



Informatica® PowerExchange for
webMethods

10.1.1

User Guide for PowerCenter

© Copyright Informatica LLC 2010, 2018

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.

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/hsqLicense.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/license.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.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, please report them to us in writing at Informatica LLC 2100 Seaport Blvd. Redwood City, CA 94063.

INFORMATICA LLC 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: 2018-09-28

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 webMethods	 8
Understanding PowerExchange for webMethods Overview.	8
Integrating PowerCenter with webMethods.	8
Designer Integration with the webMethods Broker.	9
webMethods Broker.	9
webMethods Documents.	10
PowerCenter Integration Service and webMethods Integration.	10
 Chapter 2: Configuring PowerExchange for webMethods.....	 12
Configuring PowerExchange for webMethods Overview.	12
Before You Begin.	12
Step 1. Configure JVM Options for the PowerCenter Integration Service.	13
Step 2. Register the webMethods Libraries.	13
Downloading the webMethods Service Pack.	13
Setting the Path to the webMethods Broker bin Directory.	13
Adding the Client JAR File to the PowerCenter Integration Service Classpath.	13
 Chapter 3: webMethods Source and Target Concepts.....	 14
webMethods Source and Target Concepts Overview.	14
Envelope Fields.	14
Data Fields.	15
Map Values.	16
webMethods Sequence Metadata.	16
Sequence Containing a Structure.	17
Multi-Dimensional Sequence.	18
Nested Sequence.	18
Documents with Special Characters.	19
 Chapter 4: Working with webMethods Sources.....	 20
Working with webMethods Sources Overview.	20

Creating a webMethods Source Definition.	20
Displaying Document Types in a Document Folder.	20
Importing a webMethods Source Definition.	21
Editing a webMethods Source Definition.	21
Working with the Application Multi-Group Source Qualifier.	22
Chapter 5: Working with webMethods Targets.	23
Working with webMethods Targets Overview.	23
Working with the pubSeqn Envelope Field.	23
Maintaining Transactional Consistency for webMethods Targets	23
Importing a webMethods Target Definition.	24
Editing webMethods Target Definitions.	25
Chapter 6: Creating and Configuring webMethods Workflows.	26
Creating and Configuring webMethods Workflows Overview.	26
Configuring the Delivery Model for webMethods Sources.	26
Filtering Source Documents.	27
Configuring Terminating Conditions.	27
Configuring Real-time Processing.	28
Configuring Message Recovery.	28
Configuring the Delivery Model for webMethods Targets.	28
Reading and Writing Sequences in webMethods Documents.	29
Configuring Transactional Consistency for webMethods Targets.	30
Pipeline Partitioning.	30
Configuring a Session with a webMethods Mapping.	30
Properties Tab - General Options.	30
Config Object Tab - Error Handling.	31
Mapping Tab - Sources Node.	31
Mapping Tab - Targets Node.	32
Configuring and Scheduling Workflows.	33
Configuring Workflows to Run Simultaneous Sessions.	33
Scheduling Workflows.	33
Troubleshooting webMethods Workflows.	33
Appendix A: webMethods Datatype Reference	34
webMethods and Transformation Datatypes.	34
Boolean Datatype.	35
Appendix B: Glossary.	36
Index.	38

Preface

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

This book assumes you have knowledge of relational database concepts and database engines, PowerCenter®, and webMethods. You must 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 webMethods

This chapter includes the following topics:

- [Understanding PowerExchange for webMethods Overview, 8](#)
- [Integrating PowerCenter with webMethods, 8](#)

Understanding PowerExchange for webMethods Overview

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

The PowerCenter Integration Service connects to a webMethods Broker to read and write webMethods documents. webMethods Brokers receive, queue, and send webMethods documents. They also store webMethods document metadata.

The PowerCenter Integration Service can read webMethods documents based on the client ID or the document type you define in the webMethods source definition. Similarly, it can write webMethods documents based on the client ID or the document type you define in the webMethods target definition. The PowerCenter Integration Service can also read and write webMethods request/reply documents.

Integrating PowerCenter with webMethods

PowerExchange for webMethods enables the PowerCenter Integration Service to read documents from webMethods sources and write documents to webMethods targets. To read and write webMethods documents, create mappings with webMethods source and target definitions. Connect to a webMethods Broker to import webMethods source and target metadata for a mapping.

Create a session and workflow for the mapping. The PowerCenter Integration Service connects to a webMethods Broker to read and write webMethods documents.

Designer Integration with the webMethods Broker

The Designer connects to a webMethods Broker to import metadata for webMethods source and target definitions. Before you can connect to a webMethods Broker, define the Broker, the document metadata you want the Broker to contain, and a host name and port number for the Broker. The Designer uses the host name and port number to connect to the Broker to retrieve webMethods document metadata.

webMethods Broker

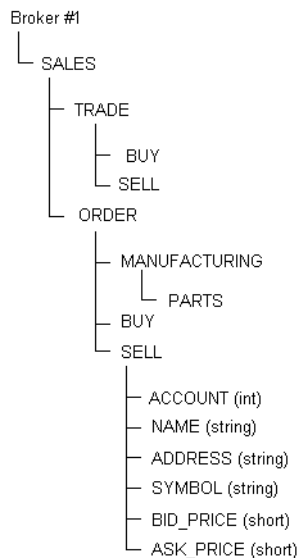
A webMethods Broker stores metadata for webMethods documents in the form of document types. The Designer uses the Broker host name and port number to connect to the Broker to import webMethods source and target metadata. The PowerCenter Integration Service integrates with webMethods Brokers to read and write webMethods documents.

webMethods Enterprise Integrator lets you define metadata for webMethods documents. The Broker stores the document metadata. When you define webMethods document metadata, specify a client group, document folder, document type, and document data field metadata. After you define document folders and types, associate them with client groups in the Enterprise Server Manager.

The *client group* controls access to document types. When you connect to a Broker to import webMethods source or target definitions, you specify a client group. The Designer displays document folders that belong to that client group.

Client groups can contain multiple *document folders*. Document folders can contain other document folders. They also contain *document types*. A document type represents the structure of the documents that belong to it. Document types contain data field names and datatypes for a webMethods document. When you import a webMethods source or target definition, select a document type. The source or target definition contains metadata that represents that document type.

The following figure shows an example of document metadata structure in a webMethods Broker:



In this example, Broker #1 is the broker name. SALES is the client group. ORDER and MANUFACTURING are examples of document folders. PARTS is an example of a document type. ACCOUNT is an example of a data field name. The datatype of the data field occurs after the data field name.

webMethods Documents

webMethods documents consist of data fields and envelope fields. You define data field names and their datatypes in a Broker. Envelope fields are header fields. webMethods source and target definitions can include a predefined set of envelope fields.

The following table describes the document envelope fields that appear in webMethods source and target definitions:

Header Field	Description
pubID	Client ID of the publisher of the document.
destID	Client ID of the recipient of the document when the delivery mode is deliver/receive.
pubSeqn	64-bit value that represents the publish sequence number of the document.
recvTime	Date and time the Broker receives the document.
enqueueTime	Date and time the Broker places the document in the queue for the recipient.
trackId	Tracks published documents.
tag	Matches a request document with the corresponding reply document.
maxResults	Maximum number of reply documents a requestor wants to receive.
startResult	Value greater than or equal to 0 that signifies the starting number of the document to be received. Generally used in conjunction with maxResults.
replyTo	Client ID to which the document publisher sends reply documents instead of the source of the request document.
errorsTo	Client ID to which the document publisher forwards a document if it encounters an error processing the document. When there is no value for the field, the publisher discards the document.
errorRequestsTo	Client ID to which the document publisher forwards a request/reply document if it encounters an error processing the request. When there is no value for the field, the publisher discards the document.
transactionId	Indicates that a document is part of a transaction.
appSeqn	Sequence number the document publisher sets for the document.
appLastSeqn	Last event of the sequence. When you include a value for this field, it should be the same value as the appSeqn field for the document.

PowerCenter Integration Service and webMethods Integration

The PowerCenter Integration Service connects to a webMethods Broker to read webMethods source documents or write webMethods target documents. During session initialization, the PowerCenter Integration Service creates a Broker Client to connect to a Broker. The PowerCenter Integration Service connects to the Broker using the Broker name, host name, and port number you specified in the application connection you configured for a webMethods source or target.

The PowerCenter Integration Service can read and write documents with different document delivery models.

The PowerCenter Integration Service can read documents from webMethods sources and write documents to webMethods targets with the following delivery models:

- Request/reply
- Publish/subscribe

The request/reply delivery model is for documents that should receive a reply. The reply can contain a response to the request or an acknowledgement of receipt.

The publish/subscribe model is for documents you want to publish or receive that do not require a response. The publish/subscribe model enables the PowerCenter Integration Service to read and write documents in the following modes:

- Deliver/receive
- Broadcast

Deliver/receive mode is a point-to-point delivery of documents by client ID. Only a Broker Client with a certain client ID can receive the documents. When the PowerCenter Integration Service reads and writes documents in deliver/receive mode, it uses a client ID to read documents from a Broker. When you configure a webMethods application connection, you can specify a client ID for the PowerCenter Integration Service to use. Or, the PowerCenter Integration Service can generate a client ID during session initialization.

Broadcast mode means that the document sender sends documents to a Broker for any Broker Client subscribing to the documents to receive. When the PowerCenter Integration Service reads documents in broadcast mode, it reads documents according to the document type that you define in the source definition in the mapping. When the PowerCenter Integration Service writes documents in broadcast mode, it writes documents according to the document type you define in the target definition in the mapping.

CHAPTER 2

Configuring PowerExchange for webMethods

This chapter includes the following topics:

- [Configuring PowerExchange for webMethods Overview, 12](#)
- [Step 1. Configure JVM Options for the PowerCenter Integration Service, 13](#)
- [Step 2. Register the webMethods Libraries, 13](#)

Configuring PowerExchange for webMethods Overview

PowerExchange for webMethods requires configuration on the PowerCenter Integration Service and PowerCenter Client to enable integration between PowerCenter and webMethods.

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

1. Configure JVM parameters the PowerCenter Integration Service. You must configure JVM options.
2. Register libraries and configure the PowerCenter Integration Service. Register webMethods libraries on the machine running the PowerCenter Integration Service and configure the PowerCenter Integration Service to set the CLASSPATH to the webMethods JAR files.

Note: Configuring the PowerCenter Integration Service to set the path and the CLASSPATH to the JVM library file is required for UNIX. It is optional for Windows.

The PowerCenter Integration Service uses webMethods application connections to connect to a webMethods Broker when it reads webMethods source documents and writes webMethods target documents.

Before You Begin

Before configuring PowerExchange for webMethods, install and configure the following webMethods components:

- webMethods Client. Install and configure on the machine hosting the PowerCenter Client.
- webMethods Enterprise Server C API. Install and configure on the machine hosting the PowerCenter Client.
- webMethods Enterprise Server Java API. Install and configure on the machine running the PowerCenter Integration Service.

Step 1. Configure JVM Options for the PowerCenter Integration Service

When you configure PowerExchange for webMethods, you must configure JVM options in the Informatica Administrator. Configure JVM options to run Java-based programs. When you configure JVM options, set the following properties:

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

Step 2. Register the webMethods Libraries

To register webMethods libraries, complete the following tasks:

- Download the webMethods service pack from webMethods.Advantage.
- Set the path to the webMethods Enterprise Server directory on Windows.
- Add the webMethods Client JAR File to the PowerCenter Integration Service Classpath.

Downloading the webMethods Service Pack

The PowerCenter Integration Service uses the webMethods service pack to subscribe to webMethods Broker documents. The webMethods service pack contains a series of dynamic link libraries (DLLs) and the webMethods client JAR file.

Download the webMethods service pack from the Software AG web site. webMethods provides instructions to copy the DLLs and access the client JAR file.

Note: You require a webMethods account to download the service pack.

Download the webMethods service pack from the Software AG web site at the following location:

<https://empower.softwareag.com/>. Follow the instructions provided by webMethods in the readme file to complete the download.

Setting the Path to the webMethods Broker bin Directory

On the PowerCenter Client machine, set the path to the webMethods \bin directory. For example:

```
;d:\<webMethods_Broker_directory>\bin
```

Adding the Client JAR File to the PowerCenter Integration Service Classpath

On the PowerCenter Integration Service machine, add the webMethods client JAR file to the PowerCenter Integration Service classpath. For webMethods 5.0, add client50.jar. For webMethods 6.0, add client60.jar.

CHAPTER 3

webMethods Source and Target Concepts

This chapter includes the following topics:

- [webMethods Source and Target Concepts Overview, 14](#)
- [webMethods Sequence Metadata, 16](#)
- [Sequence Containing a Structure, 17](#)
- [Multi-Dimensional Sequence, 18](#)
- [Nested Sequence, 18](#)
- [Documents with Special Characters, 19](#)

webMethods Source and Target Concepts Overview

webMethods source and target definitions represent metadata for webMethods documents. A webMethods document can contain the following types of fields:

- Envelope fields
- Data fields

If the document type of a source or target document does not match the document type of the source or target definition, the PowerCenter Integration Service rejects the document. When the PowerCenter Integration Service rejects a document, it enters the content of the rejected document into the session log, and the document is no longer available at the Broker.

Envelope Fields

webMethods source and target definitions can include webMethods document envelope fields. You can include envelope fields in a source definition when you want to process data that pertains to the header information in webMethods documents. For example, you may want to track the publisher ID of webMethods documents you read from a source.

Include envelope fields in a target definition when you want to write document header information to a webMethods target. For example, you may want the PowerCenter Integration Service to send target documents according to a particular publication sequence.

Include envelope fields in a source or target definition when you import the definition. When you define a webMethods target based on a source definition, and the source definition contains envelope fields, the Designer adds the appropriate envelope fields to the target definition.

The following table describes the envelope fields webMethods source and target definitions can contain:

webMethods Document Envelope Field	webMethods Source Definition	webMethods Target Definition
pubID	Yes	No
destID	Yes	Yes
pubSeqn	Yes	Yes
recvTime	Yes	No
enqueueTime	Yes	No
trackId	Yes	Yes
Tag	Yes	Yes
maxResults	No	Yes
startResult	Yes	Yes
replyTo	Yes	Yes
errorsTo	Yes	Yes
errorRequestsTo	Yes	Yes
transactionId	Yes	Yes
appSeqn	Yes	Yes
appLastSeqn	Yes	Yes

Data Fields

webMethods source and target definitions can contain webMethods document data fields. When you import webMethods definitions with data fields, the data fields represent webMethods documents of the document type you imported.

Data fields include webMethods sequence fields. You can extract data from webMethods sequence fields and load data into webMethods sequence fields. A webMethods sequence field contains a sequence of multiple values of the same datatype. When you import webMethods source or target definitions that contain sequence fields, the metadata for each sequence field is represented by a distinct group.

You can import the following kinds of webMethods sequences in the Designer:

- Sequence containing a structure
- Multi-Dimensional sequence
- Nested sequence

Map Values

Each column in a webMethods source or target definition has a map value. The map value indicates the relationship of the field in the document type hierarchy. The PowerCenter Integration Service reads and writes documents according to the document type structure defined in the map value of the webMethods definition.

The map value shows which columns in a definition represent metadata for envelope fields or data fields. In the map value, all envelope fields include the prefix “_env.”. All data fields contain the name of the corresponding field you defined for the document type in webMethods Enterprise Integrator.

For example, in Enterprise Integrator you define data fields under the document type BUY. You include some of the data fields under a structure folder called DESCRIPTION. The document type BUY is in the document folder TRADE. You import a target definition of the document type BUY from the document folder TRADE. You include the default envelope fields in the definition. When you view the Attributes tab of the target definition, you see the envelope fields preceded by _env. and the data fields.

webMethods Sequence Metadata

webMethods documents can include webMethods sequence data fields. A sequence field contains a sequence of multiple values of the same datatype. When you import webMethods document metadata that contains a sequence field, the Designer creates a source or target definition with multiple groups.

The Designer creates a root group with top-level webMethods data fields. The root group includes envelope fields and single-dimensional sequences of the Char, Unichar, or Byte webMethods datatypes. For each sequence in a webMethods document that is of a webMethods datatype other than Char, Unichar, or Byte, the Designer creates a sequence group. The Designer also creates a sequence group for Char, Unichar, or Byte datatypes if there is more than one dimension. The Designer names each sequence group according to the sequence name.

The Designer creates a key relationship between each sequence group in a webMethods source definition and its parent group. It also creates an index key for each dimension in a sequence. Each key is named according to the following convention:

```
{GPK|GFK|IDX}<group_name><index_number>
```

The following table describes the components of a key name in a webMethods source or target definition with a sequence:

Key Name Component	Description
GPK GFK IDX	Type of key. The key name begins with GPK when it is a primary key. The key name begins with GFK when it is a foreign key. The key name begins with IDX when it is an index key.
group_name	Name of the group to which the key belongs.
index_number	Distinguishes index keys in the same group. The value of this component for the first index key in a group is 0. For each subsequent index key in a group, the value is incremented by one. This component only appears for index keys.

For example, you want to import metadata for a document type with sequence data fields as a source definition into Sales. The document type contains the following fields:

Type	Name
string	a
int[]	b

The following figure shows a webMethods source definition with a sequence:

Name	Datatype
Sales::Simple_Array	
GPk_Sales::Simple_Array	int
A	string
b	
GFK_b	int
IDX_b_0	int
b	int

In this example, the root group takes the name of the webMethods document, Sales::Simple Array. The Sales::Simple_Array group contains the data field a. The Designer represents the webMethods sequence field b as a group called b. The Sales::Simple_Array and b groups share a primary key/foreign key relationship. The Sales::Simple_Array root group contains the GPk_Sales::Simple_Array primary key. Since the GFK_b foreign key in group b references the GPk_Sales::Simple_Array primary key, group b is a child group to the Sales::Simple_Array group. Group b also contains the IDX_b_0 index key.

Sequence Containing a Structure

A webMethods sequence can contain a structure. A structure is a collection of fields that defines a webMethods data field. The fields that comprise a structure can be simple datatypes, sequences, or other structures. When you import a webMethods definition with a sequence that contains a structure, the Designer creates the structure fields in the sequence group.

For example, you want to import metadata for a document type with a sequence that contains a structure as a source definition in the Designer. The document type is called dept and is in the Sales folder. The document type contains the following fields:

Type	Name
string	dept_name
struct[]	dept_people
string	ssn
string	name

The Designer creates a multi-group source definition for the dept webMethods document. The source definition for the dept webMethods document includes the dept_people sequence group with the structure fields ssn and name.

The following figure shows the source definition for the dept webMethods document:

Name	Datatype
Sales::dept	
GPk_Sales::dept	int
dept_name	string
dept_people	
GFK_dept_people	int
IDX_dept_people_0	int
ssn	string
name	string

Note: The Designer displays an error when you try to import a sequence with a structure that does not contain fields.

Multi-Dimensional Sequence

A webMethods document can contain a sequence with multiple dimensions. When you import a webMethods definition with a multi-dimensional sequence, the Designer creates a unique index key for each dimension in the sequence.

For example, you want to import metadata for a document type with a multi-dimensional sequence as a source definition in the Designer. The document type is called company and is in the Sales folder. The document type contains the following fields:

Type	Name
string	company_name
string	dept_name
struct[][]	company_people
string	ssn
string	name

The company webMethods document contains a two-dimensional sequence called company_people. When you create the target definition for the company webMethods document, the Designer creates the IDX_company_people_0 index key for the first dimension and the IDX_company_people_1 index key for the second dimension.

The following figure shows the target definition for a webMethods document with a multi-dimensional sequence:

Name	Datatype
Sales::company	
GPK_Sales::company	int
company_name	string
company_people	
GFK_company_people	int
IDX_company_people_0	int
IDX_company_people_1	int
dept_name	string
ssn	string
name	string

Nested Sequence

A webMethods document can contain a nested sequence. When you import a webMethods definition with a nested sequence, the Designer creates a group for each sequence. The Designer creates a foreign key in the child sequence group and a primary key in the parent sequence group. This foreign key references the primary key that the Designer creates in the parent sequence group.

For example, you want to import metadata for a document type with a nested sequence as a source definition in the Designer. The document type is called org and is in the Sales folder. The document type contains the following fields:

Type	Name
string	org_name
struct[]	dept
string	dept_name
struct[]	people
int	ssn
string	name

The company webMethods document contains the people sequence nested in the dept sequence. When you import the company webMethods document, the Designer creates a key relationship between the dept and people groups. Since the dept sequence contains the people sequence, the Designer includes a primary key in

the dept group and a foreign key in the people group. The Designer also creates a key relationship between the Sales::org root group and the dept group.

The following figure shows the source definition for a webMethods document with a nested sequence:

Key Types	Name	Datatype
	Sales::org	
PRIMARY KEY	GPK_Sales::org	int
NOT A KEY	org_name	string
	dept	
PRIMARY KEY	GPK_dept	int
FOREIGN KEY	GFK_dept	int
NOT A KEY	IDX_dept_0	int
NOT A KEY	dept_name	string
	people	
FOREIGN KEY	GFK_people	int
NOT A KEY	IDX_people_0	int
NOT A KEY	ssn	string
NOT A KEY	name	string

Documents with Special Characters

webMethods documents can contain special characters like asterisk (*), underscore (_), and hyphen (-). A webMethods broker connection cannot import documents with special characters as sources into PowerCenter. Create a webMethods Integration Server connection to import source documents with special characters. You cannot import webMethods documents with special characters as target definitions.

Install the InformaticaMetadata package in the webMethods Integration Server to use the webMethods Integration Server connection. When you install the InformaticaMetadata package, an infoSet called informaticaWireTags is created in the webMethods Integration Server. InfoSet adds metadata along with the webMethods document definition. informaticaWireTags stores the details of all the special characters in a webMethods document. Ensure that you always synchronize the informaticaWireTags with the current document definition in the webMethods server.

CHAPTER 4

Working with webMethods Sources

This chapter includes the following topics:

- [Working with webMethods Sources Overview, 20](#)
- [Creating a webMethods Source Definition, 20](#)
- [Editing a webMethods Source Definition, 21](#)
- [Working with the Application Multi-Group Source Qualifier, 22](#)

Working with webMethods Sources Overview

You import a webMethods source definition or create one manually. When you import or create webMethods source definitions, the Designer displays a table with document fields and webMethods datatypes. When the PowerCenter Integration Service extracts data from a webMethods source, it converts the data based on the datatypes in the Application Multi-Group Source Qualifier transformation associated with the source.

Creating a webMethods Source Definition

You can import a webMethods source definition or create one manually. You import webMethods document metadata from a Broker. To import a source definition from a Broker, enter a Broker host name. If the Broker port number is not the default port number, enter a port number. Also enter the name of a client group the Broker contains. When you connect to the Broker, it lists all document folders and document types in the folders that belong to the client group in the following format: <document folder>::<document type>.

Displaying Document Types in a Document Folder

You can display document types that belong to a single document folder. Enter a document folder name in the Scope Name option to when you import a source definition from a webMethods Broker.

For example, you want to view those document types that belong to the document folder ORDER in the client group SALES. In the Import from webMethods Broker dialog box, enter ORDER as the Scope Name to display the document types that directly belong to the document folder ORDER.

If a document type belongs to a document folder within a document folder, it does not display when you use a scope name to filter document types. For example, if a document folder ORDER contains a subfolder

SUPPLIES, when you enter ORDER for the scope name, document types in the subfolder SUPPLIES do not display.

Importing a webMethods Source Definition

The Designer names source definitions according to the document folder and document type on which you base the source definition. When you import the first source definition, the Designer places it under a database definition node in the repository folder named for the Broker host name, port number, and Broker name from which you imported the source definition.

1. In the Source Analyzer, click **Sources > Import from webMethods Broker**.

The **Import from webMethods Broker** dialog box appears.

2. Enter the following information:

Import Attribute	Description
Broker Host	Host name and port number of the Broker in the following format: <host_name>:<port_number> You can omit the port number if the port number for the Broker is the default port number. Default port number is 6849.
Broker Name	Name of the Broker.
Client Group	Client group where the document metadata is stored.
Scope Name	Document folder name that displays the document types in the document folder. Document types that belong to subfolders of the document folder you enter for the scope name do not appear.

3. Click **Connect**.

The Designer displays a list of document types and their document folders.

4. Select the document type you want to import.
5. Optionally, select Create Default Envelope Fields to add the default envelope fields to the source definition.
6. Click **Import**.

A webMethods source definition appears in the Source Analyzer.

Note: webMethods document types are case sensitive. However, source definition names are not. Therefore, when you import a webMethods source definition, and the repository already contains a source definition with the same name that uses different capitalization, the Designer might ask if you want to overwrite the existing source definition. Also, the Designer displays an error when you try to import a sequence with a structure that does not contain fields.

Editing a webMethods Source Definition

Edit webMethods source definitions to change the precision of certain fields. You can modify the name and description of source definitions. You can also change column names. When you change column names, the Designer does not update the map value to reflect the changes.

Note: To update a source definition to reflect changes in the webMethods document metadata, recreate the source definition.

1. Double-click the title bar of the source definition in the Source Analyzer.
2. On the Table tab of the **Edit Tables** dialog box, optionally edit the following settings:

Table Settings	Edit	Description
Select Table	Yes	Displays the source definition you are editing. To edit a different source or target definition, select it from the list.
Rename button	Yes	Edit the name of the source definition. You can also edit the database definition node name.
Business Name	No	Not applicable for webMethods source definitions.
Owner Name	Yes	Not applicable for webMethods source definitions.
Description	Yes	Optional description of the source definition. Character limit is 2000 bytes/K, where K is the maximum number of bytes for each character in the PowerCenter repository code page. Enter links to business documentation.
Database Type	No	Database type is set to webMethods.

3. Click the **Columns** tab.
4. Optionally, edit the precision of any applicable fields.
5. Optionally, modify the name of any fields.
6. Click **OK**.

Working with the Application Multi-Group Source Qualifier

An Application Multi-Group Source Qualifier in a mapping determines how the PowerCenter Integration Service reads data from the webMethods source. To extract data from multiple webMethods sources, you must have an Application Multi-Group Source Qualifier for each source definition in the mapping. The PowerCenter Integration Service reads documents from the webMethods source based on the connected ports and transformation properties. You cannot edit an Application Multi-Group Source Qualifier transformation.

CHAPTER 5

Working with webMethods Targets

This chapter includes the following topics:

- [Working with webMethods Targets Overview, 23](#)
- [Importing a webMethods Target Definition, 24](#)
- [Editing webMethods Target Definitions, 25](#)

Working with webMethods Targets Overview

Import a webMethods target definition or create one from a webMethods source definition.

Working with the pubSeqn Envelope Field

To write target webMethods documents with a publication sequence number, pass a value to the pubSeqn envelope field to the webMethods target. Also, ensure that the PowerCenter Integration Service writes documents to the target according to the sequence number in the pubSeqn field. The webMethods Broker does not accept documents out of sequence.

Maintaining Transactional Consistency for webMethods Targets

The PowerCenter Integration Service can maintain transactional consistency for webMethods targets. The PowerCenter Integration Service commits messages to webMethods targets after it writes all documents in a transaction group to the targets.

A transaction group consists of all documents that the PowerCenter Integration Service commits when it reaches a commit point. If a session aborts or fails during a transaction, the PowerCenter Integration Service rolls back all documents in the transaction group from the targets. When you enable recovery and a session aborts or fails, you can restart the session in recovery mode. The PowerCenter Integration Service then loads the documents to the targets.

To ensure transactional consistency, all webMethods targets in the same pipeline must belong to the same target connection group. Targets in the same target connection group receive data from the same transactional source. Also, two or more webMethods targets are in the same target connection group if they have the same value for each of the following connection properties:

- Broker Host

- Broker Name
- Client ID
- Client Group
- Application Name
- Automatic Reconnection
- Preserve Client State

Importing a webMethods Target Definition

1. In the Target Designer, click **Targets > Import from webMethods Broker**.
The **Import from webMethods Broker** dialog box appears.
2. Enter the following information:

Import Attribute	Description
Broker Host	Host name and port number of the Broker in the following format: <code><host_name:port></code> You can omit the port number if the port number for the Broker is the default port number. The default port number is 6849.
Broker Name	Name of the Broker.
Client Group	Client group where the document metadata is stored.
Scope Name	Document folder name that displays the document types in the document folder. Document types that belong to subfolders of the document folder you enter for the scope name do not appear.

3. Click **Connect**.
The Designer displays a list of document types and their document folders.
4. Select the document type you want to import.
5. Optionally, select Create Default Envelope Fields to add the default envelope fields to the target definition.
6. Click **Import**.
Note: webMethods document types are case sensitive. However, PowerCenter target definition names are not. Therefore, when you import a webMethods target definition, and the repository already contains a target definition with the same name that uses different capitalization, the Designer might ask if you want to overwrite the existing target definition.

Editing webMethods Target Definitions

You can edit webMethods target definitions to change the precision of certain fields, modify the name and description of target definition, and change column names. When you change column names, the Designer does not update the map value to reflect the changes.

To update a target definition to reflect changes in the webMethods document metadata, recreate the target definition.

After you add a webMethods target definition to a mapping, you can configure the load scope to determine how the PowerCenter Integration Service writes documents to the target. You can generate document data after each row, after each transaction, or at the end of file.

1. Double-click the title bar of the target definition in the Mapping Designer.
2. Click the **Properties** tab.
3. Select a value for Load Scope:
 - Row. Generates document data for each row.
 - Transaction. Generates document data for all fields in a transaction on commit.
 - All Input. Generates document data at end of file. Default is All Input.
4. Click **OK**.

CHAPTER 6

Creating and Configuring webMethods Workflows

This chapter includes the following topics:

- [Creating and Configuring webMethods Workflows Overview, 26](#)
- [Configuring a Session with a webMethods Mapping, 30](#)
- [Configuring and Scheduling Workflows, 33](#)
- [Troubleshooting webMethods Workflows, 33](#)

Creating and Configuring webMethods Workflows Overview

When you configure a webMethods workflow, you define the session and scheduler properties that determine how the PowerCenter Integration Service reads documents from a webMethods source or writes documents to a webMethods target. You can configure the following webMethods session and workflow properties:

- Delivery model for sources
- Document filtering
- Terminating conditions
- Real-time data extraction
- Message recovery
- Delivery model for targets
- Sequence support
- Transactional consistency for targets
- Pipeline partitioning

Configuring the Delivery Model for webMethods Sources

When the PowerCenter Integration Service reads documents from a webMethods source, it uses the publish/subscribe model by default. It reads documents in both deliver/receive mode and broadcast mode.

You can configure the session to only read source documents in deliver/receive mode. The PowerCenter Integration Service only receives documents that have a particular client ID.

You can specify the client ID in one of the following ways:

- Enter the client ID in the webMethods application connection properties.
- The webMethods Broker can generate a client ID when you run the session.

Note: If you want to run simultaneous sessions that read documents from webMethods sources, enter a separate client ID for each source.

To configure the delivery model for webMethods sources, select one of the following options:

- Select the Subscribe Document option for the PowerCenter Integration Service to subscribe to source documents using the publish/subscribe model. When the Subscribe Document option is selected, the PowerCenter Integration Service reads documents using the publish/subscribe model. It reads documents in both deliver/receive mode and broadcast mode.
- Clear the Subscribe Document option for the PowerCenter Integration Service to receive source documents in deliver/receive mode only.

Filtering Source Documents

You can filter documents the PowerCenter Integration Service reads with the publish/subscribe model in deliver/receive and broadcast modes. You cannot filter source documents in deliver/receive mode only.

Filter documents by document data fields. The Broker only makes documents available for the PowerCenter Integration Service that meet the filter condition.

For example, you want to read webMethods documents of the document type TRADE::BUY. The document type contains a data field SYMBOL. When the PowerCenter Integration Service reads source documents, you want the webMethods Broker to only make documents available that contain the value INFA in the data field SYMBOL. Enter a filter condition to receive documents of the value INFA in the data field SYMBOL from source documents of type TRADE::BUY.

To filter documents from a webMethods source, complete the following tasks in the session properties:

1. Select Subscribe Document option in the Properties settings on the Mapping tab (Sources node).
2. Enter a filter condition for the Subscription Filter Expression attribute in the Properties settings on the Mapping tab (Sources node).

Specify a filter condition according to the syntax in the webMethods documentation. If you do not use webMethods syntax when entering a filter condition, the PowerCenter Integration Service fails the session during session initialization. For example, you can enter the following filter condition to read documents with the value INFA in the data field SYMBOL:

```
SYMBOL = "INFA"
```

If you modified a field name in the source definition, make sure the filter condition uses the map value for the field, not the modified field name. For example, you imported a source definition with the data field NAME. You changed the field in the source definition to CUSTOMER_NAME. If you enter a filter condition for this field, use the value NAME instead of CUSTOMER_NAME according to the map value for the field in the source definition.

Configuring Terminating Conditions

Terminating conditions determine when the 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

Idle time, message count, and reader time limit control when the Integration Service stops reading from the webMethods Broker.

When you enter values for multiple terminating conditions, the Integration Service stops reading from the webMethods Broker when the first terminating condition is met. For example, if you set idle time to 10 seconds and message count to 100 documents, the Integration Service stops reading from the webMethods Broker after 10 seconds or after reading 100 documents, whichever comes first.

Idle Time

Configure idle time to indicate how many seconds the PowerCenter Integration Service waits when no documents arrive before it stops reading from the webMethods Broker. For example, if you enter 30 for idle time, the PowerCenter Integration Service waits 30 seconds after reading from the webMethods Broker. If no new documents arrive in the webMethods Broker within 30 seconds, the PowerCenter Integration Service stops reading from the webMethods Broker.

Message Count

Configure message count to control the number of documents the PowerCenter Integration Service reads from the webMethods Broker before stopping. For example, when you specify 100 for message count, the PowerCenter Integration Service reads 100 documents from the webMethods Broker.

If you enter a message count value, and you configure the session to use pipeline partitioning, the session can run on a single node only. The PowerCenter Integration Service that runs the session cannot run on a grid or on primary and backup nodes.

Reader Time Limit

Configure reader time limit to read documents from the webMethods Broker for a set period of time in seconds. The PowerCenter Integration Service reads documents from the webMethods Broker for the number of seconds you specify. For example, if you specify 10 for reader time limit, the PowerCenter Integration Service stops reading from the webMethods Broker 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 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 in a recovery file. If the session fails, run the session in recovery mode to recover the messages the PowerCenter Integration Service did not process.

Configuring the Delivery Model for webMethods Targets

By default the PowerCenter Integration Service writes webMethods documents to targets with the publish/subscribe model in broadcast mode. You can configure document envelope fields in the session properties to write webMethods target documents in the following ways:

- Deliver/receive mode

- Request/reply model

Configure the envelope fields in the session properties to change the delivery model when the mapping does not pass data to the corresponding envelope fields in the webMethods target definition. For example, you want to write documents to a webMethods target using the request/reply model. The mapping contains a flat file source definition. The mapping transforms the source data and writes the data to data fields in the webMethods target. However, it does not pass data to the envelope fields. You can configure the session properties to pass a value to the replyTo, errorsTo, and errorRequestsTo envelope fields in the target documents to write documents in deliver/receive mode.

The following table describes the envelope fields you can configure in the session properties to set the delivery model:

Properties Settings	Delivery Mode	Description
Destination Client ID	Deliver/Receive	Client ID with which you want the PowerCenter Integration Service to deliver target documents. The Broker Client that receives the target documents must use this client ID to receive the documents.
Reply To	Request/Reply	Client ID of the webMethods target to whom you want to send target documents that are replies. The Reply To client ID should not be the originator of the document.
Errors To	Request/Reply	Client ID of the webMethods target to whom you want to send target documents if there are any errors processing a request document.
Error Requests To	Request/Reply	If there is an error processing the request document, enter another destination for the request.

If the mapping passes values to the envelope 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.

Reading and Writing Sequences in webMethods Documents

You can configure how the PowerCenter Integration Service reads and writes webMethods documents that contain sequences.

The following table describes the options you can configure in the session properties to define how the PowerCenter Integration Service reads and writes webMethods sequences:

Properties Settings	Description
Orphan Handling	Specifies how to handle the error if sequence group data does not correspond to parent group data.
Duplicate Handling	Specifies how to handle the error if two or more rows contain the same primary key value.
Process Messages on Transaction	Specifies whether documents are flushed when a transaction is committed.

Configuring Transactional Consistency for webMethods Targets

The PowerCenter Integration Service can maintain transactional consistency for sessions with webMethods sources and targets. With transactional consistency, the PowerCenter Integration Service commits documents to webMethods 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 webMethods target in the pipeline. Using the same connection properties ensures that all webMethods 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

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 webMethods mappings:

Partition Point	Partition Type
Application Multi-Group Source Qualifier	Pass-through
webMethods target	All

Specifying Partitions and a Recovery Cache Folder

When you specify partitions for a webMethods mapping in a session, and you configure the Recovery Cache Folder attribute in the session properties, enter a cache folder on a different physical drive for each reader partition in the pipeline for better performance.

Configuring a Session with a webMethods Mapping

When creating a session with a webMethods source or target, configure the following session properties.

Properties Tab - General Options

From the General Options on the Properties tab, configure the following options:

1. Select Source as the commit type to extract data in real time and ensure transactional consistency.
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 Mapping tab (Sources node). 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

If you are configuring session properties for a webMethods source, configure the following session properties on the Sources node of the Mapping tab:

1. In the Connections settings, select a webMethods application connection for Application Multi-Group Source Qualifiers connected to webMethods source definitions.

Select a webMethods Integration Server connection to read webMethods documents with special characters.

Select a webMethods broker connection to read webMethods documents without special characters.

2. From the Properties settings, clear Subscribe Documents if you want the PowerCenter Integration Service to only receive webMethods documents in deliver/receive mode from the source.

If you want the PowerCenter Integration Service to subscribe to webMethods documents with both the broadcast and deliver/receive modes using the publish/subscribe model, make sure Subscribe Documents is selected.

3. Optionally, enter a filter condition in the Subscription Filter Expression option if you want to filter webMethods source documents.

4. Optionally, edit the values for the Idle Time, Message Count, or Reader Time Limit terminating conditions. Applicable when you select webMethods broker connection.

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

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

5. Optionally, configure real-time flush latency.

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

6. If you enabled message recovery, you can enter a value for the recovery cache folder. Or, use the default cache folder \$PMCacheDir\.

Mapping Tab - Targets Node

When creating a session that uses a webMethods target, configure the following session properties on the Targets node of the Mapping tab:

1. In the Connections settings, select a webMethods application connection for each webMethods target definition.

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

2. In the Properties settings, enter the following properties:

Properties Settings	Description
Destination Client ID	Client ID with which you want the PowerCenter Integration Service to deliver target documents. The Broker Client that receives the target documents must use this client ID to receive the documents. Use this option to send target documents in deliver/receive mode.
Reply To	Client ID of the webMethods target to whom you want to send target documents that are replies. The Reply To client ID cannot be the originator of the document. Use this option to send target documents using the request/reply delivery model.
Errors To	Client ID of the webMethods target to whom you want to send target documents if there are any errors processing a request document. Use this option to send target documents using the request/reply delivery model.
Error Requests To	If there is an error processing the request document, this field states another destination for the request. Use this option to send target documents using the request/reply delivery model.
Send Interval	<p>Minimum amount of time in milliseconds the Integration Service waits between each document when writing documents to the target. For example, if you set the send interval to 2, the PowerCenter Integration Service waits at least two milliseconds before writing documents to the target. The targets receive the documents once the PowerCenter Integration Service issues a commit.</p> <p>If you stop the session during the session run, the PowerCenter Integration Service waits until it writes the document it is processing to the target before stopping the session. Therefore, if you set a high value for Send Interval, the session may take longer to stop. Default is 1.</p>
Orphan Handling	Select Ignore to ignore errors when sequence group data does not correspond to parent group data. Otherwise, select Error.
Duplicate Handling	Select Error, First, or Last.
Process Messages on Transaction	Specifies whether documents are flushed when a transaction is committed.

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

- Destination Client ID
- Reply To
- Errors To
- Error Requests To

Configuring and Scheduling Workflows

Configuring Workflows to Run Simultaneous Sessions

When you configure a workflow to run simultaneous sessions to read documents from webMethods sources or write documents to webMethods targets, make sure that each source or target that shares the same client state uses a different client ID. Otherwise, the session may not process the expected data.

To ensure that each source uses a different client ID, create multiple application connections with different client IDs. Assign each Application Multi-Group Source Qualifier associated with a webMethods source a different webMethods application connection.

To ensure that each target uses a different client ID, create multiple application connections with different client IDs. Assign each target a different application connection.

Scheduling Workflows

Before you run a session, configure and schedule the workflow. 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.

Troubleshooting webMethods Workflows

I ran a session to write documents to webMethods. The session log reported that the PowerCenter Integration Service wrote all of the documents to the target successfully. However, the Broker Client did not receive all of the documents.

When the PowerCenter Integration Service wrote documents to the target, the documents were not in sequence according to the pubSeqn envelope field. The webMethods Broker only accepts documents in sequence.

I ran simultaneous sessions to read documents from webMethods sources. The session completed successfully. However, the PowerCenter Integration Service did not receive all of the documents.

If you run simultaneous sessions to read documents from webMethods sources, each source must use a different client ID.

A source or target document is missing after running a session.

Make sure that the document type of the source or target document matches the document type of the source or target definition. If these document types do not match, the PowerCenter Integration Service rejects the document, and the document is no longer available at the Broker. You can view the content of a rejected document in the session log file.

APPENDIX A

webMethods Datatype Reference

This appendix includes the following topic:

- [webMethods and Transformation Datatypes, 34](#)

webMethods and Transformation Datatypes

PowerCenter uses the following datatypes in webMethods mappings:

- webMethods datatypes. webMethods datatypes appear in webMethods 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 webMethods datatypes that PowerCenter supports and the corresponding transformation datatypes:

webMethods Datatype	Range and Description	Transformation datatype	Range and Description
Boolean	Precision of 1 and scale of 0.	String	1 to 104,857,600 characters
Byte	Precision of 3 and scale of 0.	Small Integer	Precision 5, scale 0
Byte_Seqn	Precision of 10 and scale of 0.	Binary	1 to 104,857,600 bytes
Char	Precision of 1 and scale of 0.	Double	Precision 15
Char_Seqn	Precision of 1 and scale of 0.	String	1 to 104,857,600 characters
Date	Precision of 23.	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
Double	Precision of 15 and scale of 0.	Double	Precision 15
Float	Precision of 10 and scale of 0.	Double	Precision 15

webMethods Datatype	Range and Description	Transformation datatype	Range and Description
Int	Precision of 10 and scale of 0.	Integer	Precision 10, scale 0
Long	Precision of 19 and scale of 0.	Decimal	Precision 1 to 28 digits, scale 0 to 28
Short	Precision of 5 and scale of 0.	Small Integer	Precision 5, scale 0
String	Precision of 1 to 65,535 and scale of 0.	String	1 to 104,857,600 characters
Unicode_Char	Precision of 1 and scale of 0.	String	1 to 104,857,600 characters
Unicode_String	Precision of 1 to 65,535 and scale of 0.	String	1 to 104,857,600 characters

Boolean Datatype

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

When the PowerCenter Integration Service writes data to a webMethods 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, it rejects the field.

APPENDIX B

Glossary

broadcast mode

A delivery mode in which the sender sends documents to a Broker for any Broker Client that wants to subscribe to the documents.

Broker Client

A client to connect to the webMethods Broker.

client group

A component of the webMethods Broker that controls access to document types. A client group can contain document folders.

data field

A field that contains message data.

deliver/receive mode

Point-to-point delivery of documents by client ID. Only a Broker Client with a certain client ID can receive the documents.

document

A webMethods message.

document folder

A folder in the webMethods Broker that stores other document folders and document types.

document type

A representation of the structure of documents. Document types contain data field names and datatypes for a webMethods document.

envelope field

A header field of a webMethods document.

map value

A map of field relationships in the document type hierarchy.

publish/subscribe model

A delivery model for documents the sender wants to publish or receive that do not require a response. The publish/subscribe model enables you to read and write documents in deliver/receive or broadcast mode.

request/reply model

A delivery model for documents that require a reply.

sequence field

A field of a webMethods document that contains a sequence of multiple values of the same datatype.

structure

A collection of fields in a webMethods document that defines a webMethods data field.

webMethods Broker

A webMethods application that stores metadata for webMethods documents.

webMethods Enterprise Integrator

A webMethods tool where you define webMethods documents.

INDEX

A

Application Multi-Group Source Qualifier
 creating for webMethods [22](#)
 description for webMethods [22](#)

C

client ID
 simultaneous sessions in webMethods, setting [33](#)
continuous workflows
 description for webMethods real-time flush latency [33](#)

D

datatypes
 Boolean, used in webMethods [35](#)
 PowerExchange for webMethods [34](#)
deliver/receive mode
 webMethods sources, configuring [26](#)
 webMethods targets, configuring [28](#)
delivery models
 webMethods sources, configuring [26](#)
 webMethods targets, configuring [28](#)

G

grid
 restriction in webMethods [28](#)

H

high availability
 restriction in a webMethods session [28](#)

I

idle time
 description for webMethods [28](#)
installing
 webMethods [12](#)
 webMethods installation and configuration steps [12](#)

J

JVM options
 configuring for webMethods [13](#)

L

Load Scope (property)
 setting for webMethods targets [25](#)

M

map value
 using with a filter condition in webMethods session [27](#)
message count
 description for webMethods [28](#)
 high availability in a webMethods session [28](#)
message recovery
 description for webMethods [28](#)
 partitions and a file cache folder from the webMethods source [30](#)

P

pipeline partitioning
 description for webMethods [30](#)
 specifying partitions and a file cache folder in webMethods [30](#)
publish/subscribe delivery model
 webMethods sources, configuring [26](#)
 webMethods targets, configuring [28](#)
pubSeqn envelope fields
 in a webMethods target definition [23](#)

R

reader time limit
 description for webMethods [28](#)
real-time flush latency
 source-based commit for a webMethods session [30](#)
real-time sessions
 description for webMethods [28](#)

S

sessions
 configuring for webMethods [30](#)
source documents
 filtering in a webMethods session [27](#)
source-based commit
 configuring for a webMethods session [30](#)

T

targets
 transactional consistency for webMethods sessions [30](#)
terminating conditions
 description for webMethods [27](#)

- terminating conditions (*continued*)
 - entering for a webMethods session [27](#)
 - reader time limit for webMethods [28](#)
 - webMethods idle time [28](#)
 - webMethods message count [28](#)
 - webMethods real-time flush latency [28](#)
- troubleshooting
 - webMethods workflows [33](#)

W

- webMethods
 - installing [12](#)
- webMethods source definition
 - creating [20](#)

- webMethods source definition (*continued*)
 - editing [21](#), [25](#)
 - overview [20](#), [23](#)
- webMethods sources
 - simultaneous sessions, running [33](#)
- webMethods target definition
 - creating [24](#)
 - editing [21](#), [25](#)
 - overview [20](#), [23](#)
- webMethods targets
 - simultaneous sessions, running [33](#)
- workflows
 - configuring a webMethods session [30](#)
 - scheduling a webMethods session [33](#)
 - troubleshooting [33](#)