-ins** view.
4. In the **Actions** menu of the **Domain** tab, select **Register Plug-in**.
5. On the **Register Plugin** page, click the **Browse** button to locate the plug-in file.
6. Enter your user name and password.
7. Click **OK**.

The PowerCenter Repository Service registers the plug-in with the repository. The results of the registration operation appear in the activity log.

8. Run the PowerCenter Repository Service in normal mode.

Registering the Plug-in from the Command Line Interface

You can use the `pmrep RegisterPlugin` command to register the plug-in from the command line interface.

1. Run the PowerCenter Repository Service in exclusive mode.
2. Run the `pmrep Connect` command to connect to the Repository Service with a user account that has the Administrator Repository privilege.

The `RegisterPlugin` command uses the following syntax:

```
pmrep connect -r <repository name> -d <domain_name> -n <domain user name> -x  
<domain_password>
```

3. Find `<adaptername>.xml` in the following directory:

```
<Informatica installation directory>\server\bin\Plugin
```

4. Run the `pmrep RegisterPlugin` command to update the repository.

The `RegisterPlugin` command uses the following syntax:

```
pmrep registerplugin -i <Informatica installation directory>\server\bin\Plugin  
\MongoDB_Plugin.xml -e
```

CHAPTER 3

Schema Definition

This chapter includes the following topics:

- [Schema Definition Overview, 14](#)
- [Virtual Tables, 14](#)

Schema Definition Overview

You can define the schema for a MongoDB collection that you want to import as a source or target definition in the Designer. You can define the schema for multiple collections with the same JDBC data source name.

A collection in MongoDB might contain several fields that you do not want to import. When you define the schema you can limit metadata that you import. The driver dynamically detects the collection schema of a MongoDB database. It flattens the MongoDB schema and displays the keys in the a tabular format with each key as a column in the Schema Editor.

You can export the collection to an external schema definition file and edit the schema definition in the Schema Editor. After you modify the collection properties and column metadata, you can save the modifications in the schema definition file. The driver does not modify the schema of the actual MongoDB collection. You can choose to store the modifications in the MongoDB database or as a file.

If you enable virtual table detection in the Informatica MongoDB JDBC driver, the driver creates virtual tables in the schema if the collection contains arrays. You can import the virtual table as a source or target definition in the Designer.

Virtual Tables

The Simba MongoDB JDBC driver creates virtual tables if the column family contains collections such as arrays and objects.

Virtual tables depict the renormalized view of a MongoDB collection. You can use virtual tables as sources and targets in mappings.

The Simba MongoDB JDBC driver creates a virtual table for each collection in the column family. The virtual table for a collection uses the following naming convention by default: `<original column family name>_<original collection name>`

Each virtual table has a foreign key column that references back to the primary key column in the original collection. The key column uses the following naming convention by default: `<original collection name>.<primary key column name>`

The virtual table has an index column that shows the position of the data within the original array. The index column uses the following naming convention by default: `<original column name>_index`

Other columns in the virtual table represent the elements in the array and are named after the array element. If the array is of scalar type, the data column uses the following naming convention by default: `<original column name>_value`

When you use a virtual table in a mapping, you can perform read, insert, and update operations on a virtual table.

Note: You cannot perform upsert or delete operation on a virtual table.

Virtual Tables Example

The collection CustomerTable contains arrays. You want to create virtual tables from the arrays and import the virtual tables as data objects in Informatica Cloud.

The following table shows the schema of CustomerTable collection:

id	Customer Name	Invoices	Service Level	Contacts	Ratings
1111	John	[[invoice_id=123,item=toaster,price=456,discout=0.2], {invoice_id=124,item=oven,price=12345, discount=0.3}]	Silver	[[type=primary,name="John Johnson"], {type=invoicing,name="Jane Johnson"}]	[7,8]
2222	Jane	[[invoice_id=125,item=blender,price=7456,discout=0.5],	Gold	[[type=primary,name="Jane Johnson"]	[5,6]

If you enable virtual table detection, the driver creates the following virtual tables:

CustomerTable

The main table uses the same name as the original table that it represents.

The following table shows the schema of CustomerTable virtual table:

id	Customer Name	Number of Invoices	Service Level	Number of Contacts	Number of Ratings
1111	John	2	Silver	2	2
2222	Jane	1	Gold	1	2

CustomerTable_Invoices

The following table shows the schema of CustomerTable_Invoices virtual table:

CustomerTable.id	Invoices_index	invoice_id	item	price	discount
1111	1	123	toaster	456	0.2
1111	2	124	oven	12345	0.3
2222	1	125	blender	7456	0.5

CustomerTable_Contacts

The following table shows the schema of CustomerTable_Contacts virtual table:

CustomerTable.id	Contacts_index	type	name
1111	1	primary	John Johnson
1111	2	invoicing	Jane Johnson
2222	1	primary	Jane Johnson

CustomerTable_Ratings

The following table shows the schema of CustomerTable_Ratings virtual table:

CustomerTable.id	Ratings_index	Ratings_value
1111	1	7
1111	2	8
2222	1	5
2222	2	6

Note: You cannot use the Designer to preview virtual tables. You cannot use a DD_DELETE strategy in an Update Strategy transformation to delete rows from a virtual table.

CHAPTER 4

MongoDB JDBC Sources and Targets

This chapter includes the following topics:

- [MongoDB JDBC Sources and Targets Overview, 17](#)
- [Import MongoDB JDBC Source and Target Definitions, 17](#)

MongoDB JDBC Sources and Targets Overview

You can create a mapping with a MongoDB JDBC source to extract data from MongoDB. You can create a mapping with any source and a MongoDB JDBC target to load data to MongoDB. You can use tables as MongoDB sources.

When the PowerCenter Integration Service extracts data from the source or loads data to the target, it converts the data based on the data types associated with the source or the target.

Import MongoDB JDBC Source and Target Definitions

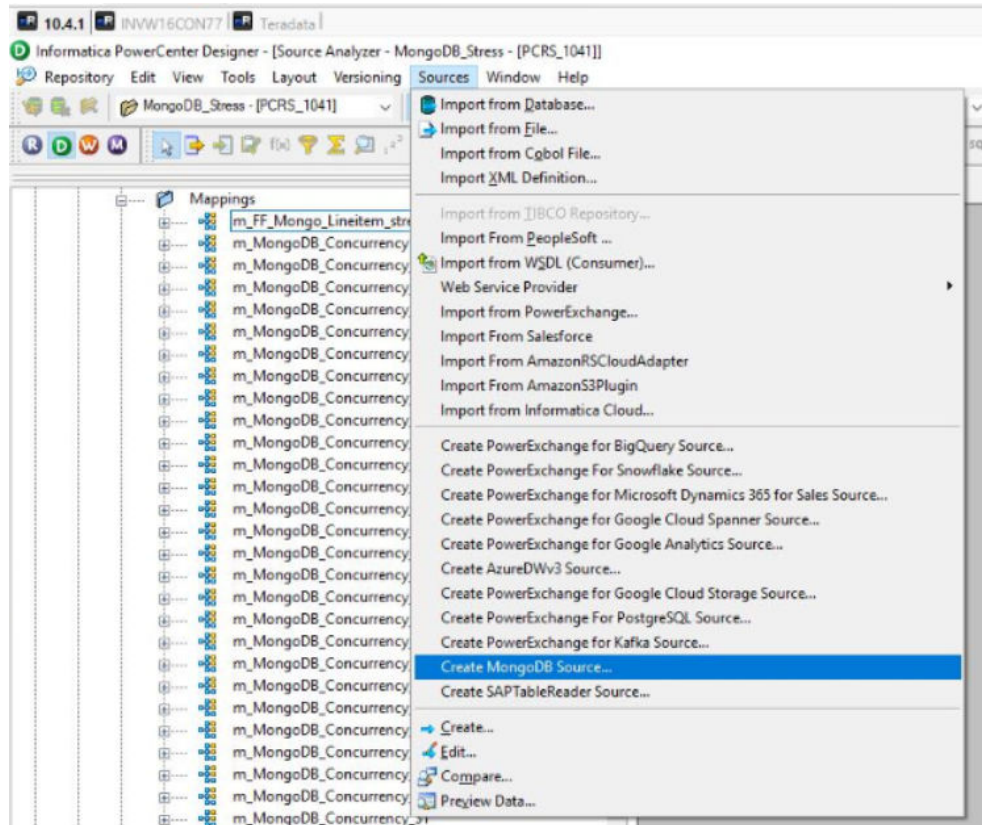
Use the **Create PowerExchange For MongoDB Source** or **Create PowerExchange For MongoDB Target** wizard to import MongoDB source and target definitions into the PowerCenter repository.

You must import MongoDB source and target objects before you create a mapping.

1. Start PowerCenter Designer and connect to a PowerCenter repository configured with a MongoDB instance.
2. Open a source or target folder.

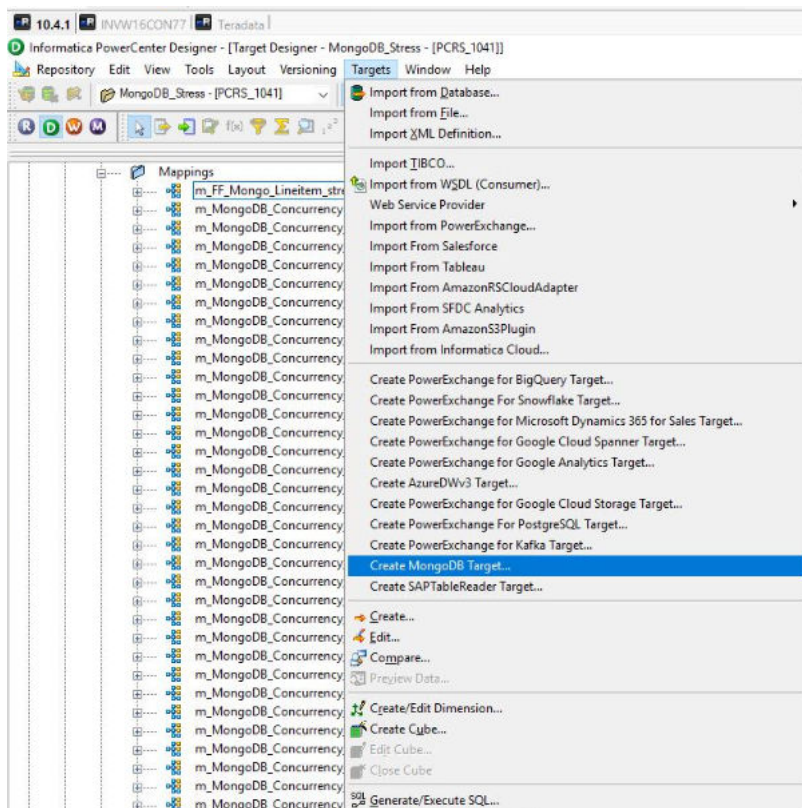
3. Select **Source Analyzer** or **Target Designer**.

- In the Source Analyzer, click **Sources > Create PowerExchange For MongoDB Source**. The following image shows the option to create a MongoDB source:



The **MongoDB Connection Details** dialog box appears.

- In the Target Analyzer, click **Targets > Create PowerExchange For MongoDB Target**. The following image shows the option to create a MongoDB target:



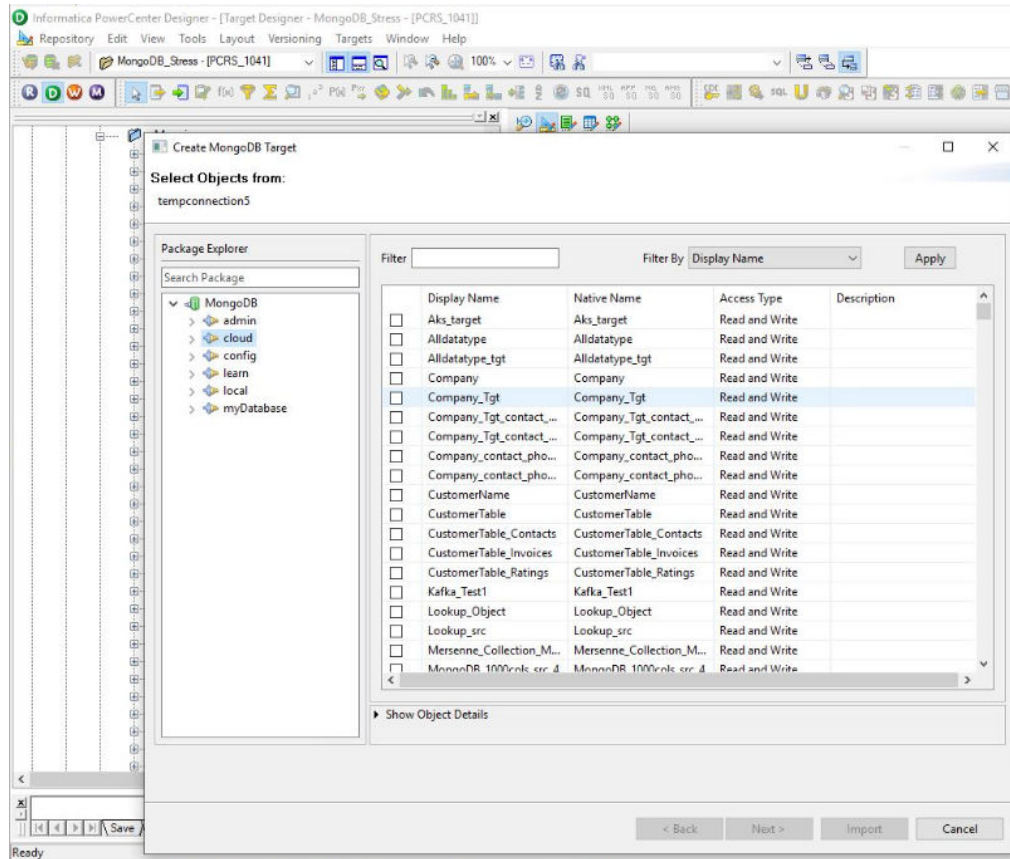
The **MongoDB Connection Details** dialog box appears.

4. Configure the MongoDB connection parameters.
5. Click **Test** to test the connection.
6. Click **Finish** to add the connection.
7. Click **Next**.

The **Select Objects from** tab appears.

8. Select the database and schema in **Package Explorer**.

The following image shows the list of tables that appear:



9. Select the table that you want to import, and then click **Import**.

To view the table metadata, select the table, and double-click the table name.

Note: You can import table collections having upto 10,000 columns and if you are unable to import 1000 columns in the table, you can increase the virtual memory size on the Windows machine that hosts the PowerCenter as mentioned in <https://kb.informatica.com/solution/23/Pages/3/152632.aspx>

CHAPTER 5

MongoDB JDBC Mappings

This chapter includes the following topics:

- [MongoDB JDBC Mappings Overview, 21](#)
- [MongoDB JDBC Mapping Example, 25](#)

MongoDB JDBC Mappings Overview

After you import a MongoDB source or target definition into the PowerCenter repository, you can create a mapping to extract data from a MongoDB source or load data to a MongoDB target.

You can extract data from one or more MongoDB sources, and load data to one or more MongoDB targets. You can join multiple MongoDB tables when you specify a join condition.

You can enter a filter condition to reduce the number of source rows the PowerCenter Integration Service returns from MongoDB sources. You can enter a single filter condition or a series of conditions.

Source Filter

You can enter a filter condition to reduce the number of source rows the PowerCenter Integration Service returns from MongoDB sources. You can enter a single filter condition or a series of conditions.

Use the source filter in the **Application Source Qualifier** to retrieve rows from an entity that meet a condition.

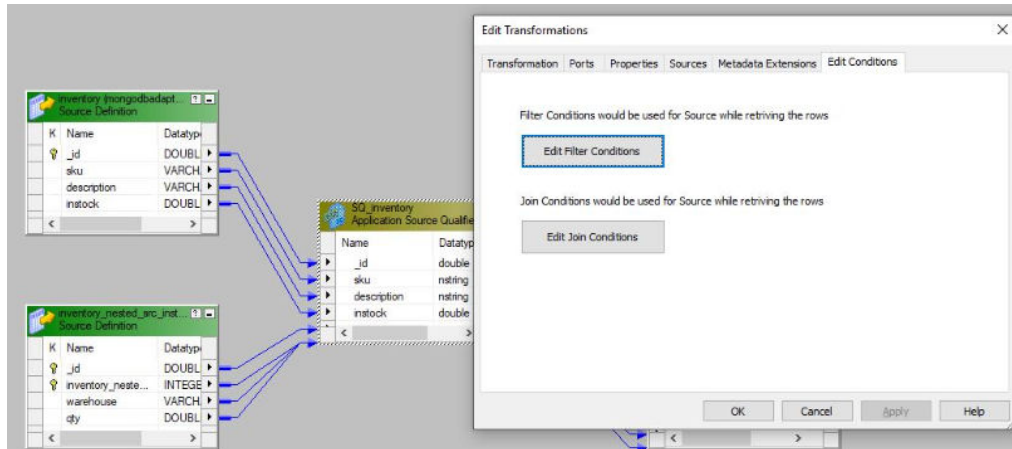
You can provide a source filter to improve the performance when you read from MongoDB.

Configuring a Source Filter

Configure a source filter from the **Application Source Qualifier**.

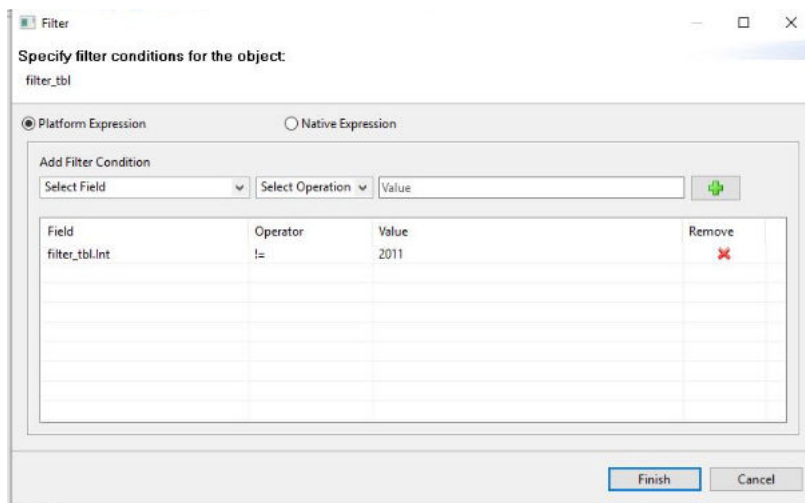
1. In the **Mapping Designer**, double-click the **Application Source Qualifier**.

The **Edit Transformation** tab appears as shown in the following image:



2. Click the **Edit Conditions** tab.
 3. Click **Edit Filter Conditions**.
- The **Add Filter Condition** dialog box appears.
4. Select the filter field and operator that you want to specify in the condition, enter a value for the condition, and click **Add Condition**.

The condition appears in the Filter Expression pane as shown in the following image:



5. Click **Finish** to add the filter condition.
6. Click **OK**.

Source Join

You can enter a join condition to join multiple MongoDB source tables.

Configure the source join from the Application Source Qualifier of the parent table. When you configure the join in the Mapping Designer, retain the source qualifier only of the parent table and then link the fields from the child tables to the parent source qualifier.

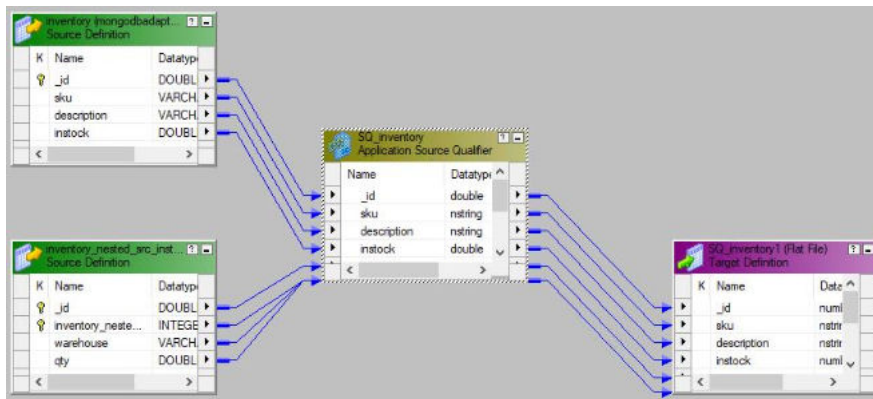
Use the Platform Expression or Native Expression type to define the relationship of the tables that you want to join.

Configuring a Source Join

You can use the Application Source Qualifier of the parent table to join multiple tables.

1. In the **Mapping Designer**, retain the source qualifier only of the parent table and manually delete the source qualifiers for the child tables.
2. Link the fields from the child tables to the parent source qualifier.

The following image shows an example mapping that contains the linked child tables with the parent source qualifier:

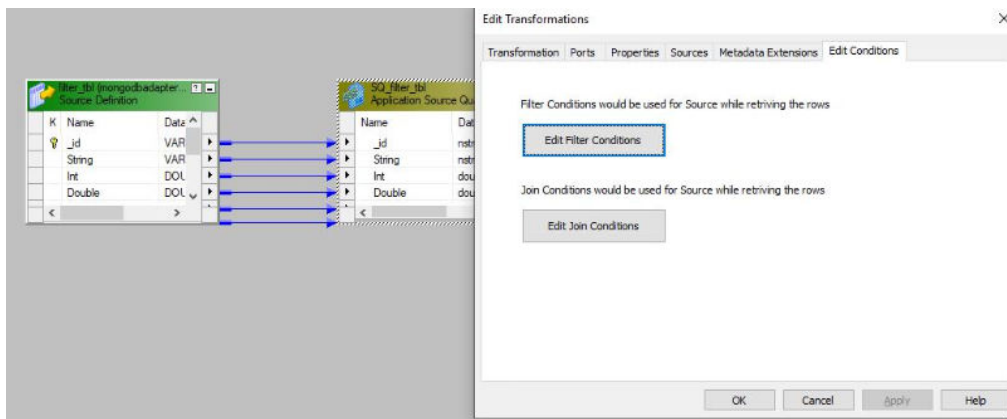


3. Double-click the **Application Source Qualifier** of the parent table.

The **Edit Transformation** tab appears.

The following image shows the **Edit Conditions** tab in the Application Source Qualifier Transformation type:

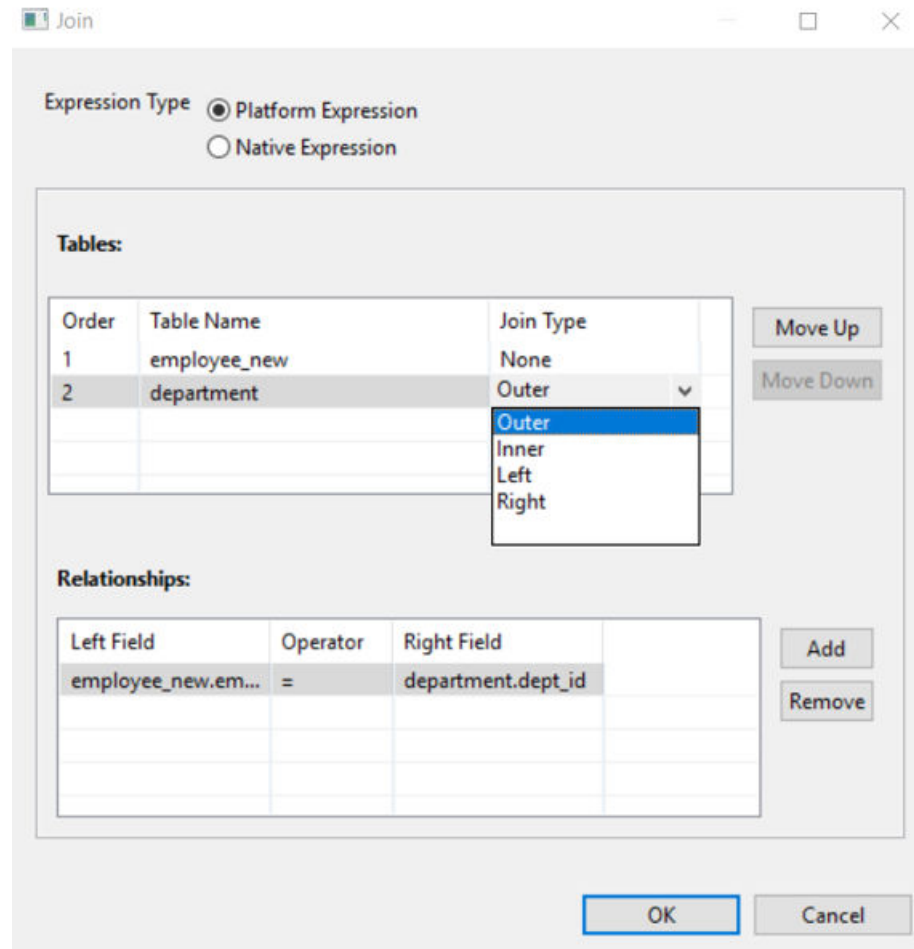
4. Click the **Edit Conditions** tab.



5. Click **Edit Join Conditions**.
6. To define a join condition for the tables using **Platform Expression**, perform the following tasks:
 - a. In the **Tables** section, select the child table, and then select the join type for the child table with the parent table.
 - b. In the **Relationships** section, define the relationship for the join.

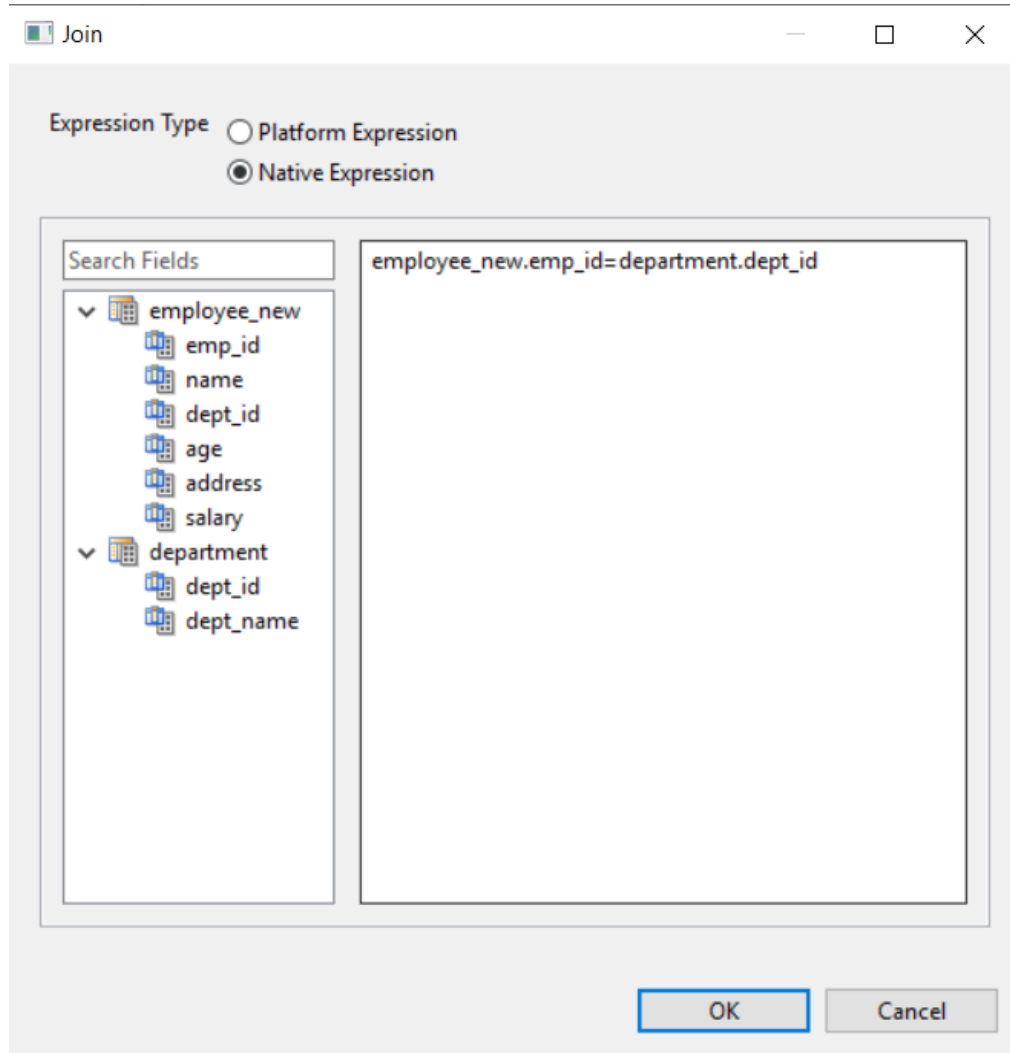
The **Add** button in the **Relationships** section enables only when you select the child table.

The following image shows the join types that you can configure and the relationship that you can define to join tables when you use the Infa expression:



7. To define a join condition using **Native Expression**, select the fields and define a join query syntax:

The following image shows the fields that you add when you define the join query:



8. Click **OK**.

Writing to an empty collection

You can write data to the empty MongoDB targets only in JSON mode by mapping the **DocumentAsJSON** field from the source to the target through which the entire MongoDB source collection will be written to the MongoDB Target collection.

MongoDB JDBC Mapping Example

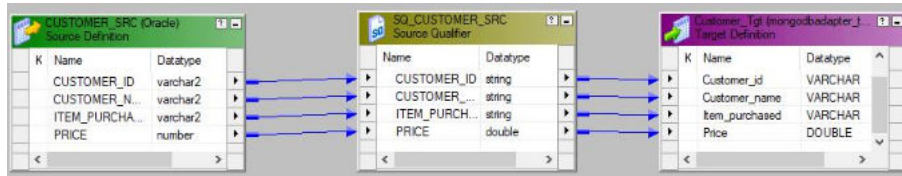
You work for an organization that stores purchase order details, such as customer ID, item codes, and item quantity in a relational database like Oracle. You need to analyze purchase order details and move data from the Oracle database to MongoDB.

Create a mapping to read all the purchase records from Oracle, process the data, and write them to a MongoDB target.

Perform the following tasks to move data from Oracle to MongoDB:

1. Import the Oracle source.
2. Import a MongoDB target.
3. Create a mapping with a Oracle source and a MongoDB target.

The following image shows the example mapping in the Mapping Designer:



4. Create a session and configure the session properties to load the data to the MongoDB target.

The mapping contains the following objects:

Source Definition

The mapping source definition is a relational database object. In the **Source Analyzer**, import the Oracle source. The PowerCenter Integration Service reads the customer transaction details from the Oracle source.

The following table describes the structure of the source definition called CUSTOMER_SRC:

Field	Data Type
CUSTOMER_ID	Varchar
CUSTOMER_NAME	Varchar
ITEM_PURCHASED	Varchar
PRICE	Float

Mapping Target

The mapping contains a MongoDB target definition.

In the **Target Designer**, import a MongoDB target definition.

The following table describes the structure of the target definition called CUSTOMER_TGT:

Field	Data Type
CUSTOMER_ID	Varchar
_ID	Varchar
CUSTOMER_NAME	Varchar
ITEM_PURCHASED	Varchar
PRICE	Double

CHAPTER 6

MongoDB JDBC Sessions

This chapter includes the following topics:

- [MongoDB JDBC Sessions Overview, 27](#)
- [PowerExchange for MongoDB JDBC Connections, 28](#)
- [Configure MongoDB JDBC Source Session Properties, 29](#)
- [Configure MongoDB JDBC Target Session Properties, 30](#)
- [Parameterization, 31](#)
- [Lookup, 31](#)
- [Update Strategy Transformation, 31](#)
- [Partitioning, 31](#)

MongoDB JDBC Sessions Overview

After you create mappings, you can create a session to extract and load data.

You must configure a MongoDB JDBC connection in the **Workflow Manager** to extract data from or load data to a MongoDB table. You can define properties in a session to determine how the PowerCenter Integration Service must extract data from a MongoDB source or load data to a MongoDB target.

PowerExchange for MongoDB JDBC Connections

When you configure a PowerExchange for MongoDB connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the MongoDB database.

The following table describes the MongoDB connection properties:

Property	Description
Connection Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! \$ % ^ & * () - + = { [] \ ; " ' < , > . ? /
Type	The connection type. Select MongoDB .
Host Name	Host name or IP address of the MongoDB server.
Port	MongoDB server port number. Default is 9042.
User Name	User name to access the MongoDB server.
Password	Password corresponding to the user name to access the MongoDB server.
Database Name	Name of the MongoDB database to connect to.
Additional Connection Properties	<p>Enter one or more JDBC connection parameters in the following format: <param1>=<value>&<param2>=<value>&<param3>=<value></p> <p>You must provide the JDBC parameters as ampersand-separated key-value pairs.</p> <p>You can configure the following JDBC connection parameters in a MongoDB connection:</p> <ul style="list-style-type: none"> - AuthSource - BatchSize - connectTimeoutMS - DefaultStringColumnLength - DmlBatchSize - EnableDoubleBuffer - EnableTransaction - LogLevel - LogPath - SamplingLimit - SamplingStepSize - SamplingStrategy - useJSONColumn <p>For example, DefaultStringColumnLength=512&DmlBatchSize=1000& EnableDoubleBuffer=false&EnableTransaction=true& SamplingLimit=200&SamplingStepSize=2&SamplingStrategy=Backwards</p> <p>Note: If you specify the host name, port number, user name, and password of the MongoDB server in the Additional Connection Properties, the values specified in the Additional Connection Properties takes precedence.</p>
SSL Mode	Options applicable for MongoDB Connector are: <ul style="list-style-type: none"> - disabled - require or one-way

Note: If you select the **Enable Reading/Writing as JSON** option, a column **documentAsJSON** appears in the collection when you read data from MongoDB through which you can read data as JSON. Default is disabled. To enable reading or writing as JSON, select **useJSONColumn=true**.

Configuring a MongoDB JDBC Connection

Configure a MongoDB JDBC connection in the Workflow Manager to define the connection attributes that the PowerCenter Integration Service uses to connect to the MongoDB database.

1. In the Workflow Manager, click **Connections > Application**.
The **Application Connection Browser** dialog box appears.
2. Click **New** and select **mongodbadapter_table**.
The **Application Connection Editor** dialog box appears.
3. Enter a name for the MongoDB JDBC connection.
4. Enter the MongoDB JDBC connection attributes.
5. Click **OK** to create a MongoDB JDBC connection.

Configure MongoDB JDBC Source Session Properties

You can configure the session properties for a MongoDB source on the **Workflow Manager** tab. Define the properties for the source instance in the session.

The following table describes the session properties that you can configure for a MongoDB source session:

Session Property	Description
Pre-SQL	The pre-SQL commands to run before the PowerCenter Integration Service reads data from the source.
Post-SQL	The post-SQL commands to run after the PowerCenter Integration Service reads data from source.
Filter Override Type	The type of filter expression that you want to override in the source qualifier. You can select Native or Platform filter expression type to override. Default is None.
Filter Override	The filter condition that overrides the filter condition you specify in the source qualifier. After you select Native or Platform Filter Override Type, specify the filter condition to override in the source qualifier. Note: When you configure a Platform filter override, you must specify the absolute qualified column name in the following format: <code>tablename.columnname</code>

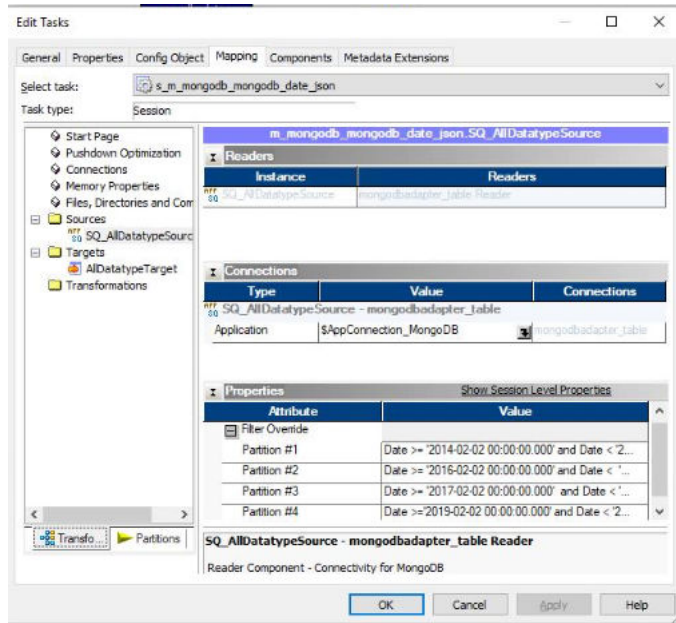
Configure a Filter Override

When you read data from a MongoDB source, you can specify the filter override type and filter override condition in the MongoDB source session properties to override the filter condition you specify in the source qualifier.

You must select the corresponding Native or Platform expression as the **Filter Override Type** in the MongoDB source session properties based on the expression used in the source qualifier. Default is None.

After you select the filter expression, specify the filter condition in the **Filter Override** field. When you run the session, the PowerCenter overrides the filter condition you specify in the source qualifier and uses the condition you specified in the session properties to filter the MongoDB data.

The following image shows the filter override type and the filter override condition fields in the MongoDB source session properties:



Configure MongoDB JDBC Target Session Properties

You can configure the session properties for a MongoDB target on the **Workflow Manager** tab. Define the properties for the target instance in the session.

The following table describes the session properties that you can configure for a MongoDB target session:

Session Property	Description
Pre-SQL	The pre-SQL commands to run before the PowerCenter Integration Service writes data to the target.
Post-SQL	The post-SQL commands to run after the PowerCenter Integration Service writes data to the target.

Parameterization

You can parameterize the MongoDB JDBC connection and session properties.

You can create a configuration file with the connection and session properties that you want to override at run time.

Lookup

You can configure a cached connected Lookup transformation in a mapping. A connected Lookup transformation is connected to a MongoDB source or target in a mapping. When you create a cached lookup, the performance increases because the Integration Service caches a large lookup source or small lookup tables. When you cache the lookup source, the Integration Service queries the lookup cache instead of querying the lookup source for each input row.

When you run a mapping that contains a connected Lookup transformation, the Integration Service passes values from another transformation to input ports in the Lookup transformation. For each input row, the Integration Service queries the lookup source or cache based on the lookup ports and the lookup condition in the transformation. The Integration Service returns data from the query and passes it to the next transformation in the mapping.

Update Strategy Transformation

You can set the update strategy at the mapping or the session level:

- Within a session. When you configure a session, you can instruct the Integration Service to either treat all rows in the same way. You can flag all rows for insert, delete, or update, or you can select the data driven option, where the Integration Service follows instructions coded into Update Strategy transformations within the session mapping.
- Within a mapping. Within a mapping, you use the Update Strategy transformation to flag all rows for insert, delete, update, or reject.

For more information, see the Update Strategy Transformation chapter in the *PowerCenter Transformation Guide*.

Partitioning

When you read from and write data to MongoDB, you can configure pass-through partitioning to optimize the session performance at run time.

When you specify pass-through partitioning for a MongoDB Source Qualifier transformation, you can specify filter conditions in the MongoDB session properties to override the filter condition you specify in the source qualifier. The Integration Service uses the filter condition you specify in the session properties when it filters data from the source.

To configure pass-through partitioning, select the Source Qualifier transformation, and add a partition point from the **Mapping** tab of the session properties. Add the number of partitions you require and select the partition type as pass through for each of the partitions.

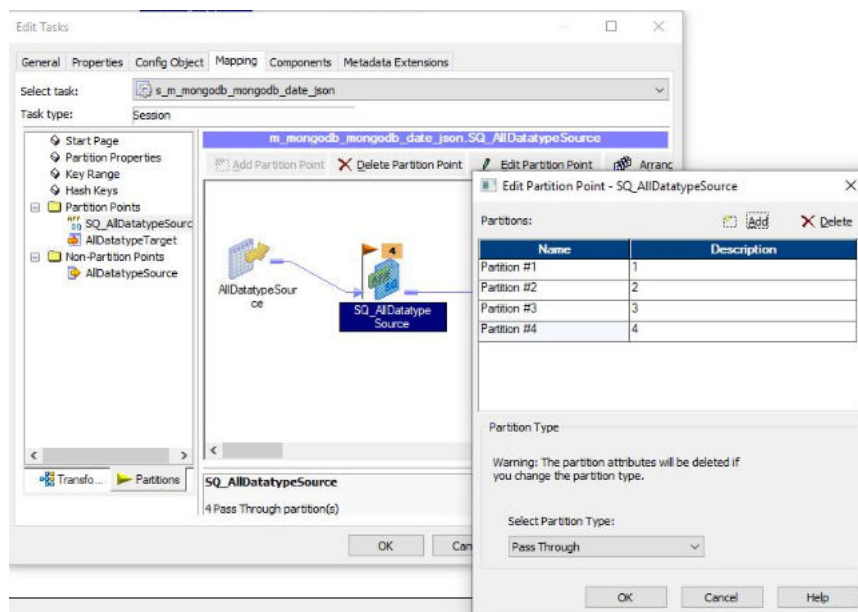
Based on the number of partitions you add, the PowerCenter Integration Service adds those many number of partition fields for the **Filter Override** attribute in the session properties. Specify the filter override condition for each of the partitions. The PowerCenter Integration Service uses the filter conditions you specify to pass data through the appropriate partition.

Similarly, you can specify the fetch size for each of the partitions in the MongoDB source session properties.

Configure Pass-Through Partitioning for a MongoDB JDBC Session

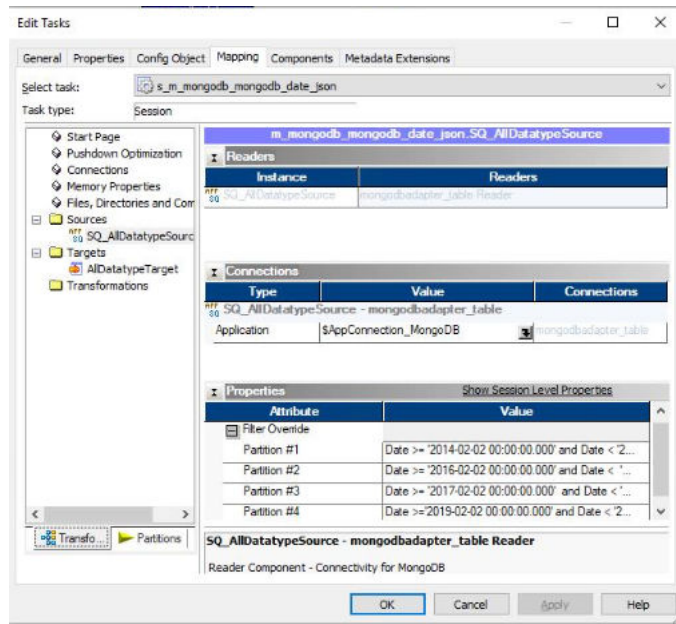
Configure pass-through partitioning for a session and specify the filter override condition for each partition. When you configure a filter override in a pass-through partitioning, key range partitioning is enabled.

1. In the Workflow Manager, double-click the Session task to open the session properties, and click the **Mapping** tab.
2. In the **Partitions** tab, double-click the source qualifier and add the required number of partitions.
3. Select the **Partition Type** as **Pass Through**.



4. Click the **Transformations** tab, and navigate to the session properties.

- In the Filter Override session property, specify the filter override condition for each partition.



- Click **OK**.

CHAPTER 7

Data Type Reference

This chapter includes the following topics:

- [Data Type Reference Overview, 34](#)
- [MongoDB and Transformation Data Types, 34](#)

Data Type Reference Overview

PowerCenter Integration uses the following data types in mappings and sessions with MongoDB:

MongoDB native data types

MongoDB data types appear in the **Fields** tab for Source and Target transformations when you choose to edit metadata for the fields.

Transformation data types

Set of data types that appear in the transformations. They are internal data types based on ANSI SQL-92 generic data types, which the Secure Agent uses to move data across platforms. Transformation data types appear in all transformations in a mapping.

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

MongoDB and Transformation Data Types

The following table describes the data types that PowerCenter Integration supports for MongoDB sources and targets:

MongoDB Data Types	Transformation Data Types	Range and Description
Array	NA	Simba MongoDB JDBC driver re-normalizes the data in the array into virtual tables.
Boolean	String	Precision of 1

MongoDB Data Types	Transformation Data Types	Range and Description
Date	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to second)
NumberDouble	Double	Precision 15
NumberInt	Integer	Precision 10, scale 0
NumberLong	Decimal	Precision 1 to 28 digits, scale 0 to 28
Object	NA	Simba MongoDB JDBC driver re-normalizes the data in the object into virtual tables.
ObjectID	String	1 to 104,857,600 characters
String	String	1 to 104,857,600 characters

INDEX

C

configuration
 overview [11](#)
 prerequisites [11](#)
 product availability matrix [11](#)
configuring source filter [22](#)
configuring source join [23](#)

F

filter
 application source qualifier [21](#)
 source filter [21](#)
filter conditions [22](#)

I

Import
 connection properties [17](#)
 source definition [17](#)
 target definition [17](#)
Introduction
 MongoDB [9](#)
 PowerExchange for MongoDB [9](#)

J

join
 application source qualifier [23](#)
 source join [23](#)
join conditions [23](#)

M

mapping
 example [25](#)
 filter [21](#)
 source [21](#)
 target [21](#)

MongoDB
 mappings [21](#)
MongoDB data types
 mapping to transformation data types [34](#)
 overview [34](#)
MongoDB JDBC
 source [17](#)
 target [17](#)

P

plug-in registration
 administrator tool [12](#)
 command line interface [13](#)
PowerExchange for MongoDB
 configuration [11](#)
PowerExchange for MongoDB JDBC
 plug-in registration [12](#)

S

session
 additional JDBC URL parameters [28](#)
 connection configuration [29](#)
 connection properties [28](#)
 database/schema [28](#)
 overview [27](#)
 source properties [29](#)
 target properties [30](#)

V

virtual table
 collection [14](#)
 list [14](#)
 map [14](#)
 set [14](#)
virtual tables
 example [15](#)