



Informatica® PowerExchange for Salesforce 10.2

User Guide

© Copyright Informatica LLC 2014, 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, and PowerExchange are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Consortium/Legal/2002/copyright-software-20021231; <http://www.slf4j.org/license.html>; <http://nanoxml.sourceforge.net/orig/copyright.html>; <http://www.json.org/license.html>; <http://forge.ow2.org/projects/javaservice/>; <http://www.postgresql.org/about/license.html>; <http://www.sqlite.org/copyright.html>; <http://www.tcl.tk/software/tcltk/license.html>; <http://www.jaxen.org/faq.html>; <http://www.jdom.org/docs/faq.html>; <http://www.slf4j.org/license.html>; <http://www.iodbc.org/dataspace/iodbc/wiki/IODBC/License>; <http://www.keplerproject.org/md5/license.html>; <http://www.toedter.com/en/jcalendar/license.html>; <http://www.edankert.com/bounce/index.html>; <http://www.net-snmp.org/about/license.html>; <http://www.openmdx.org/#FAQ>; http://www.php.net/license/3_01.txt; <http://srp.stanford.edu/license.txt>; <http://www.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 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: 2018-09-25

Table of Contents

| | |
|---|-----------|
| Preface | 6 |
| Informatica Resources. | 6 |
| Informatica Network. | 6 |
| Informatica Knowledge Base. | 6 |
| Informatica Documentation. | 7 |
| Informatica Product Availability Matrixes. | 7 |
| Informatica Velocity. | 7 |
| Informatica Marketplace. | 7 |
| Informatica Global Customer Support. | 7 |
| Chapter 1: Introduction to PowerExchange for Salesforce..... | 8 |
| PowerExchange for Salesforce Overview. | 8 |
| Example of Data Migration from Salesforce. | 9 |
| Example of Updating Real-time Data to Salesforce. | 9 |
| Chapter 2: PowerExchange for Salesforce Configuration..... | 10 |
| PowerExchange for Salesforce Configuration Overview. | 10 |
| Configuring HTTP Proxy Options at Design Time. | 10 |
| Configuring HTTP Proxy Options at Run Time. | 11 |
| Chapter 3: Salesforce Connections..... | 12 |
| Salesforce Connection Overview. | 12 |
| Salesforce Connection Properties. | 12 |
| Creating a Salesforce Connection in the Administrator Tool. | 13 |
| Creating a Salesforce Connection in the Developer Tool. | 14 |
| Chapter 4: Salesforce Data Objects..... | 15 |
| Salesforce Data Objects Overview. | 15 |
| Standard and Custom Salesforce Objects. | 15 |
| Related Objects. | 16 |
| Rules and Guidelines for Related Objects. | 16 |
| Salesforce Data Object Views. | 16 |
| Salesforce Data Object Overview Properties. | 17 |
| Salesforce Data Object Read Operation Properties. | 17 |
| Source Properties of the Data Object Read Operation. | 17 |
| Output Properties of the Data Object Read Operation. | 19 |
| Salesforce Data Object Write Operation Properties. | 21 |
| Input Properties of the Data Object Write Operation. | 21 |
| Target Properties of the Data Object Write Operation. | 24 |
| Importing a Salesforce Data Object. | 24 |

| | |
|--|-----------|
| Creating a Salesforce Data Object Read or Write Operation. | 25 |
| Chapter 5: Salesforce Mappings..... | 26 |
| Salesforce Mappings Overview. | 26 |
| Salesforce Mapping Read Example. | 26 |
| Salesforce Mapping Write Example. | 27 |
| Chapter 6: Salesforce Run Time Processing..... | 28 |
| Salesforce Run-time Processing Overview. | 28 |
| Filtering Source Data by Using the SOQL Filter Condition. | 28 |
| Capturing Deleted and Archived Salesforce Records. | 29 |
| Enable Bulk Query. | 29 |
| Use SFDC Bulk API. | 29 |
| Configuring the Upsert Target Operation. | 30 |
| Configuring the Maximum Batch Size. | 30 |
| Handling Null Values in Update and Upsert Operations. | 31 |
| Override an External ID with an idLookup for Upserts. | 31 |
| Appendix A: Data Type Reference..... | 32 |
| Data Type Reference Overview. | 32 |
| Salesforce Data Types and Transformation Data Types. | 32 |
| Index..... | 34 |

Preface

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

This book assumes that you have knowledge of web services concepts, relational database concepts, PowerExchange, and Salesforce. You must also be familiar with the interface requirements for other supporting applications. For more information about related Salesforce issues, see the Salesforce documentation.

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

Introduction to PowerExchange for Salesforce

This chapter includes the following topics:

- [PowerExchange for Salesforce Overview, 8](#)
- [Example of Data Migration from Salesforce, 9](#)
- [Example of Updating Real-time Data to Salesforce, 9](#)

PowerExchange for Salesforce Overview

PowerExchange for Salesforce provides connectivity between Informatica Developer and Salesforce. You can use PowerExchange for Salesforce to read data from and write data to Salesforce. You can add a Salesforce data object operation as a source or a target in a mapping and run the mapping to read or write data.

Salesforce sources and targets represent objects in the Salesforce object model. Salesforce objects are tables that correspond to tabs and other user interface elements on the Salesforce website. For example, the Account object contains the information that appears in fields on the Salesforce Account tab. You can view, create, update, and delete data in Salesforce objects.

PowerExchange for Salesforce uses the Salesforce security model to enforce data access controls. You can access data based on the Salesforce organization associated with the user account you use to connect to Salesforce. Your access to data also depends on the user privileges and the field-level and row-level permissions associated with the login.

PowerExchange for Salesforce uses the Simple Object Access Protocol API (SOAP API) to read or write a small volume of data in near real-time mode. PowerExchange for Salesforce uses the Salesforce Bulk API to read large amounts of data from Salesforce sources or write large amounts of data to Salesforce targets. PowerExchange for Salesforce generates a Salesforce Object Query Language (SOQL) query to read data from Salesforce objects. SOQL is a derivative of SQL.

You can run a profile on a Salesforce data object. A Salesforce data object profile discovers information about the column data and metadata in the Salesforce data source.

PowerExchange for Salesforce is listed under the Cloud connection category in the Developer tool and the Administrator tool.

PowerExchange for Salesforce supports Salesforce API up to version 39 and connections to Salesforce API up to version 39.

Example of Data Migration from Salesforce

Your organization needs to migrate sales opportunity information from a Salesforce system that the sales team uses to a relational data source that the executive management team uses. You can create a data object in the Model repository and import the Opportunity object. The executive management team can reconcile and analyze the data written to the relational data object.

Example of Updating Real-time Data to Salesforce

Your organization needs to update real-time sales order processing status from an enterprise resource planning (ERP) system that the logistics team uses to a Salesforce system that was used to create the order. You can create a data object and specify the update strategy. You can then create a mapping that reads shipping details from the ERP system and writes those records to the Salesforce data objects. Sales managers can use the updated information to track sales orders.

CHAPTER 2

PowerExchange for Salesforce Configuration

This chapter includes the following topic:

- [PowerExchange for Salesforce Configuration Overview, 10](#)

PowerExchange for Salesforce Configuration Overview

If your organization uses a proxy server to access the internet, you can configure the HTTP proxy server authentication settings at design time and at run time.

Configuring HTTP Proxy Options at Design Time

If your organization uses a proxy server to access the Internet, you can configure the HTTP proxy server authentication settings from the `developerCore.ini` file.

Perform the following tasks to configure the HTTP Proxy Options from the Developer tool:

- Ensure that you enable the proxy server settings from your web browser.
- Access the `developerCore.ini` file at the following location:
`<Informatica Installation Location>\clients\DeveloperClient`
- Add the HTTP Proxy options to the `developerCore.ini` file.

The following table describes the properties that you must add to the `developerCore.ini` file:

| Property | Description |
|--------------------------------|---|
| <code>-Dhttp.proxyHost=</code> | Name of the HTTP proxy server. |
| <code>-Dhttp.proxyPort=</code> | Port number of the HTTP proxy server. |
| <code>-Dhttp.proxyUser=</code> | Authenticated user name for the HTTP proxy server. This property is required if the proxy server requires authentication. |

| Property | Description |
|-----------------------|--|
| -Dhttp.proxyPassword= | Password for the authenticated user. This property is required if the proxy server requires authentication. Note: The password is in plaintext and not encrypted. |
| -Dhttp.nonProxyHosts= | List of host names or IP addresses for which you must not use the proxy server. Separate the list of IP addresses or host names with a pipe symbol (). For example, <code>localhost:10.20.30.40 myHost</code> Specify the IP address or name of the machine on which the Informatica gateway node runs so that the Developer tool connects to the domain. |

Configuring HTTP Proxy Options at Run Time

If your organization uses a proxy server to access the internet, you must configure the HTTP proxy server authentication settings for the Data Integration Service.

1. Open the Administrator tool.
2. Click the **Administration** tab, and then select the **Data Integration Service**.
3. Click the **Properties** tab.
4. Click **Edit** in the HTTP Proxy Server Properties section.
5. Configure the following properties:

| Property | Description |
|----------------------------|---|
| HTTP Proxy Server Host | Name of the HTTP proxy server. |
| HTTP Proxy Server Port | Port number of the HTTP proxy server. Default is 8080. |
| HTTP Proxy Server User | Authenticated user name for the HTTP proxy server. This property is required if the proxy server requires authentication. |
| HTTP Proxy Server Password | Password for the authenticated user. This property is required if the proxy server requires authentication. |
| HTTP Proxy Server Domain | Domain for authentication. |

CHAPTER 3

Salesforce Connections

This chapter includes the following topics:

- [Salesforce Connection Overview, 12](#)
- [Salesforce Connection Properties, 12](#)
- [Creating a Salesforce Connection in the Administrator Tool, 13](#)
- [Creating a Salesforce Connection in the Developer Tool, 14](#)

Salesforce Connection Overview

Use a Salesforce connection to access objects in a Salesforce application.

Create a connection to import Salesforce metadata to create data objects, preview data, and run mappings.

You can create a Salesforce connection in the Developer tool, the Administrator tool, and through infacmd isp.

Salesforce Connection Properties

Use a Salesforce connection to connect to a Salesforce object.

The following table describes the Salesforce connection properties:

| Property | Description |
|-------------|--|
| Name | The name of the connection. The name is not case sensitive and must be unique within the domain. It cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? / |
| ID | The string that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name. |
| Description | The description of the connection. The description cannot exceed 765 characters. |

| Property | Description |
|---------------|--|
| Location | The Informatica domain where you want to create the connection. |
| Type | The connection type. Select Salesforce. |
| User Name | The Salesforce user name. |
| User Password | <p>The password for the Salesforce user name.</p> <p>To access Salesforce outside your organization's trusted networks, you must append a security token to your password to log in to the API or a desktop client.</p> <p>To receive or reset your security token, log in to Salesforce and click Setup My Personal Information Reset My Security Token.</p> <p>Password is case sensitive.</p> |
| Service URL | The URL of the Salesforce service you want to access. In a test or development environment, you might want to access the Salesforce Sandbox testing environment. For more information about the Salesforce Sandbox, see the Salesforce documentation. |

Creating a Salesforce Connection in the Administrator Tool

Create a connection before you import Salesforce data objects, preview data, or run mappings. When you create a Salesforce connection, you enter information such as a connection ID and the URL of the Salesforce service you want to access.

1. In the Administrator tool, click the **Domain** tab.
2. Click the **Connections** view.
3. In the Navigator, select the domain.
4. In the Navigator, click **Actions > New > Connection**.
The **New Connection** dialog box appears.
5. In the **New Connection** dialog box, select **Cloud > Salesforce**, and then click **OK**.
The **New Connection** wizard appears.
6. Enter a connection name.
7. Enter an ID for the connection.
8. Optionally, enter a connection description.
9. Enter the connection properties.
10. Click **Test Connection** to verify that you can connect to Salesforce.
11. Click **Finish**.

Creating a Salesforce Connection in the Developer Tool

Create a connection before you import Salesforce data objects, preview data, or run mappings. When you create a Salesforce connection, you enter information such as a connection ID and the URL of the Salesforce service you want to access.

1. Click **Window > Preferences**.
2. Select **Informatica > Connections**.
3. Expand the domain.
4. Select **Cloud > Salesforce** and click **Add**.
5. Enter a connection name.
6. Enter an ID for the connection.
7. Optionally, enter a connection description.
8. Select the domain where you want to create the connection.
9. Select **Salesforce** as the connection type.
10. Click **Next**.
11. Configure the connection properties.
12. Click **Test Connection** to verify that you can connect to the Salesforce system.
13. Click **Finish**.

CHAPTER 4

Salesforce Data Objects

This chapter includes the following topics:

- [Salesforce Data Objects Overview, 15](#)
- [Standard and Custom Salesforce Objects, 15](#)
- [Related Objects, 16](#)
- [Salesforce Data Object Views, 16](#)
- [Salesforce Data Object Overview Properties, 17](#)
- [Salesforce Data Object Read Operation Properties, 17](#)
- [Salesforce Data Object Write Operation Properties, 21](#)
- [Importing a Salesforce Data Object, 24](#)
- [Creating a Salesforce Data Object Read or Write Operation, 25](#)

Salesforce Data Objects Overview

A Salesforce data object is a physical data object that uses a Salesforce object as a source and a target. A Salesforce data object is a representation of data that is based on a Salesforce object.

Import a Salesforce object into the Developer tool to create a Salesforce data object. After you create a data object, create a data object read or write operation. You can use the data object read operation as a source and the data object write operation as a target in a mapping.

Standard and Custom Salesforce Objects

Use the Developer tool to import Salesforce objects and create a Salesforce data object. You can import both standard and custom Salesforce objects.

Standard object types are objects packaged within Salesforce, such as Account, AccountPartner, and Opportunity.

Custom object types extend the Salesforce data for an organization by defining data entities that are unique to the organization. Salesforce administrators can define custom fields for both standard and custom objects.

When you import a Salesforce object, use a Salesforce login to connect to the Salesforce service. The Developer tool generates a list of objects that are available for import.

Related Objects

You might need to read data from more than one object at a time. The Data Integration Service generates relationship queries through SOQL to read data from related objects.

For example, you can read all accounts created by Tom Smith and the contacts associated with those accounts. You can use PowerExchange for Salesforce to create parent-to-child relationships that connect the objects.

Parent-to-child relationships exist between many types of objects. For example, Account is a parent of Contact, Assets, and Cases.

Use PowerExchange for Salesforce to read related objects. Each object can have one related object. For example, you can create a data object called Account Details. Select Account as the parent object, and either Contact or Opportunity as the child object. The relationship persists while creating a Salesforce data object read operation from the Salesforce data object called Account Details.

Rules and Guidelines for Related Objects

Consider the following rules and guidelines when you import related objects in a Salesforce data object:

- You must select a parent object to create a data object that has related objects.
- You cannot import multiple parent objects in a single data object.
- You can select one related object for each parent object.
- You cannot read data from a related object while using Bulk API. You can read data from one parent object.
- The Relationship dialog box does not display all related entities of the Salesforce object.

Salesforce Data Object Views

The Salesforce data object contains views to edit the object name and the properties.

After you create a Salesforce data object, you can change the data object properties in the following data object views:

- **Overview** view. Edit the Salesforce data object name, description, and object.
- **Data Object Operation** view. View and edit the properties that the Data Integration Service uses when it reads data from or writes data to a Salesforce data object.

When you create a mapping that uses a Salesforce source, you can view the data object read properties in the **Properties** view.

When you create a mapping that uses a Salesforce target, you can view the data object write properties in the **Properties** view.

Salesforce Data Object Overview Properties

The **Overview** view displays general information about the Salesforce data object and detailed information about the Salesforce object that you imported.

The following table describes the general properties that you configure for a Salesforce data object:

| Property | Description |
|-------------|--|
| Name | Name of the Salesforce data object. |
| Description | Description of the Salesforce data object. |
| Connection | Name of the Salesforce connection. |

The following table describes the Salesforce object properties that you can view:

| Property | Description |
|-------------|--|
| Name | Name of the Salesforce object. |
| Type | Native data type of the Salesforce object. |
| Description | Description of the Salesforce object. |

Salesforce Data Object Read Operation Properties

The Data Integration Service reads data from a Salesforce object based on the data object read operation. The Developer tool displays the data object read operation properties of the Salesforce data object in the **Data Object Operation** view.

You can view or configure the data object read operation from the source and output properties.

Source properties

Represents data that the Data Integration Service reads from the Salesforce object. Select the source properties to view data such as the name and description of the Salesforce object and the column properties.

Output properties

Represents data that the Data Integration Service passes into the mapping pipeline. Select the output properties to edit the port properties of the data object read operation. You can also set advanced properties, such as row limit and Salesforce bulk API.

Source Properties of the Data Object Read Operation

The source properties are populated based on the Salesforce object that you added when you created a data object. The source properties of the data object read operation include general and column properties that apply to the Salesforce object.

You can view the source properties of the data object read operation from the **General**, **Column**, and **Advanced** tabs.

General Properties

The general properties display the name and description of the data object read operation.

Column Properties

The column properties display the data types, precision, and scale of the source property in the data object read operation.

The following table describes the source column properties of the data object read operation:

| Property | Description |
|-------------|--|
| Name | Name of the column |
| Type | Native data type of the column |
| Precision | Maximum number of significant digits for numeric data types, or maximum number of characters for string data types. For numeric data types, precision includes scale. |
| Scale | Maximum number of digits after the decimal point for numeric values |
| Description | Description of the column |
| Creatable | Indicates whether the field allows inserts |
| Updateable | Indicates whether the field allows updates |
| ExternalID | Salesforce custom fields only. Indicates whether the field is designated as an external ID field. Each Salesforce object can contain a single custom field designated as the external ID field. Salesforce appends custom field names with "__c". For more information about external ID and custom fields, see the Salesforce documentation. |
| SforceName | Field name in Salesforce |
| ReferenceTo | Gets referenced object |
| IDLookup | Specifies a record in an upsert call. The ID field of each object and some Name fields have this property. There are exceptions, so use Salesforce to check for this property in any object that you want to upsert. |
| Filterable | Indicates whether the field can be used in the FROM or WHERE clause of an SOQL query. |
| Label | Field label in Salesforce |
| Access Type | Indicates whether the field has read and write permissions. |

Advanced Properties

The advanced properties display the physical name of the Salesforce object.

Output Properties of the Data Object Read Operation

The output properties represent data that the Data Integration Service passes into the mapping pipeline. Select the output properties to edit the port properties of the data object read operation.

The output properties of the data object read operation include general properties that apply to the data object operation. The output properties also include port, source, query, and advanced properties that apply to the Salesforce object.

You can view and change the output properties of the data object read operation from the **General**, **Ports**, **Sources**, **Query**, and **Advanced** tabs.

General Properties

The general properties display the name and description of the data object read operation.

Ports Properties

The output ports properties display the data types, precision, and scale of the data object read operation.

The following table describes the output ports properties that you configure in the data object read operation:

| Property | Description |
|-------------|---|
| Name | Name of the port. |
| Type | Data type of the port. |
| Precision | Maximum number of significant digits for numeric data types, or maximum number of characters for string data types. For numeric data types, precision includes scale. |
| Scale | Maximum number of digits after the decimal point for numeric values. |
| Description | Description of the port. |

Sources Properties

The sources properties list the Salesforce objects in the data object read operation.

Query Properties

Use the **Query** tab to specify a filter condition. You can specify only a platform filter expression. Platform expression uses Informatica transformation language. Use the platform filter expression to select specific records from Salesforce resources based on the filter condition that you specify.

The following table describes the properties that you can specify when you use the platform expression filter:

| Property | Description |
|-----------------|--|
| Expression Type | Type of filter expression that you want to use to filter records. Select Platform Expression . |
| Left Field | Column on which you want to apply the filter condition. |

| Property | Description |
|-------------|---|
| Operator | Simple operators that you can use to filter records. You can select one of the following operators: =, !=, <, <=, >, >= |
| Right Field | Value based on which you want to filter the records. You can also parameterize the value. |

Advanced Properties

Use the advanced properties to specify the data object read operation properties to read data from Salesforce objects.

The following table describes the advanced properties that you configure in the data object read operation:

| Property | Description |
|----------------------------|---|
| SOQL Filter Condition | Filters Salesforce source records. |
| CDC Time Limit | Time period, in seconds, that the Data Integration Service reads changed Salesforce data. When you set the CDC Time Limit to a non-zero value, the Data Integration Service performs a full initial read of the source data and then captures changes to the Salesforce data for the time period you specify. Default is 0. |
| Flush Interval | Interval, in seconds, at which the Data Integration Service captures changed Salesforce data. Default is 300. If you set the CDC Time Limit to a non-zero value, the agent captures changed data from the source every 300 seconds. Otherwise, the Data Integration Service ignores this value. |
| CDC Start Timestamp | Start date and time for the time period. The Data Integration Service extracts data added or modified after this time. Must be in the format YYYY-MM-DDTHH:MI:SS.SSSZ. |
| CDC End Timestamp | End date and time for the time period. The Data Integration Service extracts data added or modified before this time. Must be in the format YYYY-MM-DDTHH:MI:SS.SSSZ. |
| Row Limit | Specifies the maximum number of rows the Data Integration Service processes. Default is 0, which indicates that the Data Integration Services processes all records. |
| Use QueryAll | Runs a query that returns all rows, which includes active, archived, and deleted rows. Otherwise, the Data Integration Service returns active rows. |
| Use SystemModstamp for CDC | Uses the SystemModstamp as the time stamp for changed records in Salesforce. Otherwise, the Data Integration Service uses the LastModifiedDate time stamp to identify changed records in Salesforce. Default is to use the LastModifiedDate time stamp. |

| Property | Description |
|---------------------------|--|
| CDC Flush Interval Offset | The number of seconds that you want to offset the CDC flush interval. Captures real-time data that is submitted within the CDC time limit but not committed by Salesforce within the time limit. |
| Enable Bulk Query | Enable this feature to use the Salesforce Bulk API to read batch files containing large amounts of data. By default, the Data Integration Service uses the SOAP Salesforce API. |

Salesforce Data Object Write Operation Properties

The Data Integration Service writes data to a Salesforce object based on the data object write operation. The Developer tool displays the data object write operation properties for the Salesforce data object in the **Data Object Operation** section.

You can view the data object write operation from the Input and Target properties.

Input properties

Represent data that the Data Integration Service reads from an enterprise resource planning (ERP) system or a relational data object. Select the input properties to edit the port properties and specify the advanced properties of the data object write operation.

Target properties

Represent data that the Data Integration Service writes to Salesforce. Select the target properties to view data, such as the name, description, and the relationship of the Salesforce object.

Note: Information about rejected rows in a SOAP writer session is written to the Data Integration Service logs.

Input Properties of the Data Object Write Operation

Input properties represent data that the Data Integration Service reads from an enterprise resource planning (ERP) system or a relational data object. Select the input properties to edit the port properties of the data object write operation. You can also specify advanced data object write operation properties to write data to Salesforce objects.

The input properties of the data object write operation include general properties that apply to the data object write operation. They also include port, source, and advanced properties that apply to the data object write operation.

You can view and change the input properties of the data object write operation from the **General**, **Ports**, **Sources**, and **Advanced** tabs.

General Properties

The general properties list the name and description of the data object write operation.

Ports Properties

The input ports properties list the data types, precision, and scale of the data object write operation.

The following table describes the input ports properties that you must configure in the data object write operation:

| Property | Description |
|-------------|---|
| Name | Name of the port. |
| Type | Data type of the port. |
| Precision | Maximum number of significant digits for numeric data types, or maximum number of characters for string data types. For numeric data types, precision includes scale. |
| Scale | Maximum number of digits after the decimal point for numeric values. |
| Description | Description of the port. |

Sources Properties

The sources properties list the Salesforce objects in the data object write operation.

Advanced Properties

The advanced properties allow you to specify data object write operation properties to write data to Salesforce objects.

You can configure the following advanced properties in the data object write operation:

| Property | Description |
|------------------------|--|
| Treat Insert as Upsert | Upserts any record flagged as insert. By default, the Data Integration Service treats all records as insert. |
| Treat Update as Upsert | Upserts any record flagged as update. Select this property when you use the Update Strategy transformation in the mapping to flag records as update. |
| Max Batch Size | Maximum number of records the Data Integration Service writes to a Salesforce target in one batch. Default is 200 records. Not used in Bulk API target sessions. |
| Set Fields to NULL | Replaces values in the target with null values from the source. By default, the Data Integration Service does not replace values in a record with null values during an update or upsert operation. The Data Integration Service retains the existing values. |

| Property | Description |
|---|--|
| Use Idlookup Field for Upserts | Uses the Salesforce idLookup field to identify target records that need to be upserted. If you do not select this property, use an external ID for the upsert operation. If you do not select this property and do not provide an external ID, the session fails. |
| Use this ExternalId/ IdLookup field for Updates | The exact name of the external ID or idLookup field to use for updates. By default, the Data Integration Service uses the first external ID or idLookup field in the target. Use this property when you want to use a different field for updates. |
| Use SFDC Bulk API | Uses the Salesforce Bulk API to load batch files containing large amounts of data to Salesforce targets. By default, the Data Integration Service uses the standard Salesforce API. |
| Monitor Bulk Job Until All Batches Processed | Monitors a Bulk API target session. When you select this property, the Data Integration Service logs the status of each batch in the session log. If you do not select this property, the Data Integration Service does not generate complete session statistics for the session log. |
| Override Parallel Concurrency with Serial | Instructs the Salesforce Bulk API to write batches to targets serially. By default, the Bulk API writes batches in parallel. |
| Enable Field Truncation Attribute | Allows Salesforce to truncate target data that is larger than the target field. When you select this property, Salesforce truncates overflow data and writes the row to the Salesforce target. |
| Enable Hard Deletes for BULK API | Permanently deletes rows from Salesforce targets in a Bulk API target session. |
| Set the Interval for Polling BULK Job Status | Number of seconds the Data Integration Service waits before polling Salesforce for information about a Bulk API target session. Enter a positive integer. By default, the Data Integration Service polls every 10 seconds. |
| Use SFDC Error File | Generates error log files for a Bulk API target session. You must specify the SFDC error file directory and select the Monitor Bulk Job Until All Batches Processed session properties to generate the error log files. By default, the Data Integration Service does not generate error log files. Not applicable to a Standard API target session. |
| Use SFDC Success File | Generates success log files for a Bulk API target session. You must specify the SFDC success file directory and select the Monitor Bulk Job Until All Batches Processed session properties to generate the success log files. By default, the Data Integration Service does not generate success log files. Not applicable to a Standard API target session. |
| Set prefix for BULK success and error files | Adds the prefix to the BULK success and error files names when created. |
| SFDC Success File Directory | Mandatory, if you want to create success log file for the Bulk API target. Specify a directory on the machine that hosts the Data Integration Service. |
| Set the location of the BULK error files | Mandatory, if you want to create error log file for the Bulk API target. Specify a directory on the machine that hosts the Data Integration Service. |

Target Properties of the Data Object Write Operation

The target properties represent the data that is used to populate the Salesforce data object that you added when you created the data object. The target properties of the data object write operation include general and column properties that apply to the Salesforce objects. You can view the target properties of the data object write operation from the **General**, **Column**, and **Advanced** tabs.

General Properties

The general properties display the name and description of the Salesforce objects.

Column Properties

The column properties display the data types, precision, and scale of the target property in the data object write operation.

You can view the following target column properties of the data object write operation:

| Property | Description |
|-------------|---|
| Name | Name of the column |
| Type | Native data type of the column property |
| Precision | Maximum number of significant digits for numeric data types, or maximum number of characters for string data types. For numeric data types, precision includes scale. |
| Scale | Maximum number of digits after the decimal point for numeric values |
| Primary Key | Determines whether the column property is a part of the primary key |
| Description | Description of the column property |

Advanced Properties

The advanced properties displays the physical name of the Salesforce objects.

Importing a Salesforce Data Object

Import a Salesforce data object to read data from a Salesforce object.

1. Select a project or folder in the Object Explorer view.
2. Click **File > New > Data Object**.
3. Select **Salesforce Data Object** and click **Next**.
The **New Salesforce Data Object** dialog box appears.
4. Enter a name for the data object.
5. Click **Browse** next to the **Location** option and select the target project or folder.
6. Click **Browse** next to the **Connection** option and select the Salesforce connection from which you want to import the Salesforce object.

7. To add an object, click **Add** next to the **Selected Resource(s)** option.
The **Add Resource** dialog box appears.
8. Select a Salesforce object. You can search for it or navigate to it.
 - Navigate to the Salesforce object that you want to import and click **OK**.
 - To search for the Salesforce object, enter the name of the Salesforce object you want to add. Click **OK**.
9. If required, add additional objects to the Salesforce data object.
You can also add objects to a Salesforce data object after you create it.
10. Click **Finish**.
The data object appears under Physical Data Objects in the project or folder in the Object Explorer view.

Creating a Salesforce Data Object Read or Write Operation

You can add a Salesforce data object read or write operation to a mapping or maplet as a source. You can create the data object read or write operation for one or more Salesforce data objects.

Before you create a Salesforce data object read or write operation, you must create at least one Salesforce data object.

1. Select the data object in the Object Explorer view.
2. Right-click and select **New > Data Object Operation**.
The **Data Object Operation** dialog box appears.
3. Enter a name for the data object read or write operation.
4. Select **Read** or **Write** as the type of data object operation.
5. Click **Add**.
The **Select Resources** dialog box appears.
6. Select the Salesforce object for which you want to create the data object read or write operation and click **OK**.
7. Click **Finish**

The Developer tool creates the data object read or write operation for the selected data object.

CHAPTER 5

Salesforce Mappings

This chapter includes the following topics:

- [Salesforce Mappings Overview, 26](#)
- [Salesforce Mapping Read Example, 26](#)
- [Salesforce Mapping Write Example, 27](#)

Salesforce Mappings Overview

After you create a Salesforce data object read or write operation, you can develop a mapping.

You can define the following objects in the mapping:

- Salesforce data object read operation as the input to read data from Salesforce metadata.
- Relational, flat file, or any supported data object as the output.
- Relational, flat file, or any supported data object as the input.
- Salesforce data object write operation as the output to write data to Salesforce data objects.

Validate and run the mapping to read data from Salesforce sources, and write to a Salesforce object.

Salesforce Mapping Read Example

Your organization needs to migrate real-time sales opportunity information from a Salesforce system that is used by the sales team to a relational data source that is used internally by the executive sales management team.

Create a mapping that reads opportunity information in real time and writes those records to a table.

You can use the following objects in a Salesforce mapping:

Mapping Input

The mapping source is a Salesforce data object that contains the Opportunity object. Add the Opportunity object to the physical data object.

Create a data object read operation. Add the data object read operation to the mapping.

Mapping Output

Add a relational data object to the mapping as an output.

After you run the mapping, the Data Integration Service writes the extracted opportunity information to the target table. Sales managers can use the information to track sales opportunities.

Salesforce Mapping Write Example

Your organization needs to update real-time sales order processing status from an ERP system that is used by the logistics team to a Salesforce system that was used to create the order.

Create a mapping that reads real-time sales order processing status from the ERP system, and writes those records to Salesforce.

You can use the following objects in a Salesforce mapping:

Mapping Input

Add a relational data object to the mapping as an input.

Mapping Output

Add a Salesforce data object write operation to the mapping as an output.

The mapping target is a Salesforce data object that contains the Order object. Add the Order object to the physical data object.

Create a data object write operation and specify the update strategy in the data object write operation. Add the data object write operation to the mapping.

After every mapping run, the Data Integration Service writes the extracted order status information to the target table. Sales managers can use the updated information to track sales orders.

CHAPTER 6

Salesforce Run Time Processing

This chapter includes the following topics:

- [Salesforce Run-time Processing Overview, 28](#)
- [Filtering Source Data by Using the SOQL Filter Condition, 28](#)
- [Capturing Deleted and Archived Salesforce Records, 29](#)
- [Enable Bulk Query, 29](#)
- [Use SFDC Bulk API, 29](#)
- [Configuring the Upsert Target Operation, 30](#)
- [Configuring the Maximum Batch Size, 30](#)
- [Handling Null Values in Update and Upsert Operations, 31](#)
- [Override an External ID with an idLookup for Upserts, 31](#)

Salesforce Run-time Processing Overview

When you develop a Salesforce mapping, you define the data object operation read or write properties. The data object read operation determines how the Data Integration Service reads data from Salesforce sources, and the data object write operation determines how the Data Integration Service writes data to Salesforce targets.

Filtering Source Data by Using the SOQL Filter Condition

When you configure a mapping that reads data from a Salesforce source, you can enter a filter condition to filter records read from the source. When you enter a filter condition, the Data Integration Service adds the WHERE clause to the SOQL query and generates an SOQL query based on the objects and fields included in the Salesforce source.

To filter records from a Salesforce source, set the SOQL filter condition in the data object read operation. For example, enter the following filter condition to read records from the Salesforce Account object that were created before October 30, 2012:

```
CreatedDate < '2012-10-30T00:00:00.000Z'
```

Enter a filter condition based on the SOQL syntax defined in the Salesforce documentation. The Salesforce API validates the SOQL syntax at run time. If you enter a filter condition that is not valid, the mapping fails.

Capturing Deleted and Archived Salesforce Records

The Data Integration Service can capture active, deleted, and archived records from a Salesforce object. By default, mappings do not capture deleted and archived records.

To capture deleted and archived records, configure the Use queryAll data object read operation property.

Enable Bulk Query

The Data Integration Service can read data from Salesforce sources using the Salesforce Bulk API. Use the Bulk API to read large amounts of data from Salesforce while generating a minimal number of API calls.

With the Bulk API, each batch of data can contain up to approximately 1 GB of data in CSV format. When the Data Integration Service creates a batch, it adds any required characters to properly format the data, such as adding quotation marks around text.

You can also monitor the progress of batches in the log file.

To configure a mapping to use the Salesforce Bulk API, select the Enable Bulk Query data object operation read property.

Note: Bulk read mode ignores the queryAll option.

Use SFDC Bulk API

The Data Integration Service can use the Salesforce Bulk API to write data to Salesforce objects. Use the Bulk API to write large amounts of data to Salesforce with a minimal number of API calls.

With a Bulk API write, each batch of data can contain up to 10,000 records or one million characters of data in CSV format. When the Data Integration Service creates a batch, it adds required characters such as, quotation marks around text, to format the data.

You can configure a Bulk API target session to load batches serially or in parallel. By default, the data load is in parallel mode, but you can override the data load to serial mode. You can also monitor the progress of batches in the session log.

To configure a session to use the Bulk API for Salesforce targets, select the Use SFDC Bulk API session property. When you select this property, the Data Integration Service ignores the Max Batch Size session property.

Configuring the Upsert Target Operation

The Salesforce upsert operation creates a new record or updates an existing record in a Salesforce object. You must provide one of the following types of fields to upsert records to a Salesforce object:

External ID field

You can use a custom Salesforce field to uniquely identify each record in a Salesforce object. You can create a custom external ID field for each object in Salesforce. You can view the properties of a Salesforce object to check whether the object includes an external ID field.

idLookup field

You can use a Salesforce idLookup field to identify each record in a Salesforce object. Salesforce creates idLookup fields for each standard Salesforce object. For example, the Email field is an idLookup field for the Contact object. Custom Salesforce objects do not contain an idLookup field. For more information about idLookup fields, see the Salesforce documentation.

A Salesforce target object might have multiple external ID or idLookup fields. By default, the Data Integration Service uses the first external ID or idLookup field it encounters. However, you can specify the external ID or idLookup field to use for the upsert operation in the run-time properties.

To configure the upsert operation to write to a Salesforce object, perform the following steps:

1. Map the external ID or idLookup field from the source to the target in the mapping. If you are using an external ID, map the external ID to the external ID field in the Salesforce object. If you are using an idLookup field, map the field to the appropriate target field. For example, map the email source field to the Email field in the Salesforce Contact object.
2. Configure the Treat Insert as Upsert or Treat Update as Upsert run-time property to configure a Salesforce run-time property to upsert records.
3. To use the idLookup field instead of an external ID field, enable the Use idLookup Field for Upserts run-time property. By default, the Data Integration Service uses an external ID for upserts. You can configure the run-time property to override the external ID and use the idLookup, instead.
4. To specify an external ID or idLookup field, enter the external ID or idLookup field name in the Use this ExternalId/idLookup Field for Upserts run-time property.

Note: If you do not enter the name of the external ID or idLookup field, the Data Integration Service selects the first external ID or idLookup field it encounters. If you specify a field that is not an external ID or idLookup field, or if you misspell the field name, the run-time property fails.

Configuring the Maximum Batch Size

The Data Integration Service writes data to a Salesforce target as a batch. The Max Batch Size attribute in the session properties determines the maximum number of records the Data Integration Service can write to a Salesforce target in a batch. The Salesforce service can receive a maximum of 200 records in a single insert, update, or delete operation.

To minimize the number of calls made to the Salesforce service, each batch must accommodate the maximum number of records as configured in the Max Batch Size property.

Handling Null Values in Update and Upsert Operations

By default, the Data Integration Service does not replace existing values in a Salesforce record with null values from the source during an update or upsert operation. To replace existing values with null values, configure the Set Fields to NULL session property for the Salesforce target.

You cannot set the value of an external ID field in a Salesforce target to NULL. The session fails if you enable the Set Fields to NULL session property and the session tries to replace the value in an external ID field with a null value.

Override an External ID with an idLookup for Upserts

The Data Integration Service can use the external ID or idLookup fields when performing an upsert operation to identify records in a Salesforce target. By default, the Data Integration Service uses the external ID field for upserts. You can configure the session to override the external ID field and use the idLookup field, instead.

APPENDIX A

Data Type Reference

This appendix includes the following topics:

- [Data Type Reference Overview, 32](#)
- [Salesforce Data Types and Transformation Data Types, 32](#)

Data Type Reference Overview

The Developer tool uses the following data types in PowerExchange for Salesforce mappings.

Salesforce native data types

Salesforce native data types appear in the physical data object column properties.

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 Data Integration Service uses to move data across platforms.

Transformation data types appear in all transformations in a mapping.

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

Salesforce Data Types and Transformation Data Types

The following table lists the Salesforce data types that Data Integration Service supports and the corresponding transformation data types

| Salesforce Data Type | Range and Description | Transformation Data Type |
|----------------------|---|--------------------------|
| AnyType | Polymorphic data type that returns string, picklist, reference, boolean, currency, integer, double, percent, ID, date, datetime, URL, or email data | String |
| Base64 | Base64 encoded binary data | String |

| Salesforce Data Type | Range and Description | Transformation Data Type |
|----------------------------|--|--------------------------|
| Boolean | Boolean (true/false) values | Integer |
| Byte | A set of bits | String |
| Combobox | Enumerated values | String |
| Currency | Currency values | Decimal |
| DataCategoryGroupReference | Types of category groups and unique category names | String |
| Date | Date values | Date/Time |
| DateTime | Date and time values | Date/Time |
| Double | Double values | Decimal |
| Email | Email addresses | String |
| Encrypted String | Encrypted text fields contain any combination of letters, numbers, or symbols that are stored in encrypted form | String |
| ID | Primary key field for a Salesforce object | String |
| Int | Fields of this type contain numbers with no fraction portion | Integer |
| Master record | ID of the merged record | String |
| Multipicklist | Multiple-selection picklists, which provide a set of enumerated values from which you can select multiple values | String |
| Percent | Percentage values | Decimal |
| Phone | Phone numbers | String |
| Picklist | Single-selection picklists, which provide a set of enumerated values that you can select one value from | String |
| Reference | Cross-references to another Salesforce object | String |
| String | Character strings | String |
| Textarea | String that appears as a multiple-line text field | String |
| Time | Time values | Date/Time |
| URL | URL values | String |

INDEX

A

advanced properties
input [22](#)

B

batch size
description for Salesforce [30](#)
bulk API [29](#)
Bulk API target session
configuring for Salesforce [29](#)

C

capture deleted and archived records [29](#)
column properties [18](#)
configuring for Salesforce
Bulk API target sessions [29](#)
configuring HTTP proxy options
Developer tool [10](#)
creating
Salesforce connection [14](#)
Salesforce data object read operation [25](#)
custom Salesforce objects [15](#)

D

data object read operation
creating [25](#)
datatype reference overview [32](#)
DTM Buffer Size
configuring for Salesforce [30](#)

E

external ID
description [30](#)
overriding with idLookup [31](#)

G

general properties
input [22](#)

I

idLookup
description for Salesforce [30](#)
overriding the external ID [31](#)

importing
Salesforce data object [24](#)
input properties [21](#)

M

mapping output [26, 27](#)

N

nulls
handling in upserts and updates [31](#)

O

overview
Salesforce data object [15](#)

P

performance
configuring buffer block size for Salesforce [30](#)
PowerExchange for Salesforce
overview [8](#)
properties
Salesforce data object [17](#)
Salesforce data object read operation [17](#)

Q

query properties
data object read operation [19](#)

R

related objects [16](#)
rules and guidelines
related objects [16](#)
run-time processing
overview [28](#)

S

Salesforce
mapping example [26, 27](#)
mapping input [26, 27](#)
mapping overview [26](#)
Salesforce connection
creating [14](#)

- Salesforce connection (*continued*)
 - overview [12](#)
 - properties [12](#)
- Salesforce data object
 - importing [24](#)
- Salesforce data object read operation
 - creating [25](#)
 - properties [17](#)
- session conditions
 - DTM Buffer Size for Salesforce [30](#)
- Set Fields to NULL
 - session property [31](#)
- source properties [17](#)

- standard Salesforce objects [15](#)

U

- upsert
 - configuring for Salesforce [30](#)
 - description for Salesforce [30](#)
 - external ID [30](#)
 - overriding external ID with idLookup [31](#)
 - Salesforce idLookup field [30](#)
 - Salesforce session configuration [30](#)
- use query all [29](#)